D607 - Assignment 4 - Tidying Data

Marco Castro

2024-09-28

Overview

In this assigment, we were tasked with importing a messy (un-tidy) dataset, performing tiddying operations, and performing some basic analysis on our data.

First, I read the csv stored on my github page

```
# load messy dataframe from github page
messy_numbersense <- read_csv("https://raw.githubusercontent.com/mcastro64/d607-assignments/refs/heads/m
```

```
## Rows: 2 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): airline
## dbl (10): On Time - Los Angeles, On Time - Phoenix, On Time - San Diego, On ...
##
## 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.
```

```
messy_numbersense |>
  gt() |>
  tab_spanner(
    label = 'On Time',
    columns = matches("On Time*")
  ) |>
  tab_spanner(
    label = 'Delayed',
    columns = matches("Delayed*")
  )
```

	On Time			
airline	On Time - Los Angeles	On Time - Phoenix	On Time - San Diego	On Time - San Fran
ALASKA	497	221	212	
AM WEST	694	4840	383	

Pivoting Data

Observations for both on time and delayed flight status for each destination are written along the columns of the dataset, making it difficult to analyze. Our next step is to pivot the table from wide to long, breaking up

the columns to individual rows for flight status and destination using the *names_to* parameter to indicate which columns we want to create based on column names. The *names_to* parameter was set to the delimitor " - " in order for dplyr to know how to assign the values to the status and destination columns. Finally, the parament *value_to* was used to pass the value of the original column from the messy dataset to a new column called "flighs".

```
numbersense_longer <- messy_numbersense |>
  pivot_longer(
   cols = !(airline),
    names_to = c("status", "destination"),
    names_sep = " - ",
    values_to = "flights"
  )

numbersense_longer |>
  gt()
```

airline	status	destination	flights
ALASKA	On Time	Los Angeles	497
ALASKA	On Time	Phoenix	221
ALASKA	On Time	San Diego	212
ALASKA	On Time	San Francisco	503
ALASKA	On Time	Seattle	1841
ALASKA	Delayed	Los Angeles	62
ALASKA	Delayed	Phoenix	12
ALASKA	Delayed	San Diego	20
ALASKA	Delayed	San Francisco	102
ALASKA	Delayed	Seattle	305
AM WEST	On Time	Los Angeles	694
AM WEST	On Time	Phoenix	4840
AM WEST	On Time	San Diego	383
AM WEST	On Time	San Francisco	320
AM WEST	On Time	Seattle	201
AM WEST	Delayed	Los Angeles	117
AM WEST	Delayed	Phoenix	415
AM WEST	Delayed	San Diego	65
AM WEST	Delayed	San Francisco	129
AM WEST	Delayed	Seattle	61

Calculating Frequency Delayed and On-time

Step 1: Calculating total flights by destination airline

Next, I wanted to be able to calculate the percentage of flights that were delayed by airline destination. To do this, I first calculated the total number of flights per airline destination using *group_by* and *summarise* to perform a *sum* calculation across all flights for every airline going to a specific destination, iregardless of status (on-time/delayed).

```r
# calculate total number of flights per airline destination
total_flights_per_airline_destination <- numbersense_longer |>
  group_by(airline, destination) |>
  summarise(
    total_flights = sum(flights)
  )
```

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

```r
total_flights_per_airline_destination
```

```
## # A tibble: 10 x 3
## # Groups:   airline [2]
##    airline destination   total_flights
##    <chr>   <chr>                 <dbl>
##  1 ALASKA  Los Angeles             559
##  2 ALASKA  Phoenix                 233
##  3 ALASKA  San Diego               232
##  4 ALASKA  San Francisco           605
##  5 ALASKA  Seattle                2146
##  6 AM WEST Los Angeles             811
##  7 AM WEST Phoenix                5255
##  8 AM WEST San Diego               448
##  9 AM WEST San Francisco           449
## 10 AM WEST Seattle                 262
```

Step 2: Joining datasets

Next, I used a *left_join* to combine our dataframe in long form and our dataframe with total flights per airline destination. As we are focusing on delayed flights, I used the *filter* function on the status column to limit our results to flights with a "Delayed" status.

```r
# join long flights table and total flights table
# then calculate frequence of delays
delayed_flights_observations <- numbersense_longer |>
  left_join(total_flights_per_airline_destination, join_by(airline == airline, destination == destinati
  filter(status == "Delayed") |>
  rename(delayed_flights = flights)

delayed_flights_observations |>
  gt() |>
  tab_header("Delayed Flights by Airline Destination")
```

Delayed Flights by Airline Destination

airline	status	destination	delayed_flights	total_flights
ALASKA	Delayed	Los Angeles	62	559
ALASKA	Delayed	Phoenix	12	233

ALASKA	Delayed	San Diego	20	232
ALASKA	Delayed	San Francisco	102	605
ALASKA	Delayed	Seattle	305	2146
AM WEST	Delayed	Los Angeles	117	811
AM WEST	Delayed	Phoenix	415	5255
AM WEST	Delayed	San Diego	65	448
AM WEST	Delayed	San Francisco	129	449
AM WEST	Delayed	Seattle	61	262

Step 3: Calculating Overall Frequency of Delayed Flights by Airline

I wanted to get a sense of what percentage of flights were delayed overall per airline, regardless of destination. Working with the dataframe output from step 2, I again used *group_by* and *summarize* to calculate the number of delayed flights per airline, the total number of flights per airline and the percentage of flights delayed by airline.

```
summary_delayed_flights_by_airline <- delayed_flights_observations |>
  group_by(airline) |>
  summarize(
    delayed_flights_by_airline = sum(delayed_flights),
    total_flights_by_airline = sum(total_flights),
    percent_delayed_by_airline = round(delayed_flights_by_airline / total_flights_by_airline, 3)
  )


summary_delayed_flights_by_airline |>
  gt() |>
  fmt_percent(percent_delayed_by_airline) |>
  tab_header("Overall Frequency of Delayed Flights by Airline") |>
  cols_label(
    airline = "Airlane",
    delayed_flights_by_airline = "# Flights Delayed",
    total_flights_by_airline = "Total Flights",
    percent_delayed_by_airline = "% Flights Delayed"
  )
```

Overall Frequency of Delayed Flights by Airline

Airlane	# Flights Delayed	Total Flights	% Flights Delayed
ALASKA	501	3775	13.30%
AM WEST	787	7225	10.90%

Step 4: Calculating Frequency of Delayed Flights by Airline Destination

Next, I performed the same calculation on our *delayed_flights_observations* dataframe to calculate the number of delayed flights per airline destination, the total number of flights per airline destination and the percentage of flights delayed by airline destination.

```
delayed_flights_by_airline_destination <- delayed_flights_observations |>
  mutate(percent_delayed = delayed_flights/total_flights) |>
  subset(select = c(airline, destination, delayed_flights, percent_delayed))

delayed_flights_by_airline_destination |>
  gt() |>
  fmt_percent(percent_delayed) |>
  tab_header("Frequency of Delayed Flights by Airline Destination") |>
  cols_label(
    airline = "Airlane",
    delayed_flights = "# Flights Delayed",
    percent_delayed = "% Flights Delayed"
  )
```

Frequency of Delayed Flights by Airline Destination

Airlane	destination	# Flights Delayed	% Flights Delayed
ALASKA	Los Angeles	62	11.09%
ALASKA	Phoenix	12	5.15%
ALASKA	San Diego	20	8.62%
ALASKA	San Francisco	102	16.86%
ALASKA	Seattle	305	14.21%
AM WEST	Los Angeles	117	14.43%
AM WEST	Phoenix	415	7.90%
AM WEST	San Diego	65	14.51%
AM WEST	San Francisco	129	28.73%
AM WEST	Seattle	61	23.28%

Comparing our results

Building a bar chart for the count of flights delayed

Using ggplot, I created a simple bar chart plotting airlines and the count of delayed flights. I used the parameter position="dodge" to transform the result from a stacked bar chart to a individual bars for each destination along the x-axis.

```
delayed_flights_count_chart <- ggplot(delayed_flights_by_airline_destination, aes(x=airline, y=delayed_
  geom_bar(
    stat='identity',
    position = position_dodge(.85),
    width = .7
  ) +
  labs(
    title = "Counts of Airline Flights Delayed per Destination",
    x = "Airline",
    y = "Number of Flights Delayed"
  ) +
  geom_text(
    aes(label = delayed_flights),
```

```
    vjust = -0.5,
    size = 7,
    size.unit = "pt",
    position = position_dodge(.7)
  ) +
  theme(
    plot.title = element_text(size = 10),
    legend.position = "bottom",
    aspect.ratio = 4/5
  ) +
  ylim(0, 450)
```

Building a bar chart for the frequency of flights delayed

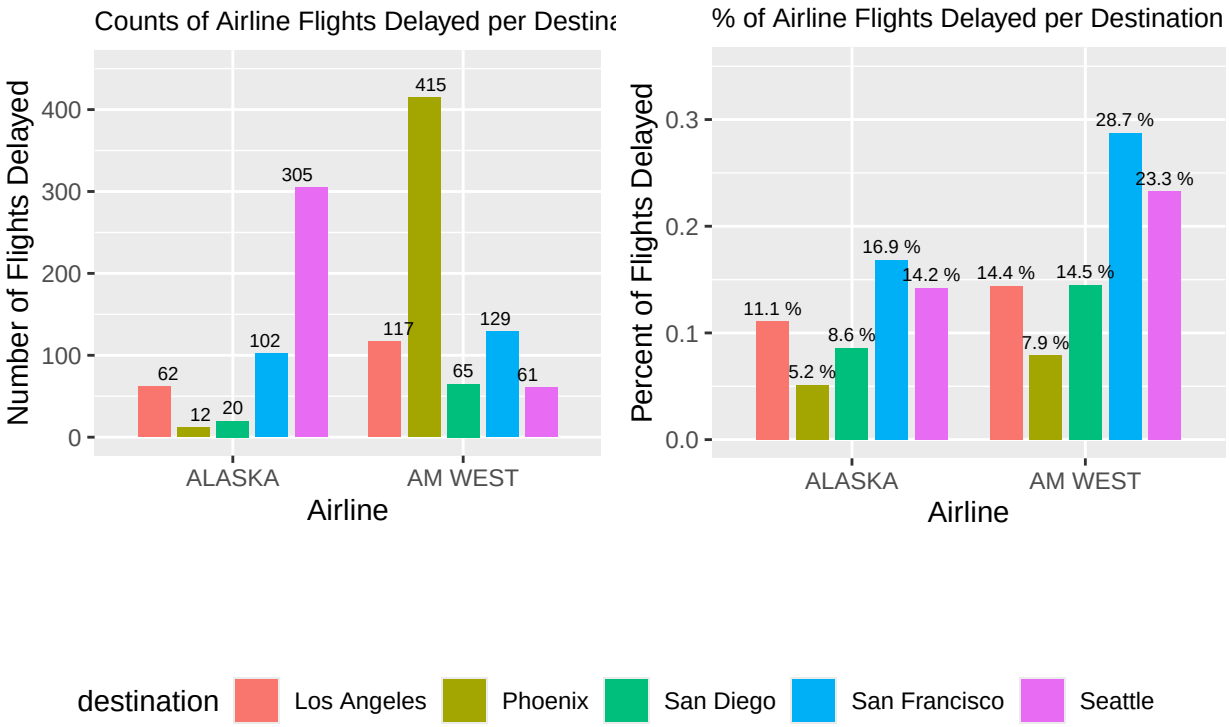
I also created a simple bar chart plotting airlines and the percentage of delayed flights. I used the parameter position="dodge" to transform the result from a stacked bar chart to a individual bars for each destination along the x-axis.

```
delayed_flights_freq_chart <- ggplot(delayed_flights_by_airline_destination, aes(x=airline, y=percent_d
  geom_bar(
    stat = 'identity',
    position=position_dodge(.85),
    width = 0.7
  ) +
  labs(
    title = "% of Airline Flights Delayed per Destination",
    x = "Airline",
    y = "Percent of Flights Delayed"
  ) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(
    aes(label = paste(round(percent_delayed* 100, 1), "%")),
    vjust = -0.5,
    size = 7,
    size.unit = "pt",
    position = position_dodge(.85)
  ) +
  theme(
    plot.title = element_text(size = 10),
    legend.position = "bottom",
    aspect.ratio = 4/5
  ) +
  ylim(0, .35)
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

I used ggpubr to plot them side-by-side for comparison.

```
ggarrange(delayed_flights_count_chart, delayed_flights_freq_chart, ncol=2, common.legend = TRUE, legend=
```



Conclusion Examining raw counts, AM West airline's flights to Phoenix had the most delays of any group with 415 delayed arrivals. However, when we compare percent of flights delayed, we see that AM West to Phoenix actually had the second lowest percent of delayed flights (7.9%), second only to Alaska Airline's flight to Phoenix (5.2%). Looking at the total number of flights per airline destination table helps us confirm that AM West to Phoenix had the most flights of any airline to any destination with 5255. Additionally, flights to Phoenix from AM West and Alaska airlines were below their overall percent of delayed flights (10.9% and 13.3% respectively).