Tidying and Transforming Data- Assignment 5A

Kiera Griffiths, Desiree Thomas

Approach

For this assignment, we are asked to reconstruct a dataset from a summarized chart, store it in CSV format (CSV or database), then use tidyr and dplyr packages in R to clean and analyze it. We will convert the dataset from wide format to tidy, then perform percentage-based comparisons between two airlines, both overall and by city. We will explain differences between comparisons and create visualizations to support our conclusions. It will be imperative that we ensure values are documented when we convert dataset to different formats.

Loads tidyverse library which includes: dplyr, tidyr, ggplot2.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.2.0     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.1
## v lubridate 1.9.4     v tidyr     1.3.2
## v purrr     1.2.1
## -- 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 error
```

Reads CSV into a dataframe called airline. Convert raw data into structured, analyzable format and also identify columns missing headers and white space.

```
airline <- read.csv("https://raw.githubusercontent.com/KieraG2026/Tidying-and-Transforming-Data/refs/hea
```

Add in column headers Airline and Status.

```r
colnames(airline) <- c(
  "Airline",
  "Status",
  "Los Angeles",
  "Phoenix",
  "San Diego",
  "San Francisco",
  "Seattle"
)

airline
```

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

Convert blank spaces to NA.

```r
airline$Airline[airline$Airline == ""] <- NA
airline$Status[airline$Status == ""] <- NA

airline
```

```
##   Airline  Status Los Angeles Phoenix San Diego San Francisco Seattle
## 1    <NA>    <NA> Los Angeles Phoenix San Diego San Francisco Seattle
## 2  ALASKA on time         497     221       212           503  1,841
## 3    <NA> delayed          62      12        20           102     305
## 4    <NA>    <NA>
## 5 AM WEST on time         694  4,840       383           320     201
## 6    <NA> delayed         117     415        65           129      61
```

Removes NA rows starting with NA cells under Status column.

```r
airline <- airline %>% drop_na(Status)

airline
```

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

Fills in remaining missing cells under Airline column with airline name.

```
airline <- airline %>% fill(Airline)

airline
```

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

Removes commas and extra spaces from numbers.

```
airline[ , 3:7] <- lapply(airline[ , 3:7], function(x) as.numeric(gsub(",", "", x)))

airline
```

```
##   Airline  Status Los Angeles Phoenix San Diego San Francisco Seattle
## 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
```

Tidy data from wide to long format.

```
library(tidyverse)

airline_long <- airline %>%
  pivot_longer(
    cols = 3:7,
    names_to = "City",
    values_to = "Flights"
  )

head(airline_long)
```

```
## # A tibble: 6 x 4
##   Airline Status  City          Flights
##   <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
```

Compare percentage of delays or arrival rates for each airline.

```
airline_summary <- airline_long %>%
  group_by(Airline, Status) %>%
  summarise(Total = sum(Flights)) %>%
  mutate(Percentage = round(Total / sum(Total) * 100, 0))
```

```
## ‘summarise()‘ has regrouped the output.
## i Summaries were computed grouped by Airline and Status.
## i Output is grouped by Airline.
## i Use ‘summarise(.groups = "drop_last")‘ to silence this message.
## i Use ‘summarise(.by = c(Airline, Status))‘ for per-operation grouping
##   (‘?dplyr::dplyr_by‘) instead.
```

```
airline_summary
```

```
## # A tibble: 4 x 4
## # Groups:   Airline [2]
##   Airline Status  Total Percentage
##   <chr>   <chr>   <dbl>      <dbl>
## 1 ALASKA  delayed   501         13
## 2 ALASKA  on time  3274         87
## 3 AM WEST delayed   787         11
## 4 AM WEST on time  6438         89
```

Compare percentage of delays or arrival rates for each airline, by city.

```
airline_city_summary <- airline_long %>%
  group_by(Airline, Status, City) %>%
  summarise(Total = sum(Flights)) %>%
  mutate(Percentage = round(Total / sum(Total) * 100, 0))
```

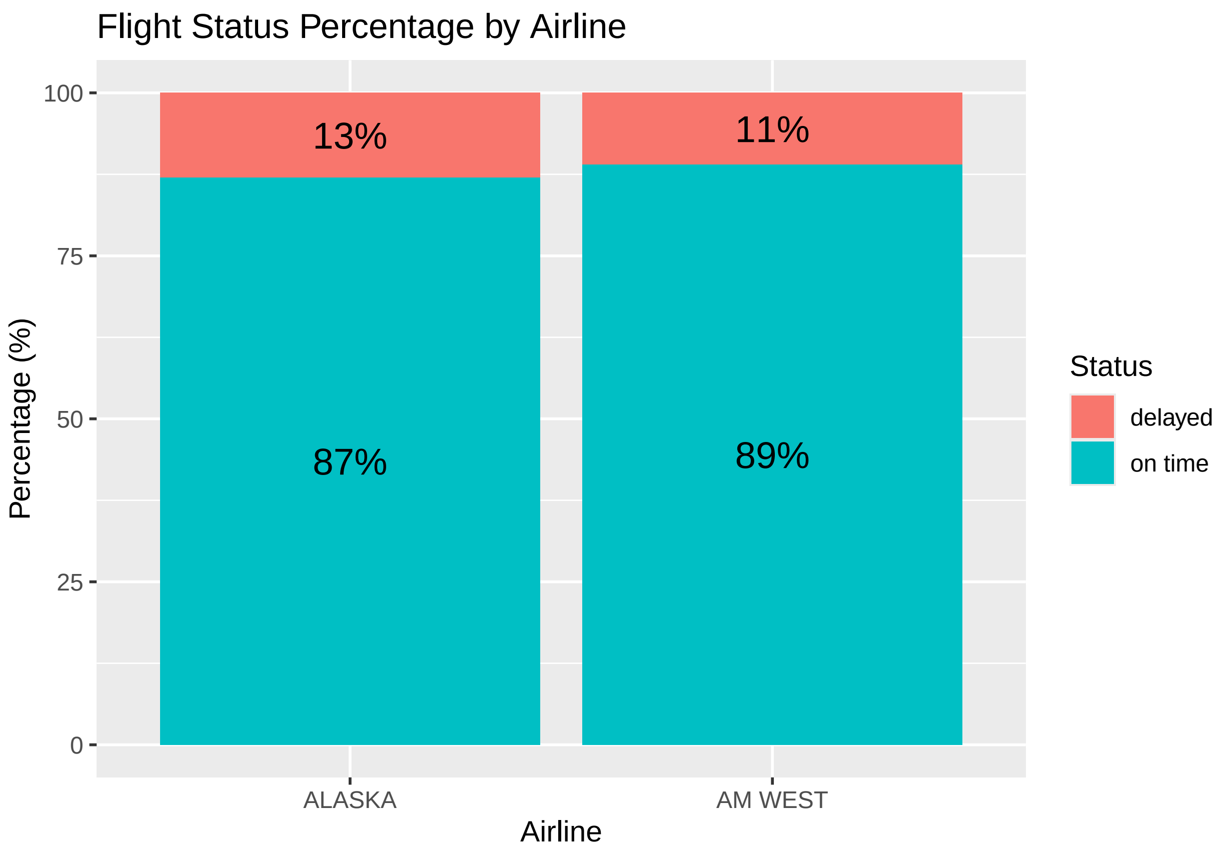
```
## ‘summarise()‘ has regrouped the output.
## i Summaries were computed grouped by Airline, Status, and City.
## i Output is grouped by Airline and Status.
## i Use ‘summarise(.groups = "drop_last")‘ to silence this message.
## i Use ‘summarise(.by = c(Airline, Status, City))‘ for per-operation grouping
##   (‘?dplyr::dplyr_by‘) instead.
```

```
airline_city_summary
```

```
## # A tibble: 20 x 5
## # Groups:   Airline, Status [4]
##    Airline Status  City          Total Percentage
##    <chr>   <chr>   <chr>         <dbl>      <dbl>
##  1 ALASKA  delayed Los Angeles      62         12
##  2 ALASKA  delayed Phoenix          12          2
##  3 ALASKA  delayed San Diego        20          4
##  4 ALASKA  delayed San Francisco   102         20
##  5 ALASKA  delayed Seattle         305         61
##  6 ALASKA  on time Los Angeles     497         15
##  7 ALASKA  on time Phoenix         221          7
##  8 ALASKA  on time San Diego       212          6
##  9 ALASKA  on time San Francisco   503         15
## 10 ALASKA  on time Seattle        1841         56
## 11 AM WEST delayed Los Angeles     117         15
## 12 AM WEST delayed Phoenix         415         53
## 13 AM WEST delayed San Diego        65          8
## 14 AM WEST delayed San Francisco   129         16
## 15 AM WEST delayed Seattle          61          8
## 16 AM WEST on time Los Angeles     694         11
## 17 AM WEST on time Phoenix        4840         75
## 18 AM WEST on time San Diego       383          6
## 19 AM WEST on time San Francisco   320          5
## 20 AM WEST on time Seattle         201          3
```

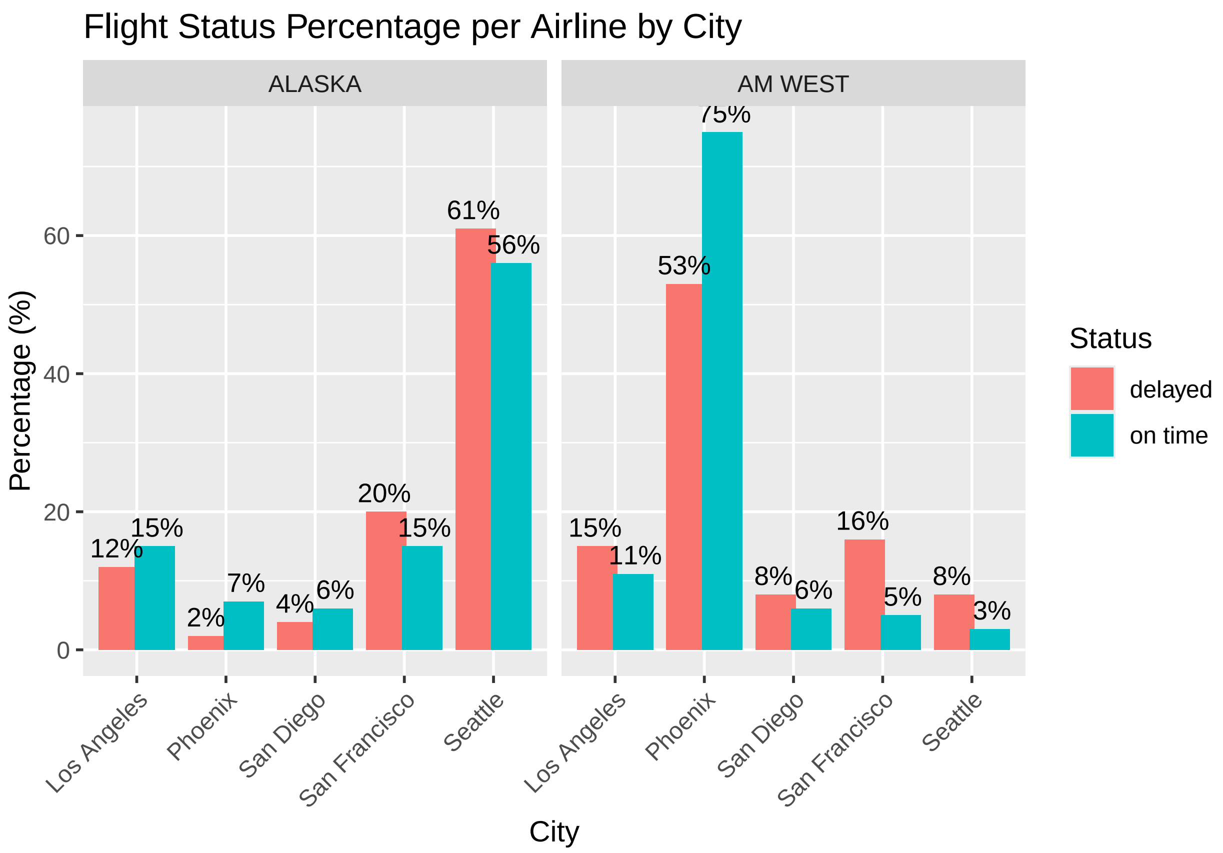
Plot flight status rates by airlines bar graph.

```
ggplot(airline_summary, aes(x = Airline, y = Percentage, fill = Status)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(Percentage, "%")),
            position = position_stack(vjust = 0.5), color = "black", size = 5) +
  labs(title = "Flight Status Percentage by Airline", y = "Percentage (%)")
```



Plot flight status rates per airlines, by city bar graph.

```
ggplot(airline_city_summary, aes(x = City, y = Percentage, fill = Status)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8)) +
  geom_text(aes(label = paste0(Percentage, "%")),
            position = position_dodge(width = 0.9),
            vjust = -.5, size = 3.5) +
  facet_wrap(~Airline) +
  labs(title = "Flight Status Percentage per Airline by City",
       y = "Percentage (%)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Flight Status Percentage per Airline by City
ALASKA
AM WEST
Percentage (%)
0
20
40
60
12%
15%
2%
7%
4%
6%
20%
15%
61%
56%
15%
11%
53%
75%
8%
6%
16%
5%
8%
3%
Los Angeles
Phoenix
San Diego
San Francisco
Seattle
Los Angeles
Phoenix
San Diego
San Francisco
Seattle
City
Status
delayed
on time