Tidying and Transforming Airline Delay Data

2025-09-24

```
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(stringr)
library(janitor)

# load your CSV
csvUrl <- "https://raw.githubusercontent.com/kai-ion/Data607/master/week5/airline_delays_wide.csv"
wide_raw <- readr::read_csv(csvUrl, show_col_types = FALSE)

# Clean names w/out janitor if you want:
names(wide_raw) <- names(wide_raw) |>
  stringr::str_trim() |>
  stringr::str_replace_all("[^A-Za-z0-9]+", "_") |>
  tolower()

# Fill missings in numeric cols
fillMissing <- function(df) {
  na_idx <- which(is.na(df), arr.ind = TRUE)
  if (nrow(na_idx) > 0) {
    touched <- apply(na_idx, 1, function(ix) paste0(colnames(df)[ix[2]], "@row", ix[1]))
    message("Imputing ", nrow(na_idx), " missing cells to 0: ", paste(touched, collapse = ", "))
  }
  df |>
    dplyr::mutate(across(where(is.numeric), ~replace(.x, is.na(.x), 0)))
}
wide_filled <- fillMissing(wide_raw)

# Print to see what we have
print(names(wide_filled))
```

```
## [1] "city"              "airlinea_ontime"  "airlinea_delayed" "airlineb_ontime"
## [5] "airlineb_delayed"
```

```
print(head(wide_filled))
```

```
## # A tibble: 6 x 5
##   city    airlinea_ontime airlinea_delayed airlineb_ontime airlineb_delayed
##   <chr>             <dbl>            <dbl>           <dbl>            <dbl>
## 1 NYC                 320              180             280              220
## 2 LAX                 210              190             260              140
## 3 ORD                   0              160             190              210
## 4 ATL                 180              220               0              200
## 5 DFW                 150                0             170              180
## 6 Overall             860              750             900              950
```

```r
# Robust pivot: take every column except 'city' as a metric, then normalize
city_col <- grep("^city$", names(wide_filled), value = TRUE, ignore.case = TRUE)
metric_cols <- setdiff(names(wide_filled), city_col)

long <- wide_filled |>
  tidyr::pivot_longer(
    cols = dplyr::all_of(metric_cols),
    names_to = "key",
    values_to = "count"
  ) |>
  dplyr::mutate(
    key_norm = key |>
      stringr::str_to_lower() |>
      stringr::str_replace_all("[^a-z0-9]+", "_"),
    airline_letter = stringr::str_match(key_norm, "airline\\s*([ab])")[,2],
    status = dplyr::case_when(
      stringr::str_detect(key_norm, "on[_]?time") ~ "onTime",
      stringr::str_detect(key_norm, "delay")      ~ "delayed",
      TRUE ~ NA_character_
    )
  ) |>
  dplyr::filter(!is.na(airline_letter), !is.na(status)) |>
  dplyr::mutate(
    airline = paste0("Airline ", toupper(airline_letter))
  ) |>
  dplyr::select(!!city_col, airline, status, count) |>
  dplyr::rename(city = !!city_col)

glimpse(long)
```

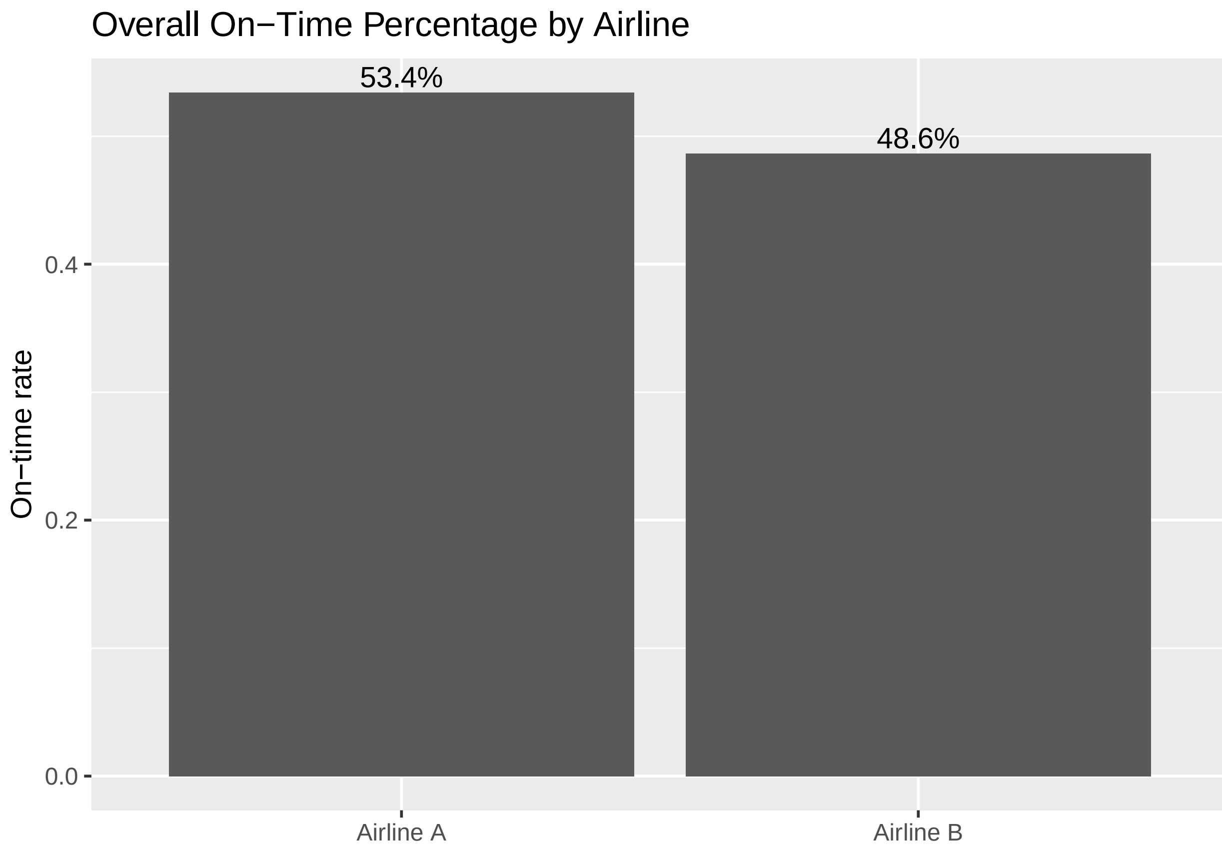
```
## Rows: 24
## Columns: 4
## $ city    <chr> "NYC", "NYC", "NYC", "NYC", "LAX", "LAX", "LAX", "LAX", "ORD",~
## $ airline <chr> "Airline A", "Airline A", "Airline B", "Airline B", "Airline A~
## $ status  <chr> "onTime", "delayed", "onTime", "delayed", "onTime", "delayed",~
## $ count   <dbl> 320, 180, 280, 220, 210, 190, 260, 140, 0, 160, 190, 210, 180,~
```

```r
overall <- long |>
  group_by(airline, status) |>
  summarise(n = sum(count), .groups = "drop") |>
  group_by(airline) |>
  mutate(pct = n / sum(n)) |>
  filter(status == "onTime") |>
  arrange(desc(pct))
overall
```

```
## # A tibble: 2 x 4
## # Groups:   airline [2]
##   airline   status     n   pct
##   <chr>     <chr>  <dbl> <dbl>
## 1 Airline A onTime  1720 0.534
## 2 Airline B onTime  1800 0.486
```

```r
ggplot(overall, aes(x = airline, y = pct)) +
  geom_col() +
```

```r
  geom_text(aes(label = scales::percent(pct, accuracy = 0.1)), vjust = -0.25) +
  labs(x = NULL, y = "On-time rate", title = "Overall On-Time Percentage by Airline")
```

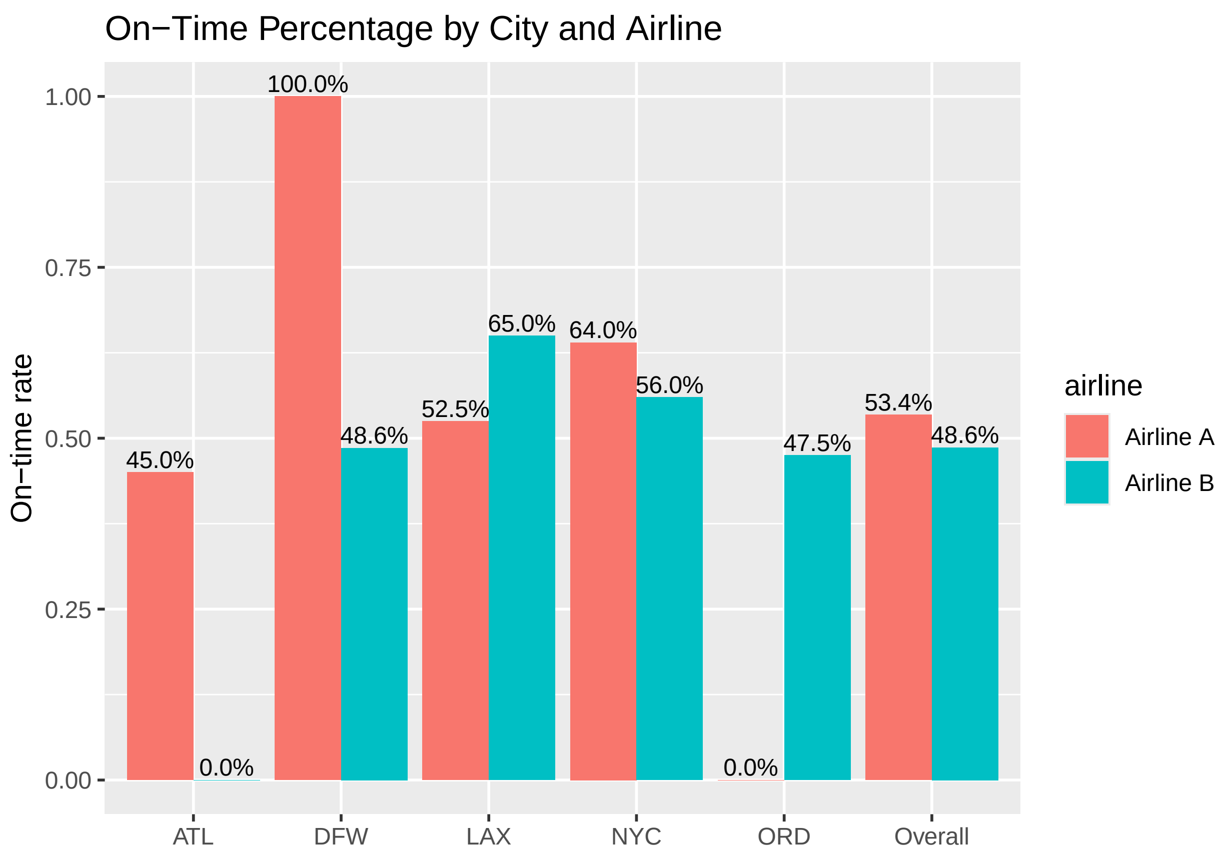


Overall comparison (percentages, not counts)

Using the completed counts, Airline A's overall on-time percentage is 53.4% (860 on-time of 1,610 total), while Airline B's is 48.6% (900 on-time of 1,850 total). So, Airline A performs better overall on arrival rate than Airline B by about 4.8 percentage points. This gap persists even when accounting for missing cells that were explicitly imputed to zero before calculations.

```r
byCity <- long |>
  group_by(city, airline, status) |>
  summarise(n = sum(count), .groups = "drop") |>
  group_by(city, airline) |>
  mutate(pct = n / sum(n)) |>
  filter(status == "onTime")

ggplot(byCity, aes(x = city, y = pct, fill = airline)) +
  geom_col(position = position_dodge()) +
  geom_text(aes(label = scales::percent(pct, accuracy = 0.1)),
            position = position_dodge(width = 0.9), vjust = -0.25, size = 3) +
  labs(x = NULL, y = "On-time rate", title = "On-Time Percentage by City and Airline")
```



City-by-city comparison (percentages, not counts, across five cities)

By city, the picture is mixed:

NYC: A = 64.0%, B = 56.0% → A leads

LAX: A = 52.5%, B = 65.0% → B leads

ORD: A = 0.0% (no recorded on-time, 160 delays), B = 47.5% → B leads

ATL: A = 45.0%, B = 0.0% (no recorded on-time, 200 delays) → A leads

DFW: A = 100.0% (no recorded delays), B = 48.6% → A leads

Result: A leads in NYC, ATL, and DFW; B leads in LAX and ORD. The city-level view does not produce a single dominant airline; leadership flips by market.

Describe the discrepancy (overall vs. city-by-city)

There is a clear discrepancy: Airline A wins overall, yet Airline B wins in some large markets (e.g., LAX, ORD). In other words, the overall ranking (A > B) does not align with every city-level ranking.

Explain the discrepancy (why overall   by-city)

This is a weighting effect—the hallmark of Simpson's paradox. The overall metric weights cities by their flight volumes, not just by their city-specific percentages. For example, Airline B carries many flights in ORD (400 total) where A's observed performance is weak (0% on-time in the recorded cells), which drags A's overall results if ORD were even larger; conversely, Airline A has substantial volume and an advantage in NYC (500 flights each) and ATL (A has 400 vs. B's 200), which boosts A's overall percentage despite losing to B in LAX. Missing cells that were imputed to zero (e.g., A's missing on-time at ORD; B's missing on-time at ATL) also accentuate these city swings—another reminder that the mix of city sizes and how missing data are handled can reverse or exaggerate conclusions when moving between disaggregated and aggregated views.