

2020-01-05

2021-01-05

TidyTuesday

Join the R4DS Online Learning Community in the weekly #TidyTuesday event! Every week we post a raw dataset, a chart or article related to that dataset, and ask you to explore the data. While the dataset will be “tamed”, it will not always be tidy! As such you might need to apply various R for Data Science techniques to wrangle the data into a true tidy format. The goal of TidyTuesday is to apply your R skills, get feedback, explore other’s work, and connect with the greater #RStats community! As such we encourage everyone of all skills to participate!

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.4
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(tidyuesdayR)
```

```
## Warning: package 'tidyuesdayR' was built under R version 3.6.2
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
library(dplyr)
```

```
library(tidyr)
```

```
#install.packages("countrycode")
```

Load the weekly Data

Download the weekly data and make available in the `tt` object.

```
# download the data
```

```
tt <- tt_load("2021-01-05")
```

```
## --- Compiling #TidyTuesday Information for 2021-01-05 ----
```

```
## --- There is 1 file available ---
## --- Starting Download ---
##
## Downloading file 1 of 1: `transit_cost.csv`
## --- Download complete ---
#saving the data as a variable
transit <- tt$`transit_cost`
```

Readme

Take a look at the readme for the weekly data to get insight on the dataset. This includes a data dictionary, source, and a link to an article on the data.

```
readme(tt)
print(tt)
```

Glimpse Data

Take an initial look at the format of the data available.

```
tt %>%
  map(glimpse)

## Observations: 544
## Variables: 20
## $ e                <dbl> 7136, 7137, 7138, 7139, 7144, 7145, 7146, 7147, 71...
## $ country          <chr> "CA", "CA", "CA", "CA", "CA", "NL", "CA", "US", "U...
## $ city             <chr> "Vancouver", "Toronto", "Toronto", "Toronto", "Tor...
## $ line             <chr> "Broadway", "Vaughan", "Scarborough", "Ontario", "...
## $ start_year       <chr> "2020", "2009", "2020", "2020", "2020", "2003", "2...
## $ end_year         <chr> "2025", "2017", "2030", "2030", "2030", "2018", "2...
## $ rr              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ length           <dbl> 5.7, 8.6, 7.8, 15.5, 7.4, 9.7, 5.8, 5.1, 4.2, 4.2,...
## $ tunnel_per       <chr> "87.72%", "100.00%", "100.00%", "57.00%", "100.00%...
## $ tunnel           <dbl> 5.0, 8.6, 7.8, 8.8, 7.4, 7.1, 5.8, 5.1, 4.2, 4.2, ...
## $ stations         <dbl> 6, 6, 3, 15, 6, 8, 5, 2, 2, 2, 3, 3, 4, 7, 13, 4, ...
## $ source1          <chr> "Plan", "Media", "Wiki", "Plan", "Plan", "Wiki", "...
## $ cost             <dbl> 2830, 3200, 5500, 8573, 5600, 3100, 4500, 1756, 36...
## $ currency         <chr> "CAD", "CAD", "CAD", "CAD", "CAD", "EUR", "CAD", "...
## $ year             <dbl> 2018, 2013, 2018, 2019, 2020, 2009, 2018, 2012, 20...
## $ ppp_rate         <dbl> 0.840, 0.810, 0.840, 0.840, 0.840, 1.300, 0.840, 1...
## $ real_cost        <chr> "2377.2", "2592", "4620", "7201.32", "4704", "4030...
## $ cost_km_millions <dbl> 417.05263, 301.39535, 592.30769, 464.60129, 635.67...
## $ source2          <chr> "Media", "Media", "Media", "Plan", "Media", "Media...
## $ reference        <chr> "https://www.translink.ca/Plans-and-Projects/Rapid...

## $transit_cost
## # A tibble: 544 x 20
##       e country city line start_year end_year rr length tunnel_per tunnel
##   <dbl> <chr>  <chr> <chr> <chr>      <chr>  <dbl>  <dbl> <chr>      <dbl>
## 1  7136 CA     Vanc~ Broa~ 2020      2025    0    5.7 87.72%      5
## 2  7137 CA     Toro~ Vaug~ 2009      2017    0    8.6 100.00%     8.6
```

```
## 3 7138 CA Toro~ Scar~ 2020 2030 0 7.8 100.00% 7.8
## 4 7139 CA Toro~ Onta~ 2020 2030 0 15.5 57.00% 8.8
## 5 7144 CA Toro~ Yong~ 2020 2030 0 7.4 100.00% 7.4
## 6 7145 NL Amst~ Nort~ 2003 2018 0 9.7 73.00% 7.1
## 7 7146 CA Mont~ Blue~ 2020 2026 0 5.8 100.00% 5.8
## 8 7147 US Seat~ U-Li~ 2009 2016 0 5.1 100.00% 5.1
## 9 7152 US Los ~ Purp~ 2020 2027 0 4.2 100.00% 4.2
## 10 7153 US Los ~ Purp~ 2018 2026 0 4.2 100.00% 4.2
## # ... with 534 more rows, and 10 more variables: stations <dbl>, source1 <chr>,
## # cost <dbl>, currency <chr>, year <dbl>, ppp_rate <dbl>, real_cost <chr>,
## # cost_km_millions <dbl>, source2 <chr>, reference <chr>
```

```
head(transit)
```

```
## # A tibble: 6 x 20
##       e country city line start_year end_year rr length tunnel_per tunnel
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <chr> <dbl>
## 1 7136 CA Vanc~ Broa~ 2020 2025 0 5.7 87.72% 5
## 2 7137 CA Toro~ Vaug~ 2009 2017 0 8.6 100.00% 8.6
## 3 7138 CA Toro~ Scar~ 2020 2030 0 7.8 100.00% 7.8
## 4 7139 CA Toro~ Onta~ 2020 2030 0 15.5 57.00% 8.8
## 5 7144 CA Toro~ Yong~ 2020 2030 0 7.4 100.00% 7.4
## 6 7145 NL Amst~ Nort~ 2003 2018 0 9.7 73.00% 7.1
## # ... with 10 more variables: stations <dbl>, source1 <chr>, cost <dbl>,
## # currency <chr>, year <dbl>, ppp_rate <dbl>, real_cost <chr>,
## # cost_km_millions <dbl>, source2 <chr>, reference <chr>
```

Wrangle

Explore the data and process it into a nice format for plotting! Access each dataset by name by using a dollarsign after the `tt` object and then the name of the data set.

```
# write the data to a csv file
write.csv(transit, "transit_cost.csv", )

transit <- readr::read_csv("transit_cost.csv") %>%
  mutate(real_cost = as.numeric(real_cost), start_year = as.numeric(start_year)) %>%
  filter(!is.na(line)) # %>%
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
```

```
## cols(
##   .default = col_character(),
##   X1 = col_double(),
##   e = col_double(),
##   rr = col_double(),
##   length = col_double(),
##   tunnel = col_double(),
##   stations = col_double(),
##   cost = col_double(),
##   year = col_double(),
##   ppp_rate = col_double(),
##   cost_km_millions = col_double()
## )
```

```
## See spec(...) for full column specifications.
## Warning: NAs introduced by coercion
## Warning: NAs introduced by coercion

# mutate(region = countrycode(country, origin = "ecb", destination = "region")) %>%
# mutate(region = case_when(country == "UK" ~ "Europe & Central Asia", TRUE ~ region))

head(transit)

## # A tibble: 6 x 21
##       X1      e country city line start_year end_year rr length tunnel_per
##   <dbl> <dbl> <chr>   <chr> <chr>      <dbl> <chr>      <dbl> <dbl> <chr>
## 1     1    7136 CA      Vanc~ Broa~      2020 2025         0    5.7 87.72%
## 2     2    7137 CA      Toro~ Vaug~      2009 2017         0    8.6 100.00%
## 3     3    7138 CA      Toro~ Scar~      2020 2030         0    7.8 100.00%
## 4     4    7139 CA      Toro~ Onta~      2020 2030         0   15.5 57.00%
## 5     5    7144 CA      Toro~ Yong~      2020 2030         0    7.4 100.00%
## 6     6    7145 NL      Amst~ Nort~      2003 2018         0    9.7 73.00%
## # ... with 11 more variables: tunnel <dbl>, stations <dbl>, source1 <chr>,
## #   cost <dbl>, currency <chr>, year <dbl>, ppp_rate <dbl>, real_cost <dbl>,
## #   cost_km_millions <dbl>, source2 <chr>, reference <chr>

tail(transit)

## # A tibble: 6 x 21
##       X1      e country city line start_year end_year rr length tunnel_per
##   <dbl> <dbl> <chr>   <chr> <chr>      <dbl> <chr>      <dbl> <dbl> <chr>
## 1    532    9507 TR      Ista~ M5 P~      2016 2022         0   17.8 100.00%
## 2    533    9508 TR      Ista~ M12      2017 2022         0    13 100.00%
## 3    534    9509 TR      Ista~ M11 ~      2016 2021         0   37.5 100.00%
## 4    535    9510 TR      Ista~ M11 ~      2019 2022         0    32 100.00%
## 5    536    9459 UZ      Tash~ Serg~      2017 2020         0    7.1 0.00%
## 6    537    9460 UZ      Tash~ Yunu~      2017 2020         0    2.9 100.00%
## # ... with 11 more variables: tunnel <dbl>, stations <dbl>, source1 <chr>,
## #   cost <dbl>, currency <chr>, year <dbl>, ppp_rate <dbl>, real_cost <dbl>,
## #   cost_km_millions <dbl>, source2 <chr>, reference <chr>

transit %>%
  group_by(country) %>%
  summarize(mean(real_cost))

## # A tibble: 56 x 2
##   country `mean(real_cost)`
##   <chr>      <dbl>
## 1 AE          6637.
## 2 AR          4646
## 3 AT          1352
## 4 AU          6238.
## 5 BD        12352.
## 6 BE          1170
## 7 BG          1016.
## 8 BH          4882.
## 9 BR          3665
## 10 CA         3283.
## # ... with 46 more rows
```

```
# looking at the average real cost by country
tapply(transit$real_cost, transit$country, mean)
```

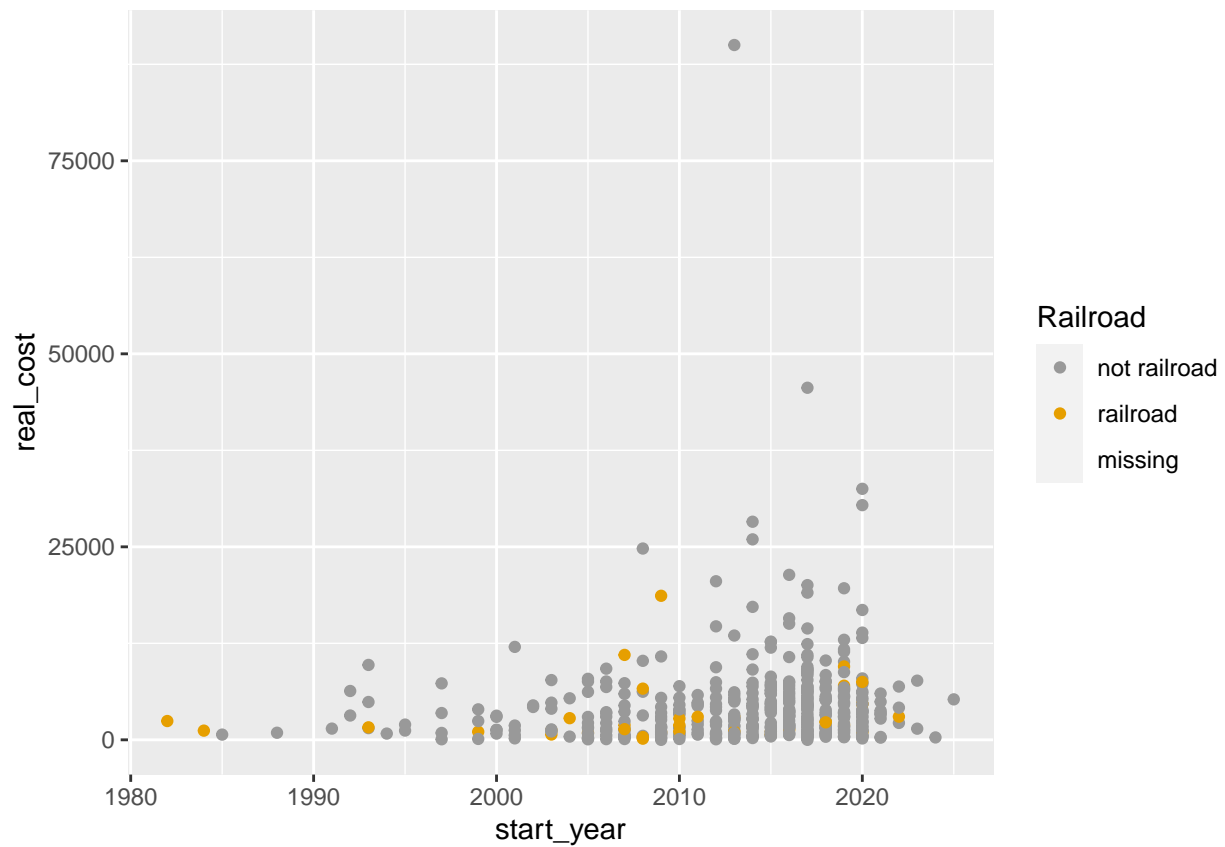
```
##      AE      AR      AT      AU      BD      BE      BG      BH
## 6636.667 4646.000 1352.000 6237.600 12351.893 1170.000 1016.295 4882.500
##      BR      CA      CH      CL      CN      CZ      DE      DK
## 3665.000 3282.997 865.232 5015.000 4240.633 1519.040 733.160 3491.400
##      EC      EG      ES      FI      FR      GR      HU      ID
## 3819.000 5784.643 1357.250 1273.870 3868.451 1218.425 3579.840 2934.303
##      IL      IN      IR      IT      JP      KR      KW      MX
## 5038.400 6753.120 4640.000 971.415 2076.532 2384.063 30400.000 4783.390
##      MY      NL      NO      NZ      PA      PE      PH      PK
## 18035.500 4030.000 871.155 2991.663 4330.507 11088.400 8338.950 6039.000
##      PL      PT      QA      RO      RU      SA      SE      SG
## 1340.817 340.600 90000.000 1860.712 5095.583 13545.280 1072.340 19503.500
##      TH      TR      TW      UA      UK      US      UZ      VN
## 5834.779 1891.235 4977.106 2738.318 8441.067 4377.769 667.500 4756.626
```

Visualize

Using your processed dataset, create your unique visualization.

```
transit %>%
  ggplot(aes(x = start_year, y = real_cost, color = as.factor(rr)) ) +
  geom_point() +
  scale_color_manual(values = c("#999999", "#E69F00", "#569BBD"),
                     name = "Railroad",
                     breaks = c(0, 1, NA),
                     labels = c("not railroad", "railroad", "missing"))
```

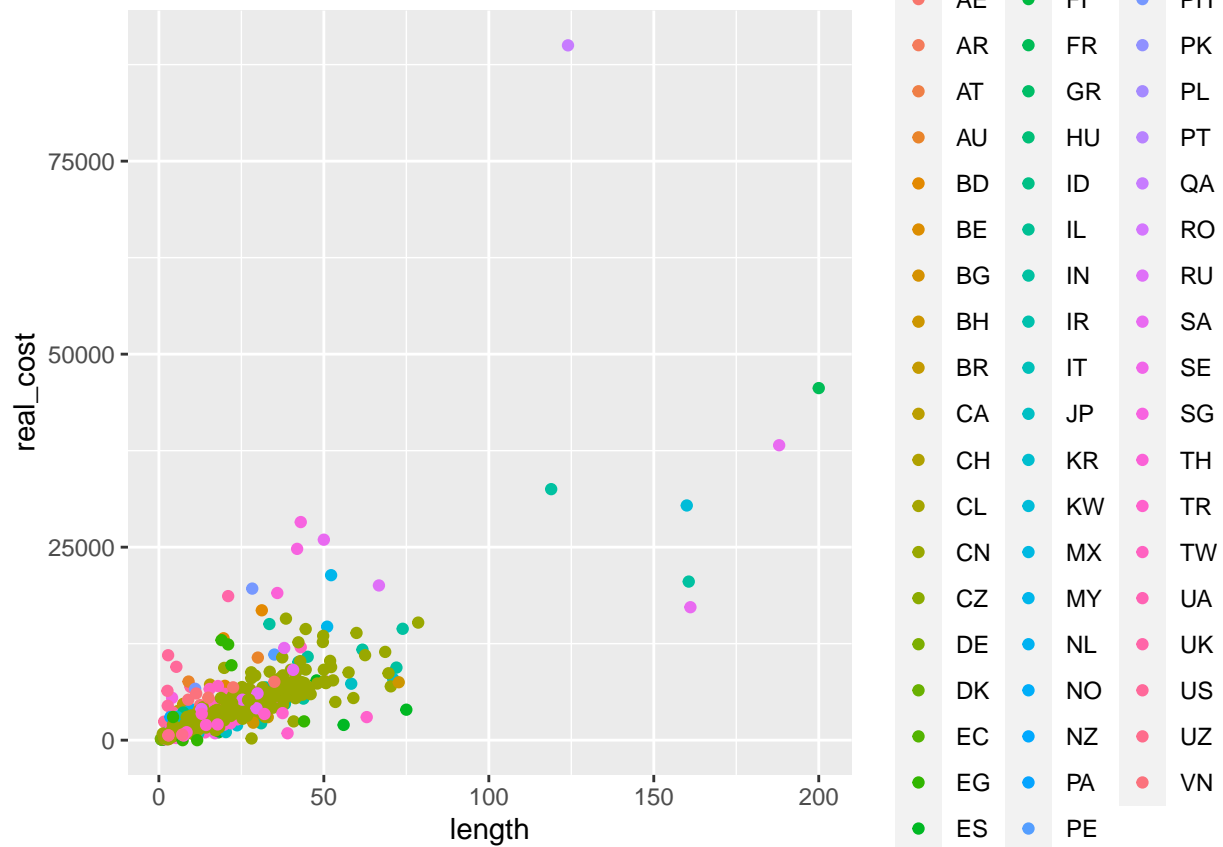
```
## Warning: Removed 61 rows containing missing values (geom_point).
```



```
# non-railroad vs railroad counts
table(transit$rr)
```

```
##
##    0    1
## 502   34
```

```
# figure out how to replace 'country' with 'region' by using the 'countrycode' package
transit %>%
  ggplot(aes(x = length, y = real_cost, color = country)) +
  geom_point()
```



```
# run this once we figure out the conuntrycode package!!!
#transit %>%
  #ggplot(aes(y = real_cost, x = region, color = region)) +
  #geom_boxplot()
```

```
library(knitr)
#knit('2021_01_05_tidy_tuesday.Rmd', encoding = 'UTF-8')
```