Homework 1

Aaron Melcher

2024-09-30

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

#[NYC Flights]

library(nycflights13)  
flights |> select(carrier, air\_time, distance)

## # A tibble: 336,776 × 3  
## carrier air\_time distance  
## <chr> <dbl> <dbl>  
## 1 UA 227 1400  
## 2 UA 227 1416  
## 3 AA 160 1089  
## 4 B6 183 1576  
## 5 DL 116 762  
## 6 UA 150 719  
## 7 B6 158 1065  
## 8 EV 53 229  
## 9 B6 140 944  
## 10 AA 138 733  
## # ℹ 336,766 more rows

##Part A

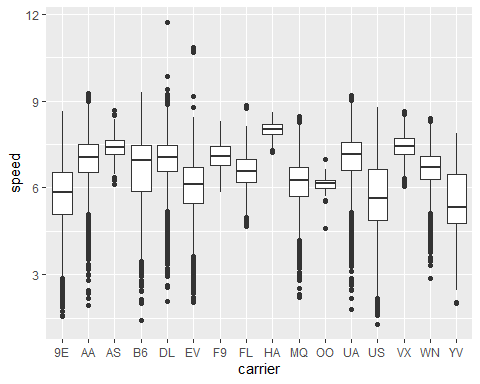
flights <- mutate(flights,  
 speed = distance / air\_time  
)  
flights

## # A tibble: 336,776 × 20  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int> <int>  
## 1 2013 1 1 517 515 2 830 819  
## 2 2013 1 1 533 529 4 850 830  
## 3 2013 1 1 542 540 2 923 850  
## 4 2013 1 1 544 545 -1 1004 1022  
## 5 2013 1 1 554 600 -6 812 837  
## 6 2013 1 1 554 558 -4 740 728  
## 7 2013 1 1 555 600 -5 913 854  
## 8 2013 1 1 557 600 -3 709 723  
## 9 2013 1 1 557 600 -3 838 846  
## 10 2013 1 1 558 600 -2 753 745  
## # ℹ 336,766 more rows  
## # ℹ 12 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
## # tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
## # hour <dbl>, minute <dbl>, time\_hour <dttm>, speed <dbl>

##Part B

ggplot(flights, aes(carrier, speed)) +  
 geom\_boxplot()

## Warning: Removed 9430 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



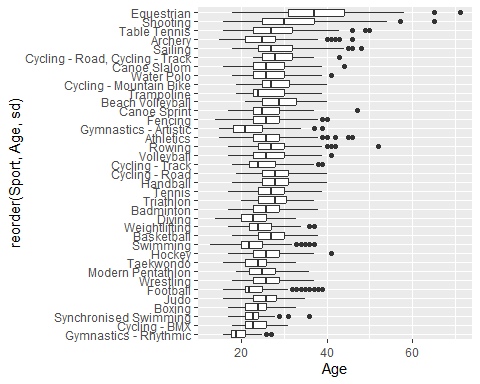
The boxplot above provides a very basic analysis of the average speed of each carrier. With this, we are able to see carrier ‘HA’ has the highest average speed of any of the carriers. Even though some of the carriers do have higher ‘highs’, HA still on average is faster than the others.

#[London Olympics]

olympics <- read\_csv("https://uwmadison.box.com/shared/static/rzw8h2x6dp5693gdbpgxaf2koqijo12l.csv")

##Part A/B

counts <- olympics |>  
 count(Sport)  
  
olympics |>  
 left\_join(counts) |>  
 filter(n > 5) |>  
 ggplot() +  
 geom\_boxplot(  
 aes(Age, reorder(Sport, Age, sd))  
 )

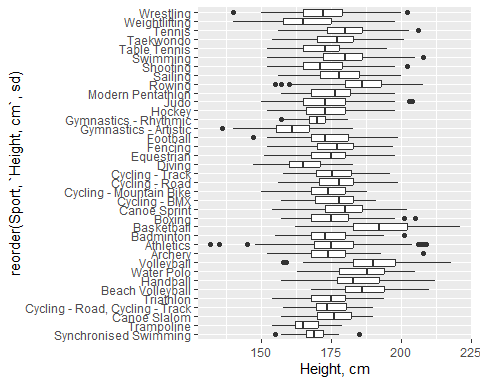


##Part C

Average height(cm) among sports. See if there’s a noticeable difference in the height among athletes that participate in different sports. Following the same idea as the age distribution, I created a visualization on based on height.

olympics |>  
 left\_join(counts) |>  
 filter(n > 5) |>  
 ggplot() +  
 geom\_boxplot(  
 aes(`Height, cm`, reorder(Sport, `Height, cm`, sd))  
 )

## Warning: Removed 561 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



There does not seem to be a very noticeable difference in height, except in basketball, which does make sense.

#[Pokemon]

pokemon <- read\_csv("https://uwmadison.box.com/shared/static/hf5cmx3ew3ch0v6t0c2x56838er1lt2c.csv")

##Part A

pokemon <- mutate(  
 pokemon,  
 'attack-to-defense' = Attack / Defense  
)  
pokemon |> select('attack-to-defense')

## # A tibble: 800 × 1  
## `attack-to-defense`  
## <dbl>  
## 1 1   
## 2 0.984  
## 3 0.988  
## 4 0.813  
## 5 1.21   
## 6 1.10   
## 7 1.08   
## 8 1.17   
## 9 1.33   
## 10 0.738  
## # ℹ 790 more rows

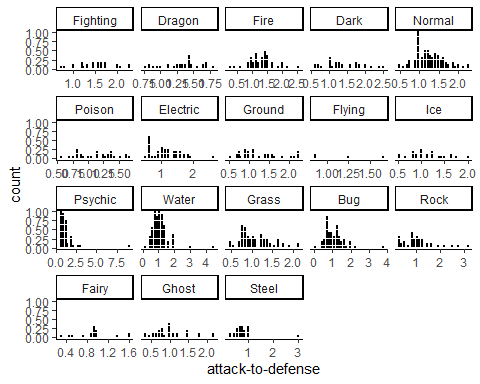
##Part B

pokemon <- pokemon |>  
 group\_by(type\_1) |>  
 mutate(`attack-to-defense-median` = median(`attack-to-defense`, na.rm = TRUE)) |>  
 ungroup()  
pokemon

## # A tibble: 800 × 14  
## Name type\_1 type\_2 Total HP Attack Defense speed\_attack speed\_defense  
## <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Bulbasaur Grass Poison 318 45 49 49 65 65  
## 2 Ivysaur Grass Poison 405 60 62 63 80 80  
## 3 Venusaur Grass Poison 525 80 82 83 100 100  
## 4 Venusaur… Grass Poison 625 80 100 123 122 120  
## 5 Charmand… Fire <NA> 309 39 52 43 60 50  
## 6 Charmele… Fire <NA> 405 58 64 58 80 65  
## 7 Charizard Fire Flying 534 78 84 78 109 85  
## 8 Charizar… Fire Dragon 634 78 130 111 130 85  
## 9 Charizar… Fire Flying 634 78 104 78 159 115  
## 10 Squirtle Water <NA> 314 44 48 65 50 64  
## # ℹ 790 more rows  
## # ℹ 5 more variables: Speed <dbl>, Generation <dbl>, Legendary <lgl>,  
## # `attack-to-defense` <dbl>, `attack-to-defense-median` <dbl>

##Part C

pokemon$type\_1 <- reorder(pokemon$type\_1, desc(pokemon$`attack-to-defense-median`))  
  
ggplot(  
 pokemon,  
 aes(`attack-to-defense`)  
 ) +  
 geom\_dotplot() +  
 facet\_wrap("type\_1", scales = "free\_x") +  
 theme\_classic()



##Part D

An example dynamic plot could compare the stat totals of Pokemon based on different factors. For example, we could see the difference based on type or legendary status. This would answer how strong some types are or how strong legendary Pokemon are compared to regular Pokemon. The user would be able to select what facet they want to view and the query would update how the stat totals are shown.

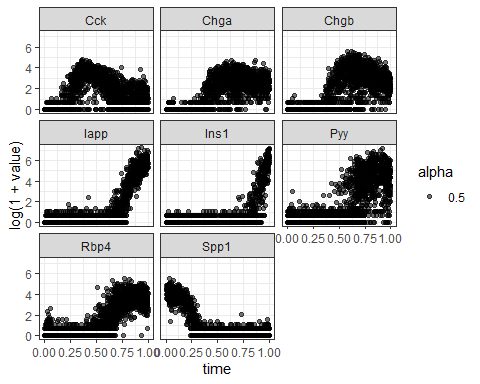
#[Gene Expression Faceting]

genes <- read\_csv("https://uwmadison.box.com/shared/static/dwzchdtfca33r0f6i055k2d0939onnlv.csv")  
head(genes, 3)

## # A tibble: 3 × 4  
## sample gene time value  
## <dbl> <chr> <dbl> <dbl>  
## 1 1 Pyy 0.677 103  
## 2 1 Iapp 0.677 0  
## 3 1 Chgb 0.677 86

##Part A

genes <- mutate(genes,  
 'log(1 + value)' = log(1 + value)  
 )  
  
ggplot(genes, aes(  
 x = time,  
 y = `log(1 + value)`,  
 alpha = 0.5  
)) +  
 geom\_point() +  
 facet\_wrap(~ gene) +  
 theme\_bw()



##Part B One advantage of small multiples is being able to compare multiple graphs one the same scale. This allows for comparisons between different facets such as shown in the graph above.

One disadvantage of small multiples is trying to display large amounts of data since too many graphs can be overwhelming and unweildy with lots of different information.

##Part C

gene\_groups <- genes |>  
group\_by(gene, rounded\_time = round(time, 2)) |>  
summarise(mean\_value = mean(value))  
head(gene\_groups, 3)

## # A tibble: 3 × 3  
## # Groups: gene [1]  
## gene rounded\_time mean\_value  
## <chr> <dbl> <dbl>  
## 1 Cck 0 0   
## 2 Cck 0.01 0   
## 3 Cck 0.02 0.0652

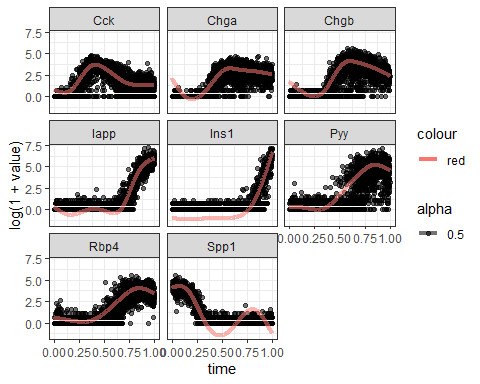
##Part D

fitted\_values <- read\_csv("https://go.wisc.edu/x678hu")  
head(fitted\_values, 3)

## # A tibble: 3 × 4  
## gene time mu sigma  
## <chr> <dbl> <dbl> <dbl>  
## 1 Pyy 0.677 52.8 3.17  
## 2 Pyy 0.432 2.47 1.23  
## 3 Pyy 0.749 112. 1.95

ggplot(genes, aes(  
 x = time,  
 y = `log(1 + value)`,  
 alpha = 0.5  
)) +  
 geom\_point() +  
 geom\_line(  
 data = fitted\_values,  
 aes(  
 x = time,  
 y = log(mu),  
 color = 'red'  
 ),  
 size = 1.5  
 ) +  
 facet\_wrap(~ gene) +  
 theme\_bw()

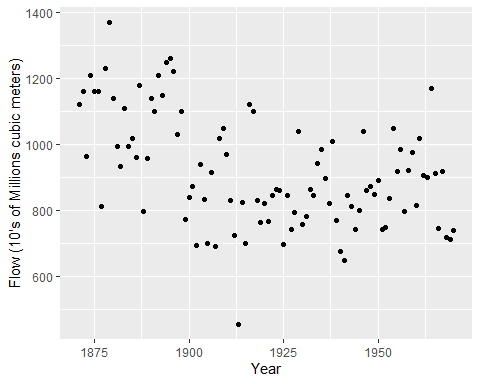
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



#[Visual Redesign]

##Part A Here is the code for a very basic visualization that was done in a past Stats class:

n\_df = read.csv("nile.csv")  
  
ggplot(n\_df) +  
 geom\_point(aes(  
 x = Year,  
 y = Flow  
 )) +  
 xlab("Year") +  
 ylab("Flow (10's of Millions cubic meters)")

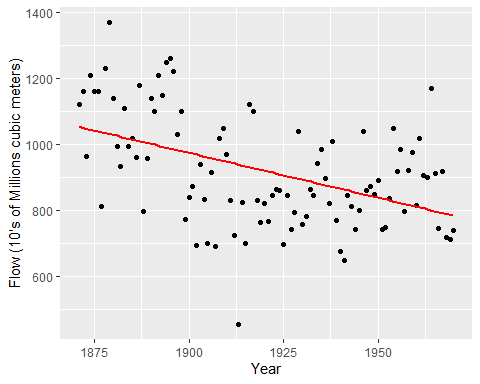


##Part B The main takeaways from the above graph are the overall decline in water flow over the time period. This is somewhat consistent from the visualization as there really isn’t an intended message and the interpretation of the data is left up to the viewer.

##Part C The visualization is legible but hard to draw any ideas from as there is any additional visualization provided to help highlight more information. The dataset itself isn’t super in-depth but additional visual elements can help with providing more in-depth analysis.

##Part D Here is a new visualization that uses the original design but adds a basic overlay to better highlight the information I want the viewer to know. With the limited data, there isn’t too much more that can be visualized.

ggplot(n\_df,   
 aes(  
 x = Year,   
 y = Flow)  
 ) +  
 geom\_point() +  
 geom\_smooth(  
 method = "lm",   
 se = FALSE,   
 color = "red",   
 size = 1  
 ) +  
 xlab("Year") +  
 ylab("Flow (10's of Millions cubic meters)")



#[California Wildfire Alternatives]

fires <- read\_csv("https://uwmadison.box.com/shared/static/k5vvekf1bhh9e16qb9s66owygc70t7dm.csv") |>  
 select(Name, Counties, year, day\_of\_year, AcresBurned, MajorIncident)  
head(fires, 3)

## # A tibble: 3 × 6  
## Name Counties year day\_of\_year AcresBurned MajorIncident  
## <chr> <chr> <dbl> <dbl> <dbl> <lgl>   
## 1 Becks Fire Lake 2013 22 296 FALSE   
## 2 River Fire Inyo 2013 55 406 TRUE   
## 3 Jurupa Fire Riverside 2013 59 311 FALSE

##Part A - This visualization approach is well suited for for comparing the distribution of fires throughout the dates in each year shown. We can see somewhat easily when each fire happened during each year.

* This visualization is not well suited for seeing which counties had the most fire damage. Based on how the graphs are structured, we cannot easily tell which counties had the most fires. The counties are arranged in a way that is hard to read the names as well as easily find the exact number of fires that occurred.

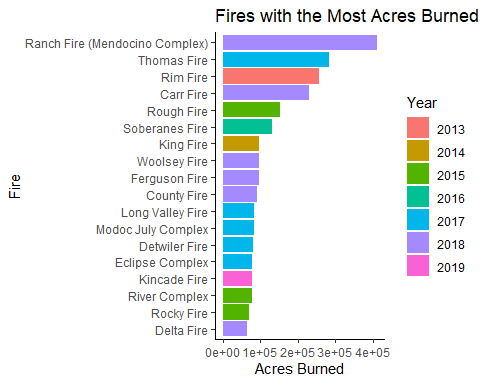
##Part B - This visualization is good for comparing the damage between the major fires from each year. We can easily see the range of acres burned by each type and are easily able to compare the damage done by each type over each year.

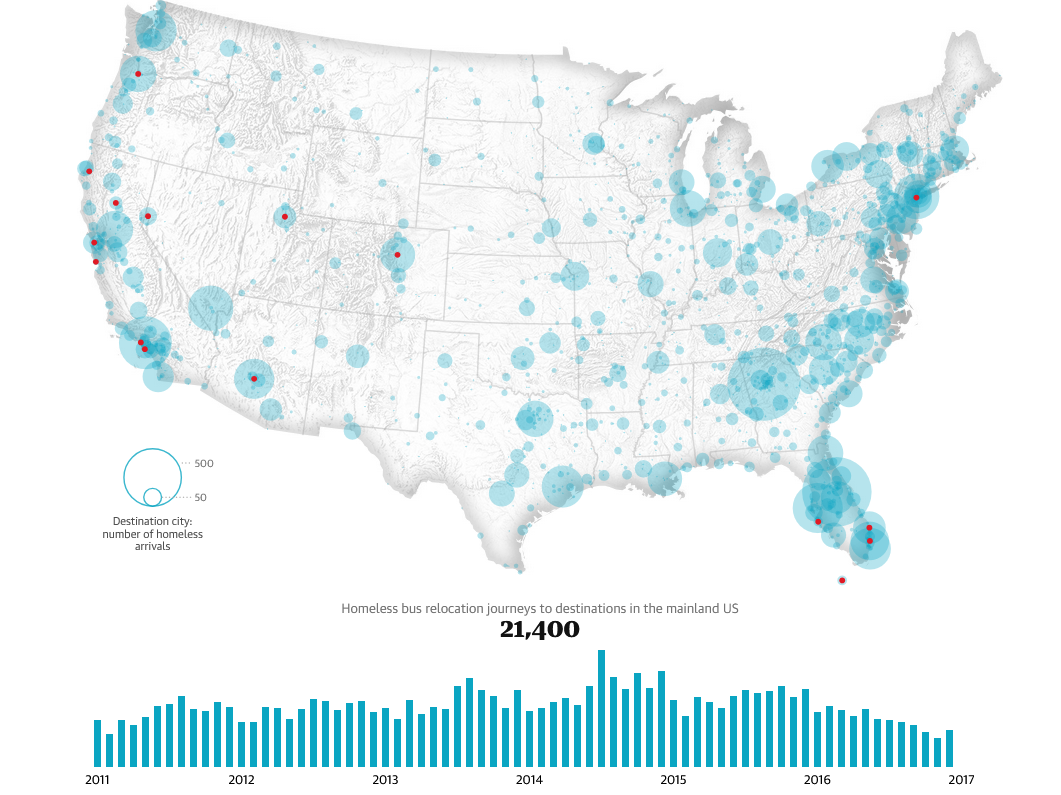
* Again this visualization is not good for comparisons done between areas burned. Since we are not given any information on the area in which each fire occurred we don’t know any of this information and thus cannot visually compare it.

##Part C - This visualization is good for comparing the damage the most damaging fires has done. We can easily see the name of the fire and also how much damage the fire did as well as the year it was in. The damage done by acre is clearly illustrated by the boxplot.

##Part D

fires$Name <- reorder(fires$Name, fires$AcresBurned)  
top\_fires <- fires |>  
 top\_n(18, AcresBurned)  
  
  
ggplot(top\_fires, aes(AcresBurned, Name, fill=factor(year))) +  
 geom\_bar(stat = 'identity') +  
 labs(title = "Fires with the Most Acres Burned", x = "Acres Burned", y = "Fire", fill = "Year") +  
 theme\_classic()



#[Homelessness]  ##Part A The data behind the graph would be the homeless bus relocations in the mainland United States over the period from 2011 to 2017. The rows would the year and the columns would the total number of homeless bus relocations to each city.

##Part B Both columns would be numeric types as we are tracking quantitative data and can count the total number of bus relocations.

##Part C The data would be encoded by city since each city is given a ‘circle’ mark to indicate how many people relocated to the city via bus during the time period. The marks would be derived from the number of relocations and the circles follow a exponential scale of the total number of relocations.

##Part D Yes since we are given a map of the US to show visually what cities were the most effected as well as a bar graph that gives a general overview of the change in number of bus relocations over the time period.