# Project2

Joyce Aldrich

2022-10-09

## example 1. loading csv into r

```
library(readr)
city_weekly_temperature_wide <-read_csv(file = '/Users/joycealdrich/Documents/SPS Data Science/Data 607/
```

```
## Rows: 6 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): city
## dbl (7): sun, mon, tue, wed, thr, fri, sat
##
## 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.
```

```
head(city_weekly_temperature_wide)
```

```
## # A tibble: 6 x 8
##   city       sun   mon   tue   wed   thr   fri   sat
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 houston     78    89    90    85    79    83    94
## 2 boston      88    82    93    87    79    82    89
## 3 seattle     64    67    69    61    70    73    65
## 4 new york    88    93    90    89    86    83    92
## 5 chicago     67    63    66    70    71    73    68
## 6 orlando     90    92    89    86    91    87    93
```

## reshaping data to long format

```
library(tidyr)
city_weekly_temperature_long <- city_weekly_temperature_wide %>%
gather(day,temperature, -c(city))
head(city_weekly_temperature_long)
```

```
## # A tibble: 6 x 3
##   city     day   temperature
##   <chr>    <chr>       <dbl>
## 1 houston  sun            78
## 2 boston   sun            88
## 3 seattle  sun            64
## 4 new york sun            88
## 5 chicago  sun            67
## 6 orlando  sun            90
```

## adding the average temperature for each city

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

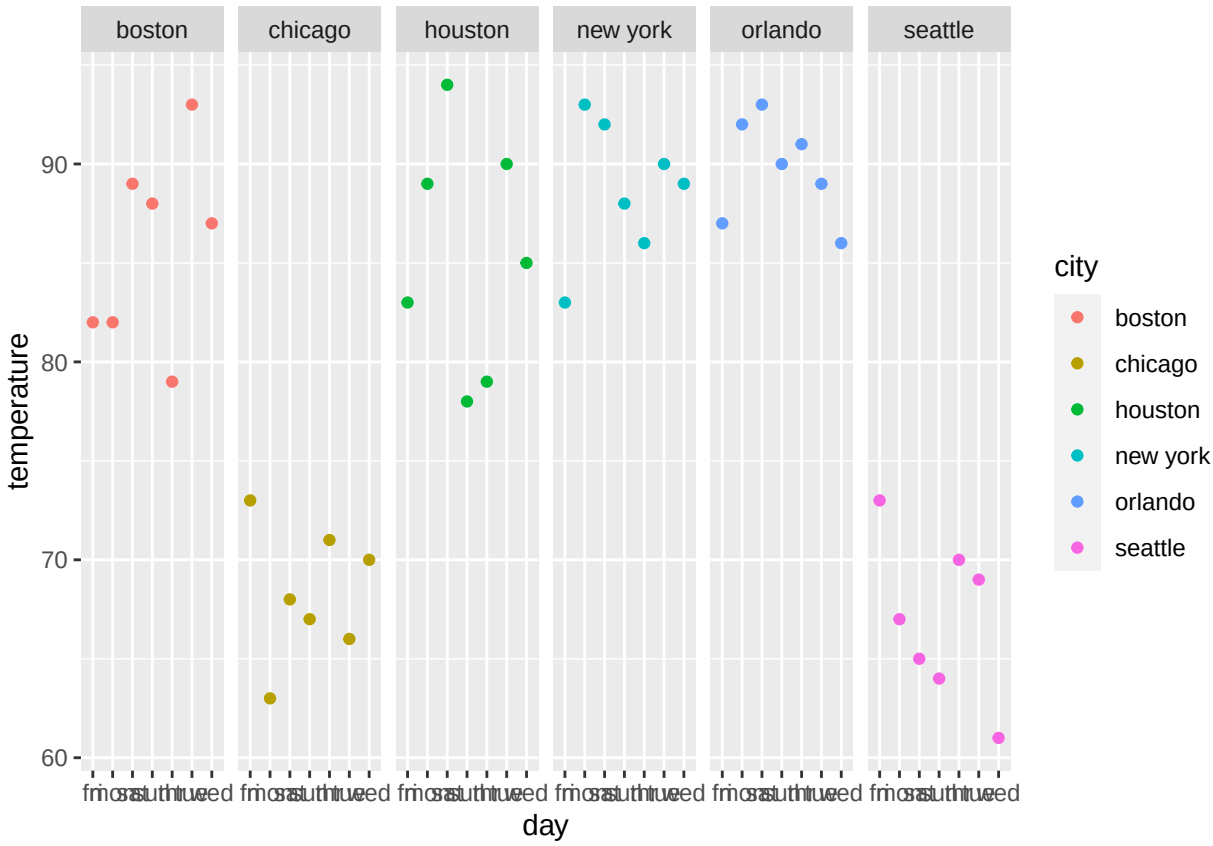
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
city_weekly_temperature_long <- city_weekly_temperature_long %>%
  group_by(city) %>%
  mutate(average_temperature = mean(temperature),
         temperature_diff = temperature - mean(temperature)
         )
head(city_weekly_temperature_long)
```

```
## # A tibble: 6 x 5
## # Groups:   city [6]
##   city     day   temperature average_temperature temperature_diff
##   <chr>    <chr>       <dbl>               <dbl>            <dbl>
## 1 houston  sun            78                85.4           -7.43
## 2 boston   sun            88                85.7            2.29
## 3 seattle  sun            64                67             -3
## 4 new york sun            88                88.7           -0.714
## 5 chicago  sun            67                68.3           -1.29
## 6 orlando  sun            90                89.7            0.286
```

## using ggplot geom_point to see each city's weekly temperature

```
library(ggplot2)
ggplot(data= city_weekly_temperature_long ) +
  geom_point(mapping = aes(x=day, y=temperature, color=city)) +
  facet_wrap(~city, nrow=1)
```



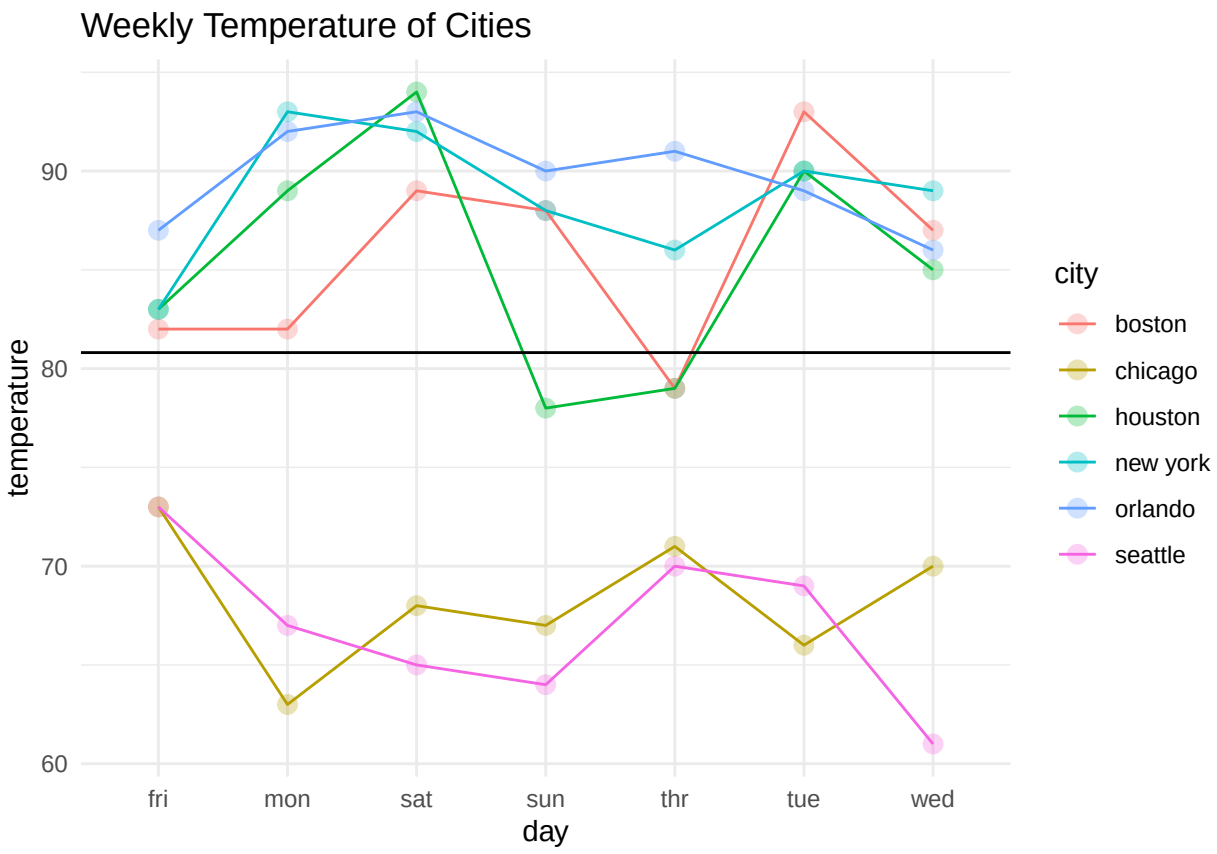

## compare 6 cities' weekly temperature and place the overall mean in the graph

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v stringr 1.4.1
## v purrr   0.3.4     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

city_weekly_temperature_long %>%
  ggplot(aes(x=day, y=temperature,group=city,colour=city))+
  geom_point(size =3, alpha=0.3)+
  geom_line(size =0.5)+
  geom_hline(aes(yintercept=mean(temperature)))+
  theme_minimal()+
  labs(title ="Weekly Temperature of Cities")
```




## example 2:

## loading cvs file into r

```
library(readr)
exam_results <-read_csv(file = '/Users/joycealdrich/Documents/SPS Data Science/Data 607/Project_2/exam_
```

```
## Rows: 6 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Student, Gender
## dbl (3): Test 1, Test 2, Test 3
##
## 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.
```

```
head(exam_results)
```

```
## # A tibble: 6 x 5
##   Student Gender `Test 1` `Test 2` `Test 3`
##   <chr>   <chr>     <dbl>    <dbl>    <dbl>
## 1 Amanda  F            88       78       92
## 2 Brenda  F            67       97       85
## 3 Cindy   F            79       84       88
## 4 Daniel  M            95       84       82
## 5 Eric    M            64       NA       80
```

```
## 6 Frank   M            45       65       NA
```

## reshaping to the long format

```
library(tidyr)
exam_results_long<- exam_results %>%
  gather(exam,score, -c(Student,Gender))
head(exam_results_long)
```

```
## # A tibble: 6 x 4
##   Student Gender exam   score
##   <chr>   <chr>  <chr>  <dbl>
## 1 Amanda  F      Test 1    88
## 2 Brenda  F      Test 1    67
## 3 Cindy   F      Test 1    79
## 4 Daniel  M      Test 1    95
## 5 Eric    M      Test 1    64
## 6 Frank   M      Test 1    45
```

## adding score_average_by_student col & removing the na row

```
exam_results_long <- exam_results_long %>%
  group_by(Student) %>%
  filter(!is.na(score)) %>%
  mutate(Score_average_by_student = mean(score),
         )
head(exam_results_long)
```

```
## # A tibble: 6 x 5
## # Groups:   Student [6]
##   Student Gender exam   score Score_average_by_student
##   <chr>   <chr>  <chr>  <dbl>                     <dbl>
## 1 Amanda  F      Test 1    88                      86
## 2 Brenda  F      Test 1    67                      83
## 3 Cindy   F      Test 1    79                      83.7
## 4 Daniel  M      Test 1    95                      87
## 5 Eric    M      Test 1    64                      72
## 6 Frank   M      Test 1    45                      55
```

## adding max_socre, min_score ,core_average_by_exam col

```
exam_results_long <- exam_results_long %>%
  group_by(exam) %>%
  filter(!is.na(score)) %>%
  mutate(Max_score_by_exam = max(score),
         Min_score_by_exam = min(score),
         average_score_by_exam = mean (score)
         )
head(exam_results_long)
```

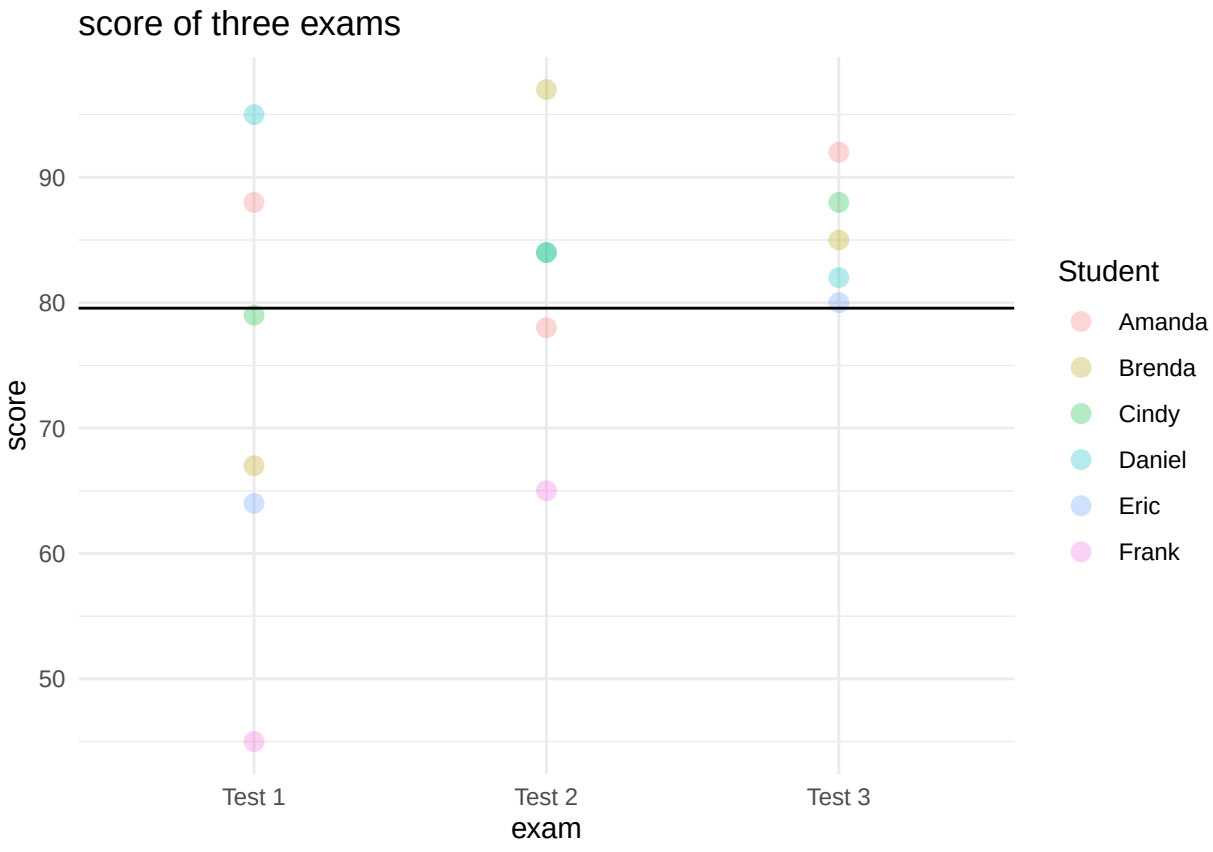
```
## # A tibble: 6 x 8
```

```
## # Groups:   exam [1]
##   Student Gender exam   score Score_average_by_student Max_sco~1 Min_s~2 avera~3
##   <chr>   <chr>  <chr>  <dbl>                    <dbl>     <dbl>   <dbl>   <dbl>
## 1 Amanda  F      Test 1    88                     86          95      45      73
## 2 Brenda  F      Test 1    67                     83          95      45      73
## 3 Cindy   F      Test 1    79                     83.7        95      45      73
## 4 Daniel  M      Test 1    95                     87          95      45      73
## 5 Eric    M      Test 1    64                     72          95      45      73
## 6 Frank   M      Test 1    45                     55          95      45      73
## # ... with abbreviated variable names 1: Max_score_by_exam,
## #   2: Min_score_by_exam, 3: average_score_by_exam
```

## compare each exam score by each student and place an overall mean in the graph

```
library(tidyverse)
library(ggplot2)

exam_results_long %>%
  ggplot(aes(x=exam, y=score, group=Student, colour=Student))+
  geom_point(size =3, alpha=0.3)+
  geom_hline(aes(yintercept=mean(score)))+
  theme_minimal()+
  labs(title ="score of three exams")
```



## example 3:

## loading excel(format download) file into r

```
library(readxl)
gdp_by_county <- read_excel("gdp_by_county.xlsx")
```

```
## New names:
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
```

```
head(gdp_by_county)
```

```
## # A tibble: 6 x 9
##   FIPS  Countyname Postal LineCode IndustryName        Gross~1 ...7  ...8  ...9
##   <chr> <chr>      <chr>     <dbl> <chr>                <chr>   <chr> <chr> <chr>
## 1 <NA>  <NA>       <NA>         NA <NA>                 (thous~ <NA>  <NA>  <NA>
## 2 <NA>  <NA>       <NA>         NA <NA>                 2012    2013  2014  2015
## 3 01001 Autauga    AL            1 All Industries      1383941 1363~ 1402~ 1539~
## 4 01001 Autauga    AL            2 Private goods-prod~ 286396  3104~ 3235~ 3463~
## 5 01001 Autauga    AL            3 Private services-p~ 948490  9045~ 9284~ 1037~
## 6 01001 Autauga    AL            4 Government and gov~ 149055  1483~ 1504~ 1557~
## # ... with abbreviated variable name
## #   1: `Gross domestic product (GDP) by county`
```

## updating the col name and remove NA rows

```
names(gdp_by_county)[6]<-paste("2012")
names(gdp_by_county)[7]<-paste("2013")
names(gdp_by_county)[8]<-paste("2014")
names(gdp_by_county)[9]<-paste("2015")

gdp_by_county <- gdp_by_county[-c(1,2),]
head(gdp_by_county)
```

```
## # A tibble: 6 x 9
##   FIPS  Countyname Postal LineCode IndustryName      `2012` `2013` `2014` `2015`
##   <chr> <chr>      <chr>     <dbl> <chr>              <chr>  <chr>  <chr>  <chr>
## 1 01001 Autauga    AL            1 All Industries    13839~ 13633~ 14025~ 15394~
## 2 01001 Autauga    AL            2 Private goods-pr~ 286396 310468 323582 346355
## 3 01001 Autauga    AL            3 Private services~ 948490 904599 928438 10373~
## 4 01001 Autauga    AL            4 Government and g~ 149055 148301 150496 155742
## 5 01003 Baldwin    AL            1 All Industries    55991~ 63650~ 65473~ 64361~
## 6 01003 Baldwin    AL            2 Private goods-pr~ 681871 698500 711443 735432
```

## keeping consolidation GDP amount

```
gdp_by_county_wide<- gdp_by_county %>%
filter(LineCode == 1)
head(gdp_by_county_wide)
```

```
## # A tibble: 6 x 9
```

```
##   FIPS  Countyname Postal LineCode IndustryName   `2012`  `2013`  `2014`  `2015`
##   <chr> <chr>      <chr>     <dbl> <chr>          <chr>   <chr>   <chr>   <chr>
## 1 01001 Autauga    AL            1 All Industries 1383941 1363368 1402516 15394~
## 2 01003 Baldwin    AL            1 All Industries 5599194 6365080 6547396 64361~
## 3 01005 Barbour    AL            1 All Industries 639833  701750  689212  743779
## 4 01007 Bibb       AL            1 All Industries 297560  325906  329087  322307
## 5 01009 Blount     AL            1 All Industries 632761  701145  688525  819608
## 6 01011 Bullock    AL            1 All Industries 191052  190103  178408  178902
```

## reshaping to long data format

```
library(tidyr)
gdp_by_county_long <- gdp_by_county_wide %>%
  gather(year, amount, -c(1:5))
head(gdp_by_county_long)
```

```
## # A tibble: 6 x 7
##   FIPS  Countyname Postal LineCode IndustryName   year  amount
##   <chr> <chr>      <chr>     <dbl> <chr>          <chr> <chr>
## 1 01001 Autauga    AL            1 All Industries 2012  1383941
## 2 01003 Baldwin    AL            1 All Industries 2012  5599194
## 3 01005 Barbour    AL            1 All Industries 2012  639833
## 4 01007 Bibb       AL            1 All Industries 2012  297560
## 5 01009 Blount     AL            1 All Industries 2012  632761
## 6 01011 Bullock    AL            1 All Industries 2012  191052
```

## random select 6 sample counties in AL state

```
gdp_by_county_long_1 <- gdp_by_county_long %>%
  filter(Postal =="AL" & FIPS %in% c('01005','01011','01017','01019','01061','01063'))
```

## change class for amount var

```
gdp_by_county_long_1 <-gdp_by_county_long_1 %>%
   mutate(amount=as.integer(amount))
```

## insert the col for growth rate

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
gdp_by_county_long_1 <- gdp_by_county_long_1 %>%
  group_by(Countyname)%>%
  mutate(Growth=(amount-lag(amount))/lag(amount))
head(gdp_by_county_long_1)
```

```
## # A tibble: 6 x 8
## # Groups:   Countyname [6]
##   FIPS  Countyname Postal LineCode IndustryName   year  amount Growth
##   <chr> <chr>      <chr>     <dbl> <chr>          <chr>  <int>  <dbl>
## 1 01005 Barbour    AL            1 All Industries 2012  639833     NA
## 2 01011 Bullock    AL            1 All Industries 2012  191052     NA
## 3 01017 Chambers   AL            1 All Industries 2012  567650     NA
## 4 01019 Cherokee   AL            1 All Industries 2012  443582     NA
## 5 01061 Geneva     AL            1 All Industries 2012  406627     NA
## 6 01063 Greene     AL            1 All Industries 2012  246604     NA
```

## creating ggplot

```
library(tidyverse)
library(ggplot2)

gdp_by_county_long_1 %>%
  ggplot(aes(x=year, y=amount,color=Countyname))+
  geom_point(size =5, alpha=1)+
  geom_hline(aes(yintercept=mean(amount)))+
  theme_minimal()+
  labs(title ="Alabama- 6 Counties - 2012-2015 GDP")
```