# Homework 4

Aditi Madhok

Special Instructions: In order to do this homework, you will need the datasets `results.csv`, `grades.csv`, and `dates.csv`. Download these from Piazza and make sure they are in the same directory on your computer as your homework assignment.

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

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

1. The following command creates a tibble that gives the number of days of rainfall for five cities over three months.

```
rf1 <- read_csv2('City;January;February;March
  Atlanta, Georgia;11;10;10
  Austin, Texas;7;7;9
  Baltimore, Maryland;10;9;10
  Birmingham, Alabama;11;10;10
  Boston, Massachusetts;11;10;12')

rf1
```

```
## # A tibble: 5 x 4
##   City                  January February March
##   <chr>                   <dbl>    <dbl> <dbl>
## 1 Atlanta, Georgia           11       10    10
## 2 Austin, Texas               7        7     9
## 3 Baltimore, Maryland        10        9    10
## 4 Birmingham, Alabama        11       10    10
## 5 Boston, Massachusetts      11       10    12
```

a. Tidy this data! The resulting tibble shoud have seperate `City` and `State` columns, a `Month` column, and a `Rainfall` column. The values in the `Rainfall` column should be integers.

```
tidy_rf1<-rf1 %>%
  separate(City, into = c("City", "State"))%>%
  pivot_longer(cols=c('January':'March'), names_to = "Month", values_to = "Rainfall")%>%
  mutate(Rainfall=as.integer(Rainfall))
tidy_rf1
```

```
## # A tibble: 15 x 4
##    City       State         Month    Rainfall
##    <chr>      <chr>         <chr>       <int>
##  1 Atlanta    Georgia       January        11
##  2 Atlanta    Georgia       February       10
##  3 Atlanta    Georgia       March          10
##  4 Austin     Texas         January         7
##  5 Austin     Texas         February        7
##  6 Austin     Texas         March           9
##  7 Baltimore  Maryland      January        10
##  8 Baltimore  Maryland      February        9
##  9 Baltimore  Maryland      March          10
## 10 Birmingham Alabama       January        11
## 11 Birmingham Alabama       February       10
## 12 Birmingham Alabama       March          10
## 13 Boston     Massachusetts January        11
## 14 Boston     Massachusetts February       10
## 15 Boston     Massachusetts March          12
```

b. Create a tibble with columns City and Avg_Rainfall showing the mean number of days of rainfall over January through March for each of the five cities. (Note that this would have been very difficult without doing part a !)

```
City_Averages <-tidy_rf1 %>%
  group_by(City) %>%
  summarize(Avg_Rainfall=mean(Rainfall))
City_Averages
```

```
## # A tibble: 5 x 2
##   City       Avg_Rainfall
## * <chr>             <dbl>
## 1 Atlanta           10.3
## 2 Austin             7.67
## 3 Baltimore          9.67
## 4 Birmingham        10.3
## 5 Boston            11
```

c. In the tible tidy_rf1 that you made in part a, assume that each observation happened on the first of the month in the year 2007. Convert the Month column to a Date column, where each entry has a datatype.

```
tidy_rf1_with_dates<-tidy_rf1 %>%
  mutate(Day=1,Year=2007)%>%
  unite(col=Date,Month,Day,Year,sep="-")%>%
  mutate(Date=parse_date(Date,"%B-%d-%Y"))
tidy_rf1_with_dates
```

```
## # A tibble: 15 x 4
##    City       State         Date       Rainfall
##    <chr>      <chr>         <date>        <int>
##  1 Atlanta    Georgia       2007-01-01       11
##  2 Atlanta    Georgia       2007-02-01       10
##  3 Atlanta    Georgia       2007-03-01       10
##  4 Austin     Texas         2007-01-01        7
##  5 Austin     Texas         2007-02-01        7
##  6 Austin     Texas         2007-03-01        9
##  7 Baltimore  Maryland      2007-01-01       10
##  8 Baltimore  Maryland      2007-02-01        9
##  9 Baltimore  Maryland      2007-03-01       10
## 10 Birmingham Alabama       2007-01-01       11
## 11 Birmingham Alabama       2007-02-01       10
## 12 Birmingham Alabama       2007-03-01       10
## 13 Boston     Massachusetts 2007-01-01       11
## 14 Boston     Massachusetts 2007-02-01       10
## 15 Boston     Massachusetts 2007-03-01       12
```

---

2. Remove the `GEOID` and `moe` columns and then tidy the dataset us_rent_income. (Your tibble should have columns 'NAME', 'income', and 'rent'.) Then, create a new column called `RTI` which gives the rent-to-income ratio. Finally, sort your rows in order of increasing RTI to find out what is the most affordable state for renters.

```
tidy_rent <- us_rent_income %>%
  select(-GEOID,-moe) %>%
  pivot_wider(names_from = "variable", values_from = "estimate") %>%
  mutate (RTI=rent/income) %>%
  arrange(RTI)
tidy_rent
```

```
## # A tibble: 52 x 4
##    NAME         income  rent    RTI
##    <chr>         <dbl> <dbl>  <dbl>
##  1 North Dakota  32336   775 0.0240
##  2 South Dakota  28821   696 0.0241
##  3 Iowa          30002   740 0.0247
##  4 Nebraska      30020   773 0.0257
##  5 Wyoming       30854   828 0.0268
##  6 Wisconsin     29868   813 0.0272
##  7 Kansas        29126   801 0.0275
##  8 Minnesota     32734   906 0.0277
##  9 Ohio          27435   764 0.0278
## 10 Montana       26249   751 0.0286
## # ... with 42 more rows
```

---

3. Run the following code block to create a tibble called `race`:

```
race<-read_csv("Name,   50, 100, 150, 200, 250, 300, 350\n Carla,   1.2,   1.8,   2.2,   2.3,   3,

race
```

```
## # A tibble: 4 x 8
##   Name   `50` `100` `150` `200` `250` `300` `350`
##   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Carla   1.2   1.8   2.2   2.3   3     2.5   1.8
## 2 Mace    1.5   1.1   1.9   2     3.6   3     2.5
## 3 Lea     1.7   1.6   2.3   2.7   2.6   2.2   2.6
## 4 Karen   1.3   1.7   1.9   2.2   3.2   1.5   1.9
```

The Name column should be self-explanatory. The other column headings are lengths of time. The entries in those column are scores. Tidy this tibble! Your answer should have columns Name, Time, and Score. Entries in the Time column should be integers, and entries in the Score column should be double.

```
tidy_race<-race %>%
  pivot_longer(cols=-Name, names_to= "Time", values_to = "Score") %>%
  mutate(Time=parse_integer(Time))
tidy_race
```

```
## # A tibble: 28 x 3
##    Name   Time Score
##    <chr> <int> <dbl>
##  1 Carla    50   1.2
##  2 Carla   100   1.8
##  3 Carla   150   2.2
##  4 Carla   200   2.3
##  5 Carla   250   3
##  6 Carla   300   2.5
##  7 Carla   350   1.8
##  8 Mace     50   1.5
##  9 Mace    100   1.1
## 10 Mace    150   1.9
## # ... with 18 more rows
```

4. Run the following code block to create the results tibble.

```
results<-read_csv("results.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Individ = col_character(),
##   Treatmnt = col_character(),
##   value = col_double()
## )
```

```
results
```

```
## # A tibble: 20 x 3
##    Individ Treatmnt value
##    <chr>   <chr>    <dbl>
##  1 Ind1    Treat      1.3
##  2 Ind2    Treat      2.1
##  3 Ind3    Treat      3.2
##  4 Ind4    Treat      4.7
##  5 Ind5    Treat      5.2
##  6 Ind6    Treat      1.3
##  7 Ind7    Treat      2.4
##  8 Ind8    Treat      2.7
##  9 Ind9    Treat      3.7
## 10 Ind10   Treat      3.3
## 11 Ind1    Cont       5
## 12 Ind2    Cont       6.9
## 13 Ind3    Cont      10.1
## 14 Ind4    Cont      11.3
## 15 Ind5    Cont       2.1
## 16 Ind6    Cont       3.2
## 17 Ind7    Cont       1.1
## 18 Ind8    Cont       0.5
## 19 Ind9    Cont       9.5
## 20 Ind10   Cont       6.2
```

The `Individ` column identifies the individual participating in the experiment. The `Treatmnt` column gives the trial type ("Treat" or "Cont"). The `value` column gives the results of the experiment. Tidy this tibble! Your answer should have 3 columns, including an `Individ` column. The `Individ` column should be numbers.

```
tidy_results<-results %>%
  mutate(Individ=parse_number(Individ))%>%
  pivot_wider(names_from = "Treatmnt",values_from = "value")
tidy_results
```

```
## # A tibble: 10 x 3
##    Individ Treat  Cont
##      <dbl> <dbl> <dbl>
##  1       1   1.3   5
##  2       2   2.1   6.9
##  3       3   3.2  10.1
##  4       4   4.7  11.3
##  5       5   5.2   2.1
##  6       6   1.3   3.2
##  7       7   2.4   1.1
##  8       8   2.7   0.5
##  9       9   3.7   9.5
## 10      10   3.3   6.2
```

5. Run the following code block to create the `grades` tibble.

```
grades<-read_csv("grades.csv")
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   ‘ID Test‘ = col_character(),
##   Year = col_double(),
##   Fall = col_double(),
##   Spring = col_double(),
##   Winter = col_double()
## )
```

```
grades
```

```
## # A tibble: 12 x 5
##    ‘ID Test‘  Year  Fall Spring Winter
##    <chr>     <dbl> <dbl>  <dbl>  <dbl>
##  1 1 Math     2008    15     16     19
##  2 1 Math     2009    12     13     27
##  3 1 Writin   2008    22     22     24
##  4 1 Writin   2009    10     14     20
##  5 2 Math     2008    12     13     25
##  6 2 Math     2009    16     14     21
##  7 2 Writin   2008    13     11     29
##  8 2 Writin   2009    23     20     26
##  9 3 Math     2008    11     12     22
## 10 3 Math     2009    13     11     27
## 11 3 Writin   2008    17     12     23
## 12 3 Writin   2009    14      9     31
```

Tidy this tibble! Some hints: 1) Start by making `ID` and `Test` two columns.
2) A single observation in the tidy version of this tibble is what hapened to one ID, in a given Year, in a specific Quarter. (So there should be one row with ID ==1, Year == 2008, Quarter == Fall. Another row will have ID ==1, Year == 2009, Quarter == Winter, etc.) 3) Your final tibble should have 5 columns and 18 rows

```
tidy_grades <- grades%>%
  separate("ID Test", into = c("ID", "Test"))%>%
  pivot_longer(cols=c('Fall':'Winter'),names_to = "Quarter")%>%
  pivot_wider(names_from=Test,values_from=value)
tidy_grades
```

```
## # A tibble: 18 x 5
##    ID     Year Quarter  Math Writin
##    <chr> <dbl> <chr>   <dbl>  <dbl>
##  1 1      2008 Fall       15     22
##  2 1      2008 Spring     16     22
##  3 1      2008 Winter     19     24
##  4 1      2009 Fall       12     10
##  5 1      2009 Spring     13     14
##  6 1      2009 Winter     27     20
##  7 2      2008 Fall       12     13
##  8 2      2008 Spring     13     11
##  9 2      2008 Winter     25     29
## 10 2      2009 Fall       16     23
## 11 2      2009 Spring     14     20
## 12 2      2009 Winter     21     26
## 13 3      2008 Fall       11     17
## 14 3      2008 Spring     12     12
## 15 3      2008 Winter     22     23
## 16 3      2009 Fall       13     14
## 17 3      2009 Spring     11      9
## 18 3      2009 Winter     27     31
```

6. Run this code block to create the `dates` tibble.

```
dates<-read_csv("dates.csv")
```

```
## Warning: Missing column names filled in: ’X4’ [4], ’X5’ [5], ’X6’ [6], ’X7’ [7]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   observation = col_double(),
##   TT = col_character(),
##   number = col_double(),
##   X4 = col_logical(),
##   X5 = col_logical(),
##   X6 = col_logical(),
##   X7 = col_logical()
## )
```

```
dates
```

```
## # A tibble: 15 x 7
##    observation TT    number X4    X5    X6    X7   
##          <dbl> <chr>  <dbl> <lgl> <lgl> <lgl> <lgl>
##  1           1 Month      3 NA    NA    NA    NA   
##  2           1 Day        2 NA    NA    NA    NA   
##  3           1 Year    2001 NA    NA    NA    NA   
##  4           2 Month      4 NA    NA    NA    NA   
##  5           2 Day        7 NA    NA    NA    NA   
##  6           2 Year    2003 NA    NA    NA    NA   
##  7           3 Month      6 NA    NA    NA    NA   
##  8           3 Day       15 NA    NA    NA    NA   
##  9           3 Year    2004 NA    NA    NA    NA   
## 10           4 Month      9 NA    NA    NA    NA   
## 11           4 Day       30 NA    NA    NA    NA   
## 12           4 Year    2007 NA    NA    NA    NA   
## 13           5 Month      8 NA    NA    NA    NA   
## 14           5 Day        1 NA    NA    NA    NA   
## 15           5 Year    2015 NA    NA    NA    NA   
```

Tidy this tibble! Your final answer should just have an `observation` column and a `Date` column, where the latter has a datatype.

```
tidy_dates<-dates%>%
  pivot_wider(names_from="TT",values_from=number)%>%
  select(-starts_with('X'))%>%
  unite(col="Date",c("Month","Day","Year"),sep="-")%>%
  mutate(Date=parse_date(Date,"%m-%d-%Y"))
tidy_dates
```

```
## # A tibble: 5 x 2
##   observation Date
##         <dbl> <date>
## 1           1 2001-03-02
## 2           2 2003-04-07
## 3           3 2004-06-15
## 4           4 2007-09-30
## 5           5 2015-08-01
```