Homework3

Task 1

1) If your working directory is myfolder/homework/, what relative path would you specify to get the file located at myfolder/MyData.csv?

2) What are the major benefits of using R projects?

3) What is git and what is github?

4) What are the two main differences between a tibble and a data.frame?

5) Rewrite the following nested function call using baseR's chaining operator:

6) What is meant by long format data and wide format data? Which do we generally prefer for statistical analysis?

Task 2 - Glass Data

```
library (tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```r
library(readr)
library(dplyr)
library(Lahman)
#1

#read in the raw data file named glass data, and put in column names.
glass_data <- read_delim("https://www4.stat.ncsu.edu/~online/datasets/glass.data",
                         delim = ",",
                         col_names = c("Id", "RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba",
```

```
Rows: 214 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (11): Id, RI, Na, Mg, Al, Si, K, Ca, Ba, Fe, X11

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.
```

```r
glass_data
```

```
# A tibble: 214 x 11
      Id    RI    Na    Mg    Al    Si     K    Ca    Ba    Fe   X11
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1     1  1.52  13.6  4.49  1.1   71.8  0.06  8.75     0  0         1
 2     2  1.52  13.9  3.6   1.36  72.7  0.48  7.83     0  0         1
 3     3  1.52  13.5  3.55  1.54  73.0  0.39  7.78     0  0         1
 4     4  1.52  13.2  3.69  1.29  72.6  0.57  8.22     0  0         1
 5     5  1.52  13.3  3.62  1.24  73.1  0.55  8.07     0  0         1
 6     6  1.52  12.8  3.61  1.62  73.0  0.64  8.07     0  0.26      1
 7     7  1.52  13.3  3.6   1.14  73.1  0.58  8.17     0  0         1
 8     8  1.52  13.2  3.61  1.05  73.2  0.57  8.24     0  0         1
 9     9  1.52  14.0  3.58  1.37  72.1  0.56  8.3      0  0         1
10    10  1.52  13    3.6   1.36  73.0  0.57  8.4      0  0.11      1
# i 204 more rows
```

```r
#2

#overwrite last column with character strings
glass_data_tbl <- as_tibble(glass_data)
glass_data_tbl |>
```

```
  mutate(desc = ifelse(glass_data_tbl$X11 == 1, "building_windows_float_processed",
                ifelse(glass_data_tbl$X11 == 2, "bulding_windows_non_float_processed",
                ifelse(glass_data_tbl$X11 == 3, "vehicle_windows_float_processed",
                ifelse(glass_data_tbl$X11 == 4, "vehicle_windows_non_float_processed",
                ifelse(glass_data_tbl$X11 == 5, "containers",
                ifelse(glass_data_tbl$X11 == 6, "tableware", "headlamps")))))) |>
#3
  #continue the chain and filter based on requirements.
  filter(Fe < 0.2, desc %in% c("tableware", "headlamps"))
```

```
# A tibble: 38 x 12
      Id    RI    Na    Mg    Al    Si     K    Ca    Ba    Fe   X11 desc
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
 1   177  1.52  14    2.39  1.56  72.4  0     9.57  0         0     6 tableware
 2   178  1.52  13.8  2.41  1.19  72.8  0     9.77  0         0     6 tableware
 3   179  1.52  14.5  2.24  1.62  72.4  0     9.26  0         0     6 tableware
 4   180  1.52  14.1  2.19  1.66  72.7  0     9.32  0         0     6 tableware
 5   181  1.51  14.4  1.74  1.54  74.6  0     7.59  0         0     6 tableware
 6   182  1.52  15.0  0.78  1.74  72.5  0     9.95  0         0     6 tableware
 7   183  1.52  14.2  0     2.09  72.7  0    10.9   0         0     6 tableware
 8   184  1.52  14.6  0     0.56  73.5  0    11.2   0         0     6 tableware
 9   185  1.51  17.4  0     0.34  75.4  0     6.65  0         0     6 tableware
10   186  1.51  13.7  3.2   1.81  72.8  1.76  5.43  1.19      0     7 headlamps
# i 28 more rows
```

```
print(glass_data_tbl)
```

```
# A tibble: 214 x 11
      Id    RI    Na    Mg    Al    Si     K    Ca    Ba    Fe   X11
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1     1  1.52  13.6  4.49  1.1   71.8  0.06  8.75     0  0         1
 2     2  1.52  13.9  3.6   1.36  72.7  0.48  7.83     0  0         1
 3     3  1.52  13.5  3.55  1.54  73.0  0.39  7.78     0  0         1
 4     4  1.52  13.2  3.69  1.29  72.6  0.57  8.22     0  0         1
 5     5  1.52  13.3  3.62  1.24  73.1  0.55  8.07     0  0         1
 6     6  1.52  12.8  3.61  1.62  73.0  0.64  8.07     0  0.26      1
 7     7  1.52  13.3  3.6   1.14  73.1  0.58  8.17     0  0         1
 8     8  1.52  13.2  3.61  1.05  73.2  0.57  8.24     0  0         1
 9     9  1.52  14.0  3.58  1.37  72.1  0.56  8.3      0  0         1
10    10  1.52  13    3.6   1.36  73.0  0.57  8.4      0  0.11      1
# i 204 more rows
```

Task 2 - Yeast Data

```
library(readr)
#1 - Read in raw delimited data and create column names based on HW requirements.
columnnames = c("seq_name",
                "mcg",
                "gvh",
                "alm",
                "mit",
                "erl",
                "pox",
                "vac",
                "nuc",
                "class")
yeast_data <- read_delim("https://www4.stat.ncsu.edu/~online/datasets/yeast.data",
                         delim = "  ",
                         col_names = columnnames)
```

```
Warning: One or more parsing issues, call `problems()` on your data frame for details,
e.g.:
  dat <- vroom(...)
  problems(dat)
```

```
Rows: 1484 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: "  "
chr (2): seq_name, class
dbl (8): mcg, gvh, alm, mit, erl, pox, vac, nuc

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.
```

```
print(yeast_data)
```

```
# A tibble: 1,484 x 10
   seq_name     mcg   gvh   alm   mit   erl   pox   vac   nuc class
   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
 1 ADT1_YEAST  0.58  0.61  0.47  0.13   0.5   0    0.48  0.22 MIT
 2 ADT2_YEAST  0.43  0.67  0.48  0.27   0.5   0    0.53  0.22 MIT
 3 ADT3_YEAST  0.64  0.62  0.49  0.15   0.5   0    0.53  0.22 MIT
```

```
 4 AAR2_YEAST  0.58  0.44  0.57  0.13   0.5   0    0.54  0.22 NUC
 5 AATM_YEAST  0.42  0.44  0.48  0.54   0.5   0    0.48  0.22 MIT
 6 AATC_YEAST  0.51  0.4   0.56  0.17   0.5   0.5  0.49  0.22 CYT
 7 ABC1_YEAST  0.5   0.54  0.48  0.65   0.5   0    0.53  0.22 MIT
 8 BAF1_YEAST  0.48  0.45  0.59  0.2    0.5   0    0.58  0.34 NUC
 9 ABF2_YEAST  0.55  0.5   0.66  0.36   0.5   0    0.49  0.22 MIT
10 ABP1_YEAST  0.4   0.39  0.6   0.15   0.5   0    0.58  0.3  CYT
# i 1,474 more rows
```

```
yeast_data_tbl <- as_tibble(yeast_data)
print(yeast_data_tbl)
```

```
# A tibble: 1,484 x 10
   seq_name     mcg   gvh   alm   mit   erl   pox   vac   nuc class
   <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
 1 ADT1_YEAST  0.58  0.61  0.47  0.13   0.5   0    0.48  0.22 MIT
 2 ADT2_YEAST  0.43  0.67  0.48  0.27   0.5   0    0.53  0.22 MIT
 3 ADT3_YEAST  0.64  0.62  0.49  0.15   0.5   0    0.53  0.22 MIT
 4 AAR2_YEAST  0.58  0.44  0.57  0.13   0.5   0    0.54  0.22 NUC
 5 AATM_YEAST  0.42  0.44  0.48  0.54   0.5   0    0.48  0.22 MIT
 6 AATC_YEAST  0.51  0.4   0.56  0.17   0.5   0.5  0.49  0.22 CYT
 7 ABC1_YEAST  0.5   0.54  0.48  0.65   0.5   0    0.53  0.22 MIT
 8 BAF1_YEAST  0.48  0.45  0.59  0.2    0.5   0    0.58  0.34 NUC
 9 ABF2_YEAST  0.55  0.5   0.66  0.36   0.5   0    0.49  0.22 MIT
10 ABP1_YEAST  0.4   0.39  0.6   0.15   0.5   0    0.58  0.3  CYT
# i 1,474 more rows
```

```
#2 - Remove seq_name and nuc columns from tibble.
yeast_data_tbl |>
  select(-seq_name, -nuc) |>
#add mean and median columns corresponding to each numeric variable.
  mutate(across(where(is.numeric), list(mean = mean, median = median), na.rm = TRUE, .names =
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `across(...)`.
Caused by warning:
! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
Supply arguments directly to `.fns` through an anonymous function instead.

  # Previously
  across(a:b, mean, na.rm = TRUE)
```

```
  # Now
  across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
# A tibble: 1,484 x 22
     mcg   gvh   alm   mit   erl   pox   vac class mcg_mean mcg_median gvh_mean
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>    <dbl>      <dbl>    <dbl>
 1  0.58  0.61  0.47  0.13   0.5   0    0.48 MIT      0.501       0.49    0.500
 2  0.43  0.67  0.48  0.27   0.5   0    0.53 MIT      0.501       0.49    0.500
 3  0.64  0.62  0.49  0.15   0.5   0    0.53 MIT      0.501       0.49    0.500
 4  0.58  0.44  0.57  0.13   0.5   0    0.54 NUC      0.501       0.49    0.500
 5  0.42  0.44  0.48  0.54   0.5   0    0.48 MIT      0.501       0.49    0.500
 6  0.51  0.4   0.56  0.17   0.5   0.5  0.49 CYT      0.501       0.49    0.500
 7  0.5   0.54  0.48  0.65   0.5   0    0.53 MIT      0.501       0.49    0.500
 8  0.48  0.45  0.59  0.2    0.5   0    0.58 NUC      0.501       0.49    0.500
 9  0.55  0.5   0.66  0.36   0.5   0    0.49 MIT      0.501       0.49    0.500
10  0.4   0.39  0.6   0.15   0.5   0    0.58 CYT      0.501       0.49    0.500
# i 1,474 more rows
# i 11 more variables: gvh_median <dbl>, alm_mean <dbl>, alm_median <dbl>,
#   mit_mean <dbl>, mit_median <dbl>, erl_mean <dbl>, erl_median <dbl>,
#   pox_mean <dbl>, pox_median <dbl>, vac_mean <dbl>, vac_median <dbl>
```

#Task 2: Combining Excel and Delimited Data

```
#1
library(readxl)
#import raw excel data, just the first sheet.
white_wine <- read_excel("white-wine.xlsx",
                         sheet = excel_sheets("white-wine.xlsx")[1])
white_wine_tbl <- as_tibble(white_wine)
view(white_wine_tbl)

print(white_wine_tbl)
```

```
# A tibble: 4,898 x 12
   `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
             <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
 1               7               0.27          0.36             20.7     0.045
 2              63               0.3           0.34              1.6     0.049
 3              81               0.28          0.4               6.9     0.05
 4              72               0.23          0.32              8.5     0.058
```

```
 5              72               0.23          0.32              8.5      0.058
 6              81               0.28          0.4               6.9      0.05
 7              62               0.32          0.16              7        0.045
 8               7               0.27          0.36             20.7      0.045
 9              63               0.3           0.34              1.6      0.049
10              81               0.22          0.43              1.5      0.044
# i 4,888 more rows
# i 7 more variables: `free sulfur dioxide` <dbl>,
#   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
#   alcohol <dbl>, quality <dbl>
```

```
#2
#import second sheet of raw excel data with all variable names and info.
white_wine_2nd <- read_excel("white-wine.xlsx",
                             sheet = excel_sheets("white-wine.xlsx")[2])
#make column names of RData object as variables from 2nd sheet.
colnames(white_wine) <- white_wine_2nd$Variables

#3
#add additional column as per HW requirements.
white_wine_tbl$Type = c("White")

white_wine$Type = c("White")



#4 - do same as above to red wine raw delimited file.
red_wine <- read_delim("https://www4.stat.ncsu.edu/~online/datasets/red-wine.csv",
                       delim = ";")
```

```
Rows: 1599 Columns: 12
-- Column specification --------------------------------------------------------
Delimiter: ";"
dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar, chlo...

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.
```

```
colnames(red_wine) <- white_wine_2nd$Variables

red_wine$Type = c("Red")
```

```r
#5 - combine the two datasets together
wines_overall <- bind_rows(white_wine, red_wine)
print(wines_overall)
```

```
# A tibble: 6,497 x 13
   fixed_acidity volatile_acidity citric_acid residual_sugar chlorides
           <dbl>            <dbl>       <dbl>          <dbl>     <dbl>
 1             7             0.27        0.36           20.7     0.045
 2            63             0.3         0.34            1.6     0.049
 3            81             0.28        0.4             6.9     0.05
 4            72             0.23        0.32            8.5     0.058
 5            72             0.23        0.32            8.5     0.058
 6            81             0.28        0.4             6.9     0.05
 7            62             0.32        0.16            7       0.045
 8             7             0.27        0.36           20.7     0.045
 9            63             0.3         0.34            1.6     0.049
10            81             0.22        0.43            1.5     0.044
# i 6,487 more rows
# i 8 more variables: free_sulfur_dioxide <dbl>, total_sulfur_dioxide <dbl>,
#   density <dbl>, pH <dbl>, sulphates <dbl>, alcohol <dbl>, quality <dbl>,
#   Type <chr>
```

```r
#6 through 9
library(dplyr)
filtered_wines_overall <- wines_overall |>
  #arrange quality variable from descending order, largest to smallest.
  arrange(desc(wines_overall$quality)) |>
  #filter data with quality < 6.5, and alcohol < 132.
  filter(wines_overall$quality > 6.5, wines_overall$alcohol < 132) |>
  #Select only certain variables.
  select(contains("acid"),contains("alcohol"), "Type", "quality") |>
  group_by(quality) |>
  #add mean and standard deviation of alcohol variable for each quality setting.  To do this
  mutate(wine_alcohol_mean = mean(alcohol),
         wine_alcohol_sd = sd(alcohol))

  print(filtered_wines_overall)
```

```
# A tibble: 1,206 x 8
# Groups:   quality [6]
   fixed_acidity volatile_acidity citric_acid alcohol Type  quality
```

```
           <dbl>            <dbl>       <dbl>   <dbl> <chr>   <dbl>
 1            67             0.26        0.39      96 White       8
 2            61             0.31        0.58     123 White       8
 3            64             0.32        0.35     125 White       8
 4             6             0.25        0.28     129 White       8
 5            59             0.27        0.29     105 White       8
 6            65             0.36        0.28     124 White       8
 7            73             0.3         0.34     128 White       8
 8            53             0.24        0.33      11 White       8
 9            82             0.37        0.36     117 White       8
10            72             0.26        0.44     111 White       8
# i 1,196 more rows
# i 2 more variables: wine_alcohol_mean <dbl>, wine_alcohol_sd <dbl>
```

#Task 3 - Database Practice

```
library(DBI)
library(RSQLite)

#1 - Connect to database
lahman_db <- dbConnect(RSQLite::SQLite(), "lahman.db")
dbListTables(lahman_db)
```

```
 [1] "AllstarFull"         "Appearances"         "AwardsManagers"
 [4] "AwardsPlayers"       "AwardsShareManagers" "AwardsSharePlayers"
 [7] "Batting"             "BattingPost"         "CollegePlaying"
[10] "Fielding"            "FieldingOF"          "FieldingOFsplit"
[13] "FieldingPost"        "HallOfFame"          "HomeGames"
[16] "LahmanData"          "Managers"            "ManagersHalf"
[19] "Parks"               "People"              "Pitching"
[22] "PitchingPost"        "Salaries"            "Schools"
[25] "SeriesPost"          "Teams"               "TeamsFranchises"
[28] "TeamsHalf"           "battingLabels"       "fieldingLabels"
[31] "pitchingLabels"
```

```
#2 -  return all data from Teams table in the year 2015.
tbl(lahman_db, "Teams") |>
  filter(yearID == 2015)
```

```
# Source:   SQL [?? x 48]
# Database: sqlite 3.45.2 [/Users/pranavnair/Documents/Grad School/ST 558/Homework3/lahman.db
```

```
   yearID lgID  teamID franchID divID  Rank     G Ghome     W     L DivWin WCWin
    <int> <chr> <chr>  <chr>    <chr> <int> <int> <int> <int> <int> <chr>  <chr>
 1   2015 NL    ARI    ARI      W         3   162    81    79    83 N      N
 2   2015 NL    ATL    ATL      E         4   162    81    67    95 N      N
 3   2015 AL    BAL    BAL      E         3   162    78    81    81 N      N
 4   2015 AL    BOS    BOS      E         5   162    81    78    84 N      N
 5   2015 AL    CHA    CHW      C         4   162    81    76    86 N      N
 6   2015 NL    CHN    CHC      C         3   162    81    97    65 N      Y
 7   2015 NL    CIN    CIN      C         5   162    81    64    98 N      N
 8   2015 AL    CLE    CLE      C         3   161    80    81    80 N      N
 9   2015 NL    COL    COL      W         5   162    81    68    94 N      N
10   2015 AL    DET    DET      C         5   161    81    74    87 N      N
# i more rows
# i 36 more variables: LgWin <chr>, WSWin <chr>, R <int>, AB <int>, H <int>,
#   X2B <int>, X3B <int>, HR <int>, BB <int>, SO <int>, SB <int>, CS <int>,
#   HBP <int>, SF <int>, RA <int>, ER <int>, ERA <dbl>, CG <int>, SHO <int>,
#   SV <int>, IPouts <int>, HA <int>, HRA <int>, BBA <int>, SOA <int>, E <int>,
#   DP <int>, FP <dbl>, name <chr>, park <chr>, attendance <int>, BPF <int>,
#   PPF <int>, teamIDBR <chr>, teamIDlahman45 <chr>, teamIDretro <chr>
```

```r
#3 -  Use SQL to do the same as shown in question 2.
lahman_db2 <- dbConnect(RSQLite::SQLite(), "lahman.db")

tbl(lahman_db2, sql(
  "SELECT 'yearID', 'playerID', 'teamID', 'lgID'
  FROM 'Teams'
  WHERE ('yearID' = 2015.0)")
  )
```

```
# Source:   SQL [0 x 4]
# Database: sqlite 3.45.2 [/Users/pranavnair/Documents/Grad School/ST 558/Homework3/lahman.db
# i 4 variables: 'yearID' <lgl>, 'playerID' <lgl>, 'teamID' <lgl>, 'lgID' <lgl>
```

```r
#4 - Return all players from Hall of Fame table.
lahman_db_modified <- tbl(lahman_db, "HallOfFame") |>
  select(playerID, yearID, category) |>
print(lahman_db_modified)
```

```
# Source:   SQL [?? x 3]
# Database: sqlite 3.45.2 [/Users/pranavnair/Documents/Grad School/ST 558/Homework3/lahman.db
   playerID  yearID category
```

```
   <chr>      <int> <chr>
 1 cobbty01    1936 Player
 2 ruthba01    1936 Player
 3 wagneho01   1936 Player
 4 mathech01   1936 Player
 5 johnswa01   1936 Player
 6 lajoina01   1936 Player
 7 speaktr01   1936 Player
 8 youngcy01   1936 Player
 9 hornsro01   1936 Player
10 cochrmi01   1936 Player
# i more rows
```

```
#5 - Combine People table with previously modified HallOfFame table.
lahman_db_join <- tbl(lahman_db, "People") |>
left_join(lahman_db_modified, tbl(lahman_db,"People"),
          by = join_by(playerID == playerID)) |>
  collect() |>
  select(playerID, yearID, nameFirst, nameLast)
print(lahman_db_join)
```

```
# A tibble: 23,655 x 4
   playerID  yearID nameFirst nameLast
   <chr>      <int> <chr>     <chr>
 1 aardsda01     NA David     Aardsma
 2 aaronha01   1982 Hank      Aaron
 3 aaronto01     NA Tommie    Aaron
 4 aasedo01      NA Don       Aase
 5 abadan01      NA Andy      Abad
 6 abadfe01      NA Fernando  Abad
 7 abadijo01     NA John      Abadie
 8 abbated01     NA Ed        Abbaticchio
 9 abbeybe01     NA Bert      Abbey
10 abbeych01     NA Charlie   Abbey
# i 23,645 more rows
```

```
#6 - Using chaining to select certain variables in table, group by one variable, and summari
lahman_db_managers <- tbl(lahman_db, "Managers") |>
      select(playerID, G, W, L) |>
      group_by(playerID) |>
      summarize(G_managed = sum(G, na.rm = TRUE),
             Total_W = sum(W, na.rm = TRUE),
```

```
            Total_L = sum(L, na.rm = TRUE)) |>
      collect() |>
      mutate(win_loss_percentage = Total_W/G_managed) |>
      arrange(desc(win_loss_percentage))
print(lahman_db_managers)
```

```
# A tibble: 749 x 5
   playerID  G_managed Total_W Total_L win_loss_percentage
   <chr>         <int>   <int>   <int>               <dbl>
 1 bensove01         1       1       0                   1
 2 burwebi01         1       1       0                   1
 3 cohenan01         1       1       0                   1
 4 ebeldi99          3       3       0                   1
 5 falkbi01          1       1       0                   1
 6 hardeme01         3       3       0                   1
 7 simmote01         1       1       0                   1
 8 steinte01         2       2       0                   1
 9 sukefcl01         2       2       0                   1
10 tamarjo01         1       1       0                   1
# i 739 more rows
```

```
#7 - returning all variables from last two questions.
lahman_total_join <- full_join(lahman_db_managers, lahman_db_join,
                               by = join_by(playerID == playerID)) |>
                     select(everything())

print(lahman_total_join)
```

```
# A tibble: 23,655 x 8
   playerID  G_managed Total_W Total_L win_loss_percentage yearID nameFirst
   <chr>         <int>   <int>   <int>               <dbl>  <int> <chr>
 1 bensove01         1       1       0                   1     NA Vern
 2 burwebi01         1       1       0                   1     NA Bill
 3 cohenan01         1       1       0                   1     NA Andy
 4 ebeldi99          3       3       0                   1     NA Dino
 5 falkbi01          1       1       0                   1     NA Bibb
 6 hardeme01         3       3       0                   1   1949 Mel
 7 hardeme01         3       3       0                   1   1950 Mel
 8 hardeme01         3       3       0                   1   1951 Mel
 9 hardeme01         3       3       0                   1   1952 Mel
10 hardeme01         3       3       0                   1   1953 Mel
```

```
# i 23,645 more rows
# i 1 more variable: nameLast <chr>
```