hw4

Exercise 4.13

Name every player in baseball history who has accumulated at least 300 home runs (HR) and at least 300 stolen bases (SB).

```
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
```

```
library(Lahman)
str(Batting)
```

```
## 'data.frame':    105861 obs. of  22 variables:
##  $ playerID: chr  "abercda01" "addybo01" "allisar01" "allisdo01" ...
##  $ yearID  : int  1871 1871 1871 1871 1871 1871 1871 1871 1871 1871 ...
##  $ stint   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ teamID  : Factor w/ 149 levels "ALT","ANA","ARI",..: 136 111 39 142 111 56 111 24 56 24 ...
##  $ lgID    : Factor w/ 7 levels "AA","AL","FL",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ G       : int  1 25 29 27 25 12 1 31 1 18 ...
##  $ AB      : int  4 118 137 133 120 49 4 157 5 86 ...
##  $ R       : int  0 30 28 28 29 9 0 66 1 13 ...
##  $ H       : int  0 32 40 44 39 11 1 63 1 13 ...
##  $ X2B     : int  0 6 4 10 11 2 0 10 1 2 ...
##  $ X3B     : int  0 0 5 2 3 1 0 9 0 1 ...
##  $ HR      : int  0 0 0 2 0 0 0 0 0 0 ...
##  $ RBI     : int  0 13 19 27 16 5 2 34 1 11 ...
##  $ SB      : int  0 8 3 1 6 0 0 11 0 1 ...
##  $ CS      : int  0 1 1 1 2 1 0 6 0 0 ...
##  $ BB      : int  0 4 2 0 2 0 1 13 0 0 ...
##  $ SO      : int  0 0 5 2 1 1 0 1 0 0 ...
##  $ IBB     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ HBP     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SH      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SF      : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ GIDP    : int  0 0 1 0 0 0 0 1 0 0 ...
```

```
str(Master)
```

```
## 'data.frame':    19617 obs. of  26 variables:
##  $ playerID    : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
##  $ birthYear   : int  1981 1934 1939 1954 1972 1985 1850 1877 1869 1866 ...
##  $ birthMonth  : int  12 2 8 9 8 12 11 4 11 10 ...
##  $ birthDay    : int  27 5 5 8 25 17 4 15 11 14 ...
##  $ birthCountry: chr  "USA" "USA" "USA" "USA" ...
##  $ birthState  : chr  "CO" "AL" "AL" "CA" ...
##  $ birthCity   : chr  "Denver" "Mobile" "Mobile" "Orange" ...
##  $ deathYear   : int  NA NA 1984 NA NA NA 1905 1957 1962 1926 ...
##  $ deathMonth  : int  NA NA 8 NA NA NA 5 1 6 4 ...
##  $ deathDay    : int  NA NA 16 NA NA NA 17 6 11 27 ...
##  $ deathCountry: chr  NA NA "USA" NA ...
##  $ deathState  : chr  NA NA "GA" NA ...
##  $ deathCity   : chr  NA NA "Atlanta" NA ...
##  $ nameFirst   : chr  "David" "Hank" "Tommie" "Don" ...
##  $ nameLast    : chr  "Aardsma" "Aaron" "Aaron" "Aase" ...
##  $ nameGiven   : chr  "David Allan" "Henry Louis" "Tommie Lee" "Donald William" ...
##  $ weight      : int  215 180 190 190 184 220 192 170 175 169 ...
##  $ height      : int  75 72 75 75 73 73 72 71 71 68 ...
##  $ bats        : Factor w/ 3 levels "B","L","R": 3 3 3 3 2 2 3 3 3 2 ...
##  $ throws      : Factor w/ 3 levels "L","R","S": 2 2 2 2 1 1 2 2 2 1 ...
##  $ debut       : chr  "2004-04-06" "1954-04-13" "1962-04-10" "1977-07-26" ...
##  $ finalGame   : chr  "2015-08-23" "1976-10-03" "1971-09-26" "1990-10-03" ...
##  $ retroID     : chr  "aardd001" "aaroh101" "aarot101" "aased001" ...
##  $ bbrefID     : chr  "aardsda01" "aaronha01" "aaronto01" "aasedo01" ...
##  $ deathDate   : Date, format: NA NA ...
##  $ birthDate   : Date, format: "1981-12-27" "1934-02-05" ...
```

```
str(Teams)
```

```
## 'data.frame':    2895 obs. of  48 variables:
##  $ yearID        : int  1871 1871 1871 1871 1871 1871 1871 1871 1871 1872 ...
##  $ lgID          : Factor w/ 7 levels "AA","AL","FL",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ teamID        : Factor w/ 149 levels "ALT","ANA","ARI",..: 24 31 39 56 90 97 111 136 142 8 ...
##  $ franchID      : Factor w/ 120 levels "ALT","ANA","ARI",..: 13 36 25 56 70 85 91 109 77 9 ...
##  $ divID         : chr  NA NA NA NA ...
##  $ Rank          : int  3 2 8 7 5 1 9 6 4 2 ...
##  $ G             : int  31 28 29 19 33 28 25 29 32 58 ...
##  $ Ghome         : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ W             : int  20 19 10 7 16 21 4 13 15 35 ...
##  $ L             : int  10 9 19 12 17 7 21 15 15 19 ...
##  $ DivWin        : chr  NA NA NA NA ...
##  $ WCWin         : chr  NA NA NA NA ...
##  $ LgWin         : chr  "N" "N" "N" "N" ...
##  $ WSWin         : chr  NA NA NA NA ...
##  $ R             : int  401 302 249 137 302 376 231 351 310 617 ...
##  $ AB            : int  1372 1196 1186 746 1404 1281 1036 1248 1353 2571 ...
##  $ H             : int  426 323 328 178 403 410 274 384 375 753 ...
##  $ X2B           : int  70 52 35 19 43 66 44 51 54 106 ...
##  $ X3B           : int  37 21 40 8 21 27 25 34 26 31 ...
##  $ HR            : int  3 10 7 2 1 9 3 6 6 14 ...
##  $ BB            : num  60 60 26 33 33 46 38 49 48 29 ...
##  $ SO            : int  19 22 25 9 15 23 30 19 13 28 ...
##  $ SB            : num  73 69 18 16 46 56 53 62 48 53 ...
```

```
##  $ CS            : num  16 21 8 4 15 12 10 24 13 18 ...
##  $ HBP           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SF            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ RA            : int  303 241 341 243 313 266 287 362 303 434 ...
##  $ ER            : int  109 77 116 97 121 137 108 153 137 166 ...
##  $ ERA           : num  3.55 2.76 4.11 5.17 3.72 4.95 4.3 5.51 4.37 2.9 ...
##  $ CG            : int  22 25 23 19 32 27 23 28 32 48 ...
##  $ SHO           : int  1 0 0 1 1 0 1 0 0 1 ...
##  $ SV            : int  3 1 0 0 0 0 0 0 0 1 ...
##  $ IPouts        : int  828 753 762 507 879 747 678 750 846 1548 ...
##  $ HA            : int  367 308 346 261 373 329 315 431 371 573 ...
##  $ HRA           : int  2 6 13 5 7 3 3 4 4 3 ...
##  $ BBA           : int  42 28 53 21 42 53 34 75 45 63 ...
##  $ SOA           : int  23 22 34 17 22 16 16 12 13 77 ...
##  $ E             : int  243 229 234 163 235 194 220 198 218 432 ...
##  $ DP            : int  24 16 15 8 14 13 14 22 20 22 ...
##  $ FP            : num  0.834 0.829 0.818 0.803 0.84 0.845 0.821 0.845 0.85 0.83 ...
##  $ name          : chr  "Boston Red Stockings" "Chicago White Stockings" "Cleveland Forest Citys" "Fo
##  $ park          : chr  "South End Grounds I" "Union Base-Ball Grounds" "National Association Grounds
##  $ attendance    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BPF           : int  103 104 96 101 90 102 97 101 94 106 ...
##  $ PPF           : int  98 102 100 107 88 98 99 100 98 102 ...
##  $ teamIDBR      : chr  "BOS" "CHI" "CLE" "KEK" ...
##  $ teamIDlahman45: chr  "BS1" "CH1" "CL1" "FW1" ...
##  $ teamIDretro   : chr  "BS1" "CH1" "CL1" "FW1" ...
```

```
x<-Batting %>% group_by(playerID) %>%summarize(NHR=sum(HR), NSB=sum(SB)) %>% filter(NHR >= 300 & NSB>=30
x
```

```
## # A tibble: 8 x 1
##   nameGiven
##   <chr>
## 1 Alexander Enmanuel
## 2 Andre Nolan
## 3 Barry Lamar
## 4 Bobby Lee
## 5 Carlos Ivan
## 6 Reginald Laverne
## 7 Steven Allen
## 8 Willie Howard
```

Exercise 4.14

Name every pitcher in baseball history who has accumulated at least 300 wins (W) and at least 3,000 strikeouts (SO).

```
x<-Batting %>% group_by(playerID, teamID) %>% summarize(NSO=sum(SO)) %>% select(playerID, teamID, NSO)
y<-Teams %>% group_by(teamID) %>% summarize(NW=sum(W)) %>% filter(NW >= 300) %>% select(teamID, NW)
z<-x %>% left_join(y, by = c("teamID"="teamID")) %>% group_by(playerID) %>% summarize(NNSO=sum(NSO), NNW
z
```

```
## # A tibble: 2 x 1
```

```
##   nameGiven
##   <chr>
## 1 James Howard
## 2 Reginald Martinez
```

Exercise 4.15

Identify the name and year of every player who has hit at least 50 home runs in a single season. Which player had the lowest batting average in that season?

```
players_50_hr <- Batting %>% group_by(playerID, yearID) %>% summarize(NHR=sum(HR)) %>% filter(NHR>=50)
```

```
## Adding missing grouping variables: `playerID`
```

```
players_50_hr
```

```
## # A tibble: 45 x 3
## # Groups:   playerID [29]
##    playerID  nameGiven          yearID
##    <chr>     <chr>               <int>
##  1 judgeaa01 Aaron James          2017
##  2 belleal01 Albert Jojuan        1995
##  3 rodrial01 Alexander Enmanuel   2001
##  4 rodrial01 Alexander Enmanuel   2002
##  5 rodrial01 Alexander Enmanuel   2007
##  6 jonesan01 Andruw Rudolf        2005
##  7 bondsba01 Barry Lamar          2001
##  8 anderbr01 Brady Kevin          1996
##  9 fieldce01 Cecil Grant          1990
## 10 davisch02 Christopher Lyn      2013
## # ... with 35 more rows
```

Part 2:

Exercise 5.6

An analyst wants to calculate the pairwise differences between the Treatment and Control values for a small data set from a crossover trial (all subjects received both treatments) that consists of the following observations.

```
tab <- xtable(ds1)
print(tab, floating=FALSE)
```

	id	group	vals
1	1	T	4.00
2	2	T	6.00
3	3	T	8.00
4	1	C	5.00
5	2	C	6.00
6	3	C	10.00

They use the following code to create the new `diff` variable.

```
Treat <- filter(ds1, group=="T")
Control <- filter(ds1, group=="C")
all <- mutate(Treat, diff = Treat$vals - Control$vals)
all
```

Verify that this code works for this example and generates the correct values of -1, 0, and -2. Describe two problems that might arise if the data set is not sorted in a particular order or if one of the observations is missing for one of the subjects. Provide an alternative approach to generate this variable that is more robust (hint: use `tidyr::spread()`).

The code would work only when the Treat and Control objects are gathered ordered by id, which could or could not happen as there is no order specified. In any other case this code won't do what we intended. One problem, as mentioned earlier is that the results are not ordered by id and so the difference is arbitrary and not a pairwise difference. Another thing that could happen is the values are not in the correct data type or are NA (null), in which case we need to use coalesce() to specify a default value (probably 0) for any NA records.

res<- ds1%>%spread(group, vals)%>%mutate(diff=T-C)

This way we ensure that the difference is done among records with the same id.

Exercise 5.7

Generate the code to convert the following data frame to wide format.

---

	grp	sex	meanL	sdL	meanR	sdR
1	A	F	0.22	0.11	0.34	0.08
2	A	M	0.47	0.33	0.57	0.33
3	B	F	0.33	0.11	0.40	0.07
4	B	M	0.55	0.31	0.65	0.27

The result should look like the following display.

	grp	F.meanL	F.meanR	F.sdL	F.sdR	M.meanL	M.meanR	M.sdL	M.sdR
1	A	0.22	0.34	0.11	0.08	0.47	0.57	0.33	0.33
2	B	0.33	0.40	0.11	0.07	0.55	0.65	0.31	0.27

Hint: use `gather()` in conjunction with `spread()`.

First I'll create a new column called field that will have meanL, sdL, meanR and sdR as values. As a product, we'll have 16 rows and 3 columns

res->ds1%>%gather(field, value, meanL:sdR)

Then I'll spread those rows into columns

res->ds1%>%spread(sex, values)

In one step:

ds1%>%gather(field, value, meanL:sdR) %>%spread(sex, values)

Part 3:

A random sample of 60 individuals was selected as part of a study on drug usage. The average usage was found to be 350 mg. For the population in the March quarter of the previous year it was found that the population mean usage was 355 mg and standard deviation of the usage was 81mg. Significance level $\alpha = 0.05$.

The researcher wishes to contest the claim that the drug usage is different from 355 mg.

1. **Formulate the Null and Alternate Hypothesis:**

$$H_0 : \mu = 355$$
$$H_1 : \mu \neq 355$$

2. **Which test will you use in this scenario? Give your rationale why?**

   The sample size is greater than 30 (n=60), the sample is an SRS and we assume that the population is 20 times larger than the sample. This satisfies normality conditions. Because we have an estimation for the standard deviation of the population and not the actual standard deviation, we'll use t-test to calculate the confidence intervals. Because the alternative hypothesis is for different than 355, we will use a two-tailed test.

3. **Calculate the test statistics for two-tailed test. Using the test statistics state whether you will reject or accept the null hypothesis.**

$$t = \frac{\bar{x} - u}{s/\sqrt{n}}$$
$$t = \frac{350 - 355}{81/\sqrt{60}}$$
$$t = -0.478$$

$T_{critical}$ value at $\alpha = 0.05$ is $t_c = 2.001$

$$t_c \geq |t| = 0.478$$

$t_c \geq$ test statistics

   It is concluded that null hypothesis is not rejected.

4. **Find the p-value from the table**

   p-value at t=-0.478 is 0.6343

5. **Use the p-value and the significance level α to decide and explain whether to reject or accept the Null Hypothesis.**

$$\text{p} - \text{value} > \text{significance level}$$
$$0.643 > 0.05$$

It is concluded that the null hypothesis is not rejected

6. Create a graph to show the test statistics, critical statistics, alpha and p-value.