Homework 1 – Due Monday September 21, 2015

Cheryl Calhoun

09/03/2015

Add your code blocks to this document and alter the Author name to yours. Your submission will consist of your own R Markdown file plus the compiled HTML version of the document.

I encourage you to submit your completed version as soon as possible. We will grade this quickly so that any opportunities for a regrade can be done quickly.

Assignment

This homework will continue using `dplyr` and the operations to join related data frames together. We will use the `GSS-extract-2.RData` data set from class on September 14.

Recall the in-class exercise from Monday the 14th wherein we made smaller data frames based on two factor variables, first by making a long table. Let's make a few more.

Part I

1. Construct a long data frame wherein you find the mean hours of TV watched and radio listened to for each subgroup of marital status and education achieved, only with respondents after 1980 who are under 50 years of age. Join this to the original data frame.

```
## Filter out records which are missing values. Create subGSS.2I1 to hold data set for homework 2, Part
subGSS.2I1 <- filter(subGSS.2, !is.na(radiohrs) & !is.na(tvhours))

## Filter out respondents over 50 years, and before 1980
subGSS.2I1 <- filter(subGSS.2I1, age < 50 & year >= 1980)

## Group data set by marital status and degree level, calculate mean tvhours for each level, calculate m
by_age_year <- subGSS.2I1 %>%
  group_by (degree, marital) %>%
  mutate (MTVhours = mean(tvhours, na.rm=TRUE), Mradiohrs = mean(radiohrs, na.rm=TRUE))

## display by_age_year
by_age_year
```

```
## Source: local data frame [1,994 x 13]
## Groups: degree, marital [27]
##
##     year  grass       marital   age         degree tvhours radiohrs racmar
##    <int> <fctr>        <fctr> <int>         <fctr>   <int>    <int> <fctr>
## 1   1982     NA       married    41 lt high school       2        8     no
## 2   1982     NA       married    49    high school       4        1    yes
## 3   1982     NA never married    27    high school       2        2     no
## 4   1982     NA never married    24    high school       3        2     no
## 5   1982     NA never married    29       bachelor       1        1     no
## 6   1982     NA never married    21 junior college       1        1     no
## 7   1982     NA       married    30       bachelor       1        3     no
## 8   1982     NA       married    39    high school       3        1    yes
```

```
## 9   1982     NA       married    36    high school       1        1     no
## 10  1982     NA never married    20    high school       2        4     no
## # ... with 1,984 more rows, and 5 more variables: premarsx <fctr>,
## #   SSMar <fctr>, conrinc <dbl>, MTVhours <dbl>, Mradiohrs <dbl>
```

```
## Add MTVhours & Mradiohrs back into subGSS.2I1
subGSS.2I1 <- left_join(subGSS.2, by_age_year)
```

```
## Joining, by = c("year", "grass", "marital", "age", "degree", "tvhours", "radiohrs", "racmar", "prema
```

```
## Verify MTVhours and Mradiohrs are joined with subGSS.2.
head(subGSS.2I1, 2)
```

```
##   year grass       marital age         degree tvhours radiohrs racmar
## 1 1972  <NA> never married  23       bachelor      NA       NA     no
## 2 1972  <NA>       married  70 lt high school      NA       NA    yes
##           premarsx SSMar conrinc MTVhours Mradiohrs
## 1 not wrong at all  <NA>      NA       NA        NA
## 2     always wrong  <NA>      NA       NA        NA
```

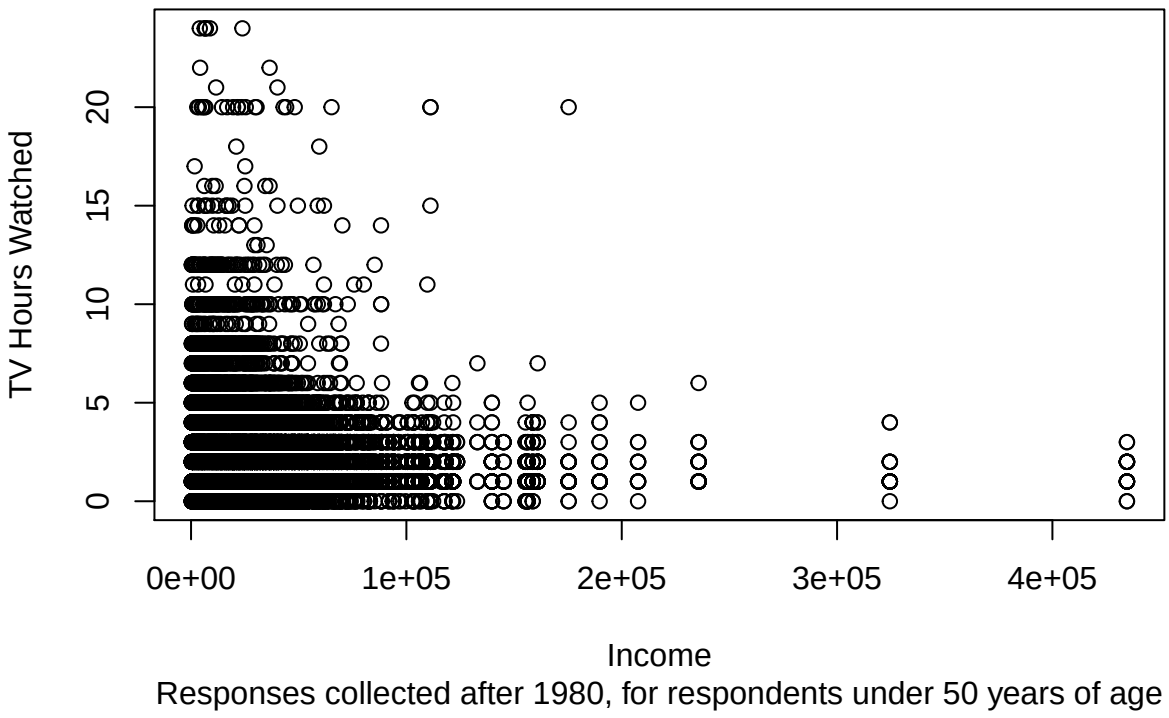
Results: subGSS.2I1 now contains 13 variables (added mean values for radio hours and TV hours), vs. the 11 variables it started with. The data set represents responses obtained after 1980 for respondents who are under 50 years of age.

2. Create a scatterplot using the `plot` function of TV hours watched (y) versus income (x) for all individuals. Add arguments `main`, `xlab` and `ylab` for a title and axis labels.

```
## Create a scatterplot of TV hours watched (y) versus income (x) for all individuals in subGSS.2I2.

## Remove rows with missing values for conrinc and assign to subGSS.2I2.
subGSS.2I2 <- filter(subGSS.2I1, !is.na(conrinc))

plot(subGSS.2I2$conrinc, subGSS.2I2$tvhours, type="p", main="TV hours watched based on income levels", 
```



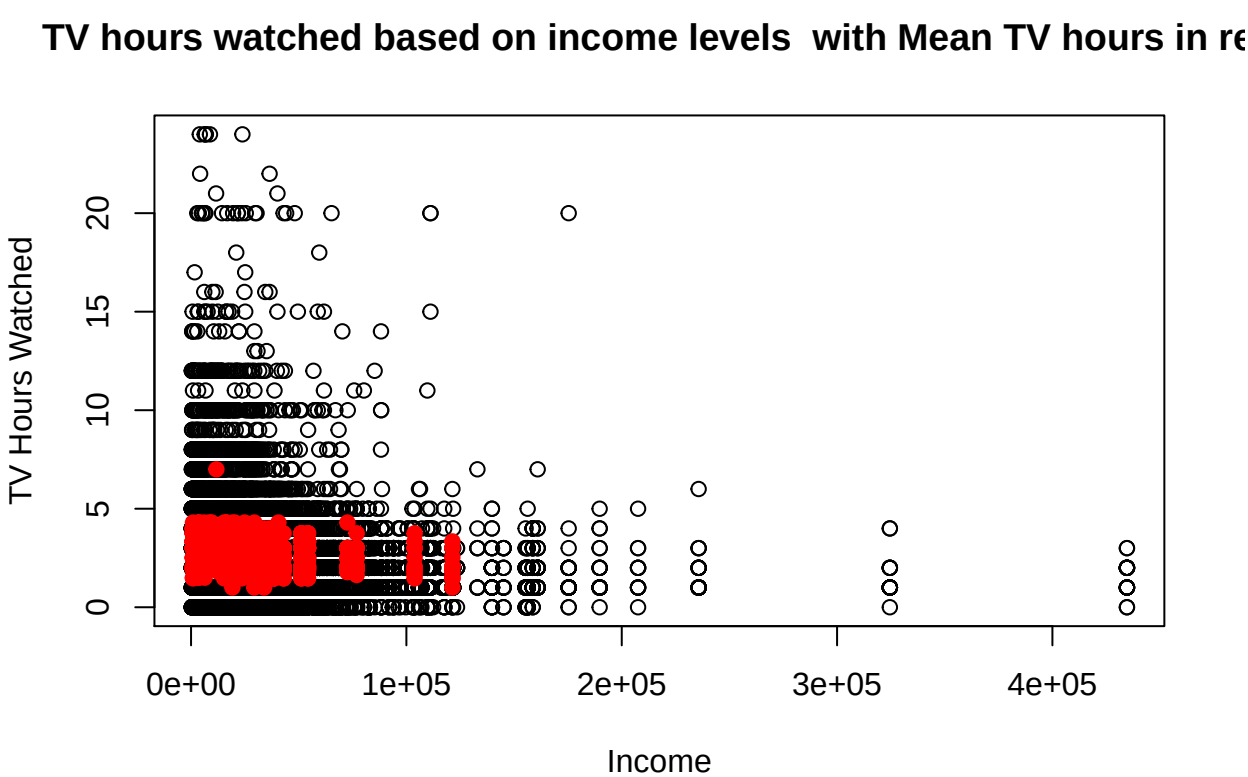
Results: The resulting plot shows TV hours watched to income for responses collected after 1980 for respondents who are under age 50.

3. Use the function `points`, which takes the same first two arguments as `plot` to add points to this scatterplot corresponding to the group mean for TV hours watched (y) versus individual income (x). Add the option `co="red"` to this command so that the "group mean" points are clearly visible.

```
## Create a scatterplot of TV hours watched (y) versus income (x) for all individuals in subGSS.2I2.

plot(subGSS.2I2$conrinc, subGSS.2I2$tvhours, type="p", main="TV hours watched based on income levels  wi

## Adding mean (x,y) value in solid red.
points(subGSS.2I2$conrinc, subGSS.2I2$MTVhours, pch = 19, col="red")
```



Results: The resulting plot shows TV hours watched to income for responses collected after 1980 for respondents who are under age 50. Overlayed on the original plot, is a plot of income to mean TV hours for each category (year, marital, degree) shown in red dots.

4. Construct a long data frame wherein you find the mean hours of TV watched and radio listened to for each subgroup of opinion of marijuana legalization and education achieved, only with respondents who make less than $50,000. Join this to the original data frame.

```
## Filter out records which are missing values for radiohrs and tvhours. Create subGSS.2I4 to hold data
subGSS.2I4 <- filter(subGSS.2, !is.na(radiohrs) & !is.na(tvhours))

## `filter` out respondents over 50 year.
subGSS.2I4 <- filter(subGSS.2I4, conrinc < 50000)

## Group data set by degree and legalization opinion, calculate mean tvhours for each level, calculate r
by_degree_grass <- subGSS.2I4 %>%
  group_by (degree, grass) %>%
  mutate (MTVhours = mean(tvhours, na.rm=TRUE), Mradiohrs = mean(radiohrs, na.rm=TRUE))

## display by_degree_grass
by_degree_grass
```

```
## Source: local data frame [2,442 x 13]
## Groups: degree, grass [16]
##
##     year     grass       marital   age         degree tvhours radiohrs
##    <int>    <fctr>        <fctr> <int>         <fctr>   <int>    <int>
```

```
## 1   1978 not legal       married    35 lt high school       4        8
## 2   1978 not legal       married    50    high school       3        6
## 3   1978     legal       married    41    high school       2        2
## 4   1978 not legal never married    19    high school       2        6
## 5   1978 not legal       married    59    high school       3        6
## 6   1978 not legal       married    49       bachelor       1        1
## 7   1978     legal never married    21    high school       1        4
## 8   1978     legal       married    52       graduate       1        1
## 9   1978     legal       married    53 lt high school       2        7
## 10  1978     legal       married    24    high school       2        5
## # ... with 2,432 more rows, and 6 more variables: racmar <fctr>,
## #   premarsx <fctr>, SSMar <fctr>, conrinc <dbl>, MTVhours <dbl>,
## #   Mradiohrs <dbl>
```

```
## Add MTVhours & Mradiohrs back into subGSS.2I4
subGSS.2I4 <- left_join(subGSS.2, by_degree_grass)
```

```
## Joining, by = c("year", "grass", "marital", "age", "degree", "tvhours", "radiohrs", "racmar", "prema
```

```
str(subGSS.2I4)
```

```
## 'data.frame':    59609 obs. of  13 variables:
##  $ year     : int  1972 1972 1972 1972 1972 1972 1972 1972 1972 1972 ...
##  $ grass    : Factor w/ 5 levels "iap","legal",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ marital  : Factor w/ 6 levels "married","widowed",..: 5 1 1 1 1 5 3 5 5 1 ...
##  $ age      : int  23 70 48 27 61 26 28 27 21 30 ...
##  $ degree   : Factor w/ 8 levels "lt high school",..: 4 1 2 4 2 2 2 4 2 2 ...
##  $ tvhours  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ radiohrs : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ racmar   : Factor w/ 5 levels "iap","yes","no",..: 3 2 2 2 3 3 2 3 NA NA ...
##  $ premarsx : Factor w/ 8 levels "iap","always wrong",..: 5 2 2 2 4 4 5 4 5 2 ...
##  $ SSMar    : Factor w/ 8 levels "iap","strongly agree",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ conrinc  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ MTVhours : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Mradiohrs: num  NA NA NA NA NA NA NA NA NA NA ...
```

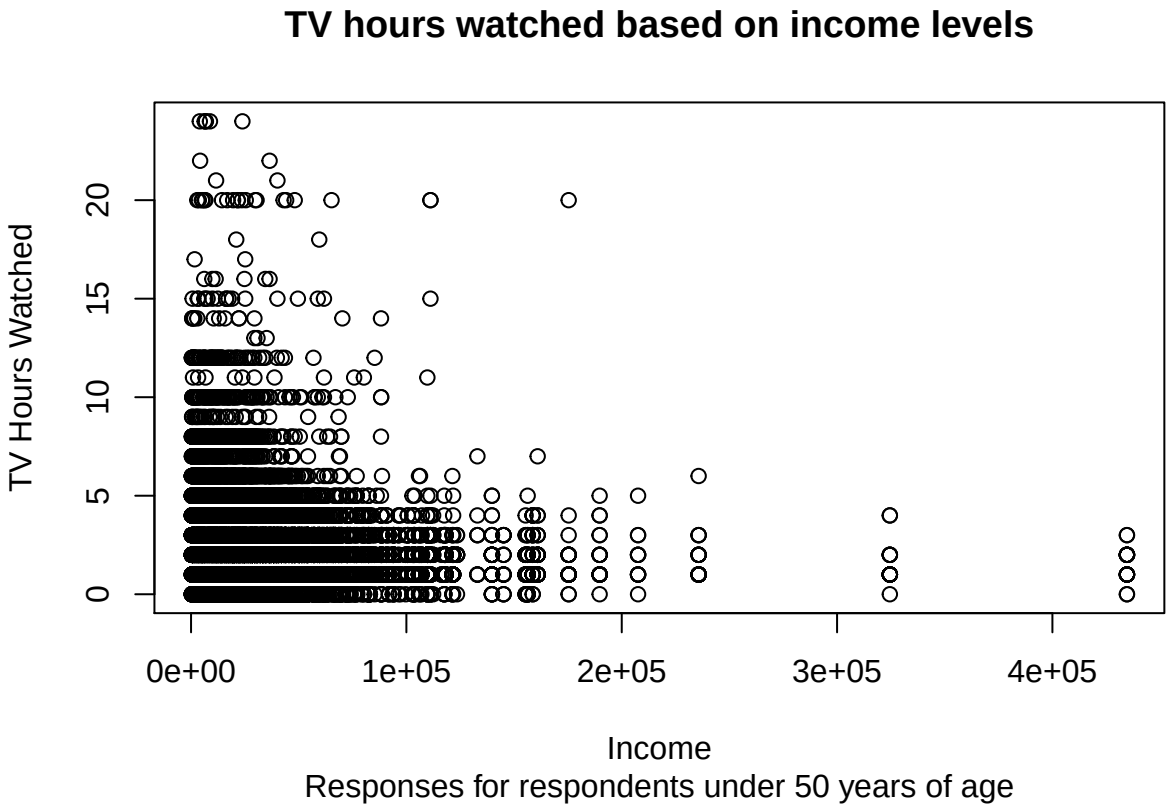
Results: The resulting plot shows TV hours watched to income for responses collected after 1980 for respondents who are under age 50.

5. Create a scatterplot using the `plot` function of TV hours watched (y) versus income (x) for all individuals. Add arguments `main`, `xlab` and `ylab` for a title and axis labels.

```
## Create a scatterplot of TV hours watched (y) versus income (x) for all individuals in subGSS.2I2.

## Remove rows with missing values for conrinc and assign to subGSS.2I2.
subGSS.2I4 <- filter(subGSS.2I4, !is.na(conrinc))

plot(subGSS.2I4$conrinc, subGSS.2I4$tvhours, type="p", main="TV hours watched based on income levels", s
```



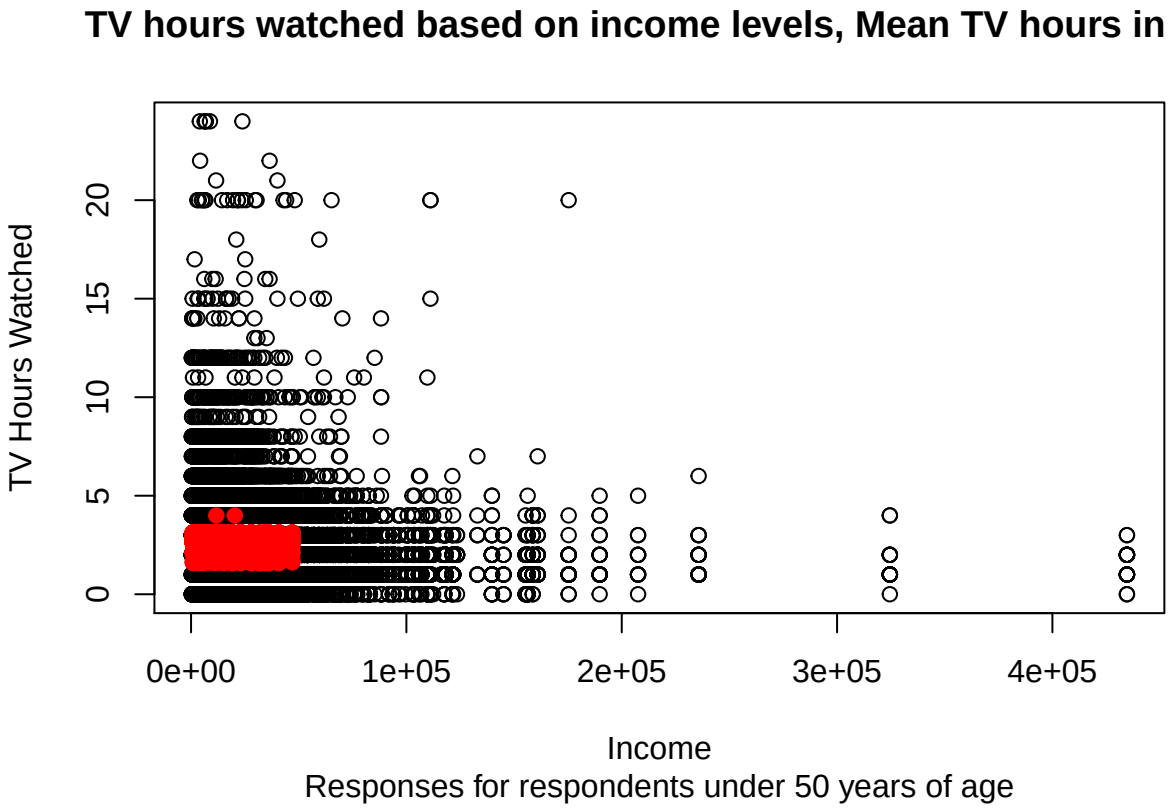
Results: The resulting plot shows TV hours watched to income for respondents who are under age 50.

6. Use the function `points`, which takes the same first two arguments as `plot` to add points to this scatterplot corresponding to the group mean for TV hours watched (y) versus individual income (x). Add the option `col="red"` to this command so that the "group mean" points are clearly visible.

```
## Create a scatterplot of TV hours watched (y) versus income (x) for all individuals in subGSS.2I2.

plot(subGSS.2I4$conrinc, subGSS.2I4$tvhours, type="p", main="TV hours watched based on income levels, M


## Adding mean (x,y) value in solid red.
points(subGSS.2I4$conrinc, subGSS.2I4$MTVhours, pch = 19, col="red")
```



Results: The resulting plot shows TV hours watched to income for respondents who are under age 50. Overlayed on the original plot, is a plot of income to mean TV hours for each category (year, marital, degree) shown in red dots.

Part II

We have class records for several different groups of people. There is an overlap among the rosters of the three classes and we wish to compose a complete grade sheet for all students. You will need to load the file `hw2students.RData` into your workspace as part of your script.

7. Join the three tables together into one. Which `join` function will you need to use to include all student names across all three tables?

```
## Load dplyr.
library(dplyr)
## Set working directory.
##setwd("C:/Users/Cheryl/OneDrive/Education/UF/2015/Fall/EDF6938/Week 3")
## Load data file.
load ("hw2students.RData")
## Join report.card.1, report.card.2, and report.card.3 together into report.card. Use full_join to reta
report.card <- full_join(report.card.1, report.card.2, by="Name")
```

```
## Warning in full_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
report.card <- full_join(report.card, report.card.3, by="Name")
```

```
## Warning in full_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factor and character vector, coercing into character vector
```

Results: The resulting report.card data frame contains all students with grades from all 9 courses as applicable.

8. Determine the mean grade achieved for all students in the classes they take. You will need to use `group_by`, `mutate` and `mean` while using a function to remove the missing values for each row. Add this to the data frame

```
## Using pipe & group_by to create mean for each row.
report.card.avg <- report.card %>% group_by(Name) %>% mutate(Avg = mean(c(Math, English, History, Biolo
## Display report.card.avg
report.card.avg
```

```
## Source: local data frame [198 x 11]
## Groups: Name [198]
##
##              Name  Math English History Biology Chemistry Physics
##             <chr> <int>   <int>   <int>   <int>     <int>   <int>
## 1      Abel Wolfe    81      82      83      84        89      82
## 2      Louis Dunn    75      86      80      NA        NA      NA
## 3      Glenda Fox    84      85      84      80        84      86
## 4    Homer Larson    85      75      76      68        84      72
## 5   Keith Alvarez    77      82      86      78        78      78
## 6   Sara Mcdonald    74      77      87      79        69      88
## 7     Irma Newton    82      78      84      79        73      74
## 8      Shelly Cox    85      80      74      80        80      78
## 9  Brenda Higgins    75      81      78      78        76      76
## 10    Vivian Rose    81      78      77      81        79      80
## # ... with 188 more rows, and 4 more variables: Semiotics <int>,
## #   Mixology <int>, Showmanship <int>, Avg <dbl>
```

```
str(report.card.avg, list.len = 12)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  198 obs. of  11 variables:
##  $ Name       : chr  "Abel Wolfe" "Louis Dunn" "Glenda Fox" "Homer Larson" ...
##  $ Math       : int  81 75 84 85 77 74 82 85 75 81 ...
##  $ English    : int  82 86 85 75 82 77 78 80 81 78 ...
##  $ History    : int  83 80 84 76 86 87 84 74 78 77 ...
##  $ Biology    : int  84 NA 80 68 78 79 79 80 78 81 ...
##  $ Chemistry  : int  89 NA 84 84 78 69 73 80 76 79 ...
##  $ Physics    : int  82 NA 86 72 78 88 74 78 76 80 ...
##  $ Semiotics  : int  77 82 NA 75 84 77 83 73 NA 75 ...
##  $ Mixology   : int  79 73 NA 77 82 77 78 81 NA 86 ...
##  $ Showmanship: int  75 77 NA 74 80 86 78 72 NA 75 ...
##  $ Avg        : num  81.3 78.8 83.8 76.2 80.6 ...
##  - attr(*, "vars")=List of 1
##   ..$ : symbol Name
##  - attr(*, "labels")='data.frame':   198 obs. of  1 variable:
##   ..$ Name: chr  "Abel Wolfe" "Alfonso Day" "Alice Farmer" "Allen Craig" ...
##   ..- attr(*, "vars")=List of 1
##   .. ..$ : symbol Name
##   ..- attr(*, "drop")= logi TRUE
##  - attr(*, "indices")=List of 198
##   ..$ : int 0
```

```
##   ..$ : int 40
##   ..$ : int 83
##   ..$ : int 169
##   ..$ : int 99
##   ..$ : int 22
##   ..$ : int 129
##   ..$ : int 172
##   ..$ : int 126
##   ..$ : int 153
##   ..$ : int 12
##   ..$ : int 136
##   .. [list output truncated]
##  - attr(*, "drop")= logi TRUE
##  - attr(*, "group_sizes")= int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "biggest_group_size")= int 1
```

```
head(report.card.avg$Avg)
```

```
## [1] 81.33333 78.83333 83.83333 76.22222 80.55556 79.33333
```

Results: The resulting report.card.avg contains a new column (Avg) with the average grade from all courses taken.

9. Determine the number of classes taken by each student and add this to the data frame. Find the top-achieving student in each group: those who took all 9 classes, only 6, and only 3.

```
## Using pipe & group_by to create count for each row.
ungroup(report.card.avg)
```

```
## # A tibble: 198 × 11
##              Name  Math English History Biology Chemistry Physics
##             <chr> <int>   <int>   <int>   <int>     <int>   <int>
## 1      Abel Wolfe    81      82      83      84        89      82
## 2      Louis Dunn    75      86      80      NA        NA      NA
## 3      Glenda Fox    84      85      84      80        84      86
## 4    Homer Larson    85      75      76      68        84      72
## 5   Keith Alvarez    77      82      86      78        78      78
## 6   Sara Mcdonald    74      77      87      79        69      88
## 7     Irma Newton    82      78      84      79        73      74
## 8      Shelly Cox    85      80      74      80        80      78
## 9  Brenda Higgins    75      81      78      78        76      76
## 10    Vivian Rose    81      78      77      81        79      80
## # ... with 188 more rows, and 4 more variables: Semiotics <int>,
## #   Mixology <int>, Showmanship <int>, Avg <dbl>
```

```
report.card.classes <-
  report.card.avg %>%
  mutate(Classes = (sum(c(Math, English, History, Biology, Chemistry, Physics, Semiotics, Mixology, Sho
top_n(group_by(report.card.classes, Classes), 1, Avg)
```

```
## Source: local data frame [3 x 12]
## Groups: Classes [3]
##
##            Name  Math English History Biology Chemistry Physics Semiotics
##           <chr> <int>   <int>   <int>   <int>     <int>   <int>     <int>
## 1    Glenda Fox    84      85      84      80        84      86        NA
## 2 Hazel Hubbard    88      86      83      NA        NA      NA        NA
```

```
## 3  Lance Holmes    79      86      86      84        83      85        86
## # ... with 4 more variables: Mixology <int>, Showmanship <int>, Avg <dbl>,
## #   Classes <dbl>
```

```
str(report.card.avg, list.len = 12)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  198 obs. of  11 variables:
##  $ Name       : chr  "Abel Wolfe" "Louis Dunn" "Glenda Fox" "Homer Larson" ...
##  $ Math       : int  81 75 84 85 77 74 82 85 75 81 ...
##  $ English    : int  82 86 85 75 82 77 78 80 81 78 ...
##  $ History    : int  83 80 84 76 86 87 84 74 78 77 ...
##  $ Biology    : int  84 NA 80 68 78 79 79 80 78 81 ...
##  $ Chemistry  : int  89 NA 84 84 78 69 73 80 76 79 ...
##  $ Physics    : int  82 NA 86 72 78 88 74 78 76 80 ...
##  $ Semiotics  : int  77 82 NA 75 84 77 83 73 NA 75 ...
##  $ Mixology   : int  79 73 NA 77 82 77 78 81 NA 86 ...
##  $ Showmanship: int  75 77 NA 74 80 86 78 72 NA 75 ...
##  $ Avg        : num  81.3 78.8 83.8 76.2 80.6 ...
##  - attr(*, "vars")=List of 1
##   ..$ : symbol Name
##  - attr(*, "labels")='data.frame':   198 obs. of  1 variable:
##   ..$ Name: chr  "Abel Wolfe" "Alfonso Day" "Alice Farmer" "Allen Craig" ...
##   ..- attr(*, "vars")=List of 1
##   .. ..$ : symbol Name
##   ..- attr(*, "drop")= logi TRUE
##  - attr(*, "indices")=List of 198
##   ..$ : int 0
##   ..$ : int 40
##   ..$ : int 83
##   ..$ : int 169
##   ..$ : int 99
##   ..$ : int 22
##   ..$ : int 129
##   ..$ : int 172
##   ..$ : int 126
##   ..$ : int 153
##   ..$ : int 12
##   ..$ : int 136
##   .. [list output truncated]
##  - attr(*, "drop")= logi TRUE
##  - attr(*, "group_sizes")= int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "biggest_group_size")= int 1
```

Results: The results produce a list of the student with the highest average grade from each level of participation (3 courses, 6 courses, 9 courses).