Hello World, part 2

Load relevant libraries

```
## Get necessary libraries-- won't work the first time, because you need to install them!
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     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()
```

Load The Data

```
df<-readRDS("sc_debt.Rds")
names(df)
```

```
##  [1] "unitid"         "instnm"         "stabbr"         "grad_debt_mdn"
##  [5] "control"        "region"         "preddeg"        "openadmp"
##  [9] "adm_rate"       "ccbasic"        "sat_avg"        "md_earn_wne_p6"
## [13] "ugds"           "selective"      "research_u"
```

Name	Definition
unitid	Unit ID
instnm	Institution Name
stabbr	State Abbreviation
grad_debt_mdn	Median Debt of Graduates
control	Control Public or Private
region	Census Region
preddeg	Predominant Degree Offered: Associates or Bachelors
openadmp	Open Admissions Policy: 1= Yes, 2=No,3=No 1st time students
adm_rate	Admissions Rate: proportion of applications accepted
ccbasic	Type of institution– see here
selective	Institution admits fewer than 10 % of applicants, 1=Yes, 0=No
research_u	Instituition is a research university 1=Yes, 0=No
sat_avg	Average Sat Scores
md_earn_wne_p6	Average Earnings of Recent Graduates
ugds	Number of undergraduates

Looking at datasets

We can use "glimpse" to see what's in a dataset

```
glimpse(df)
```

```
## Rows: 2,555
## Columns: 15
## $ unitid         <int> 132657, 130217, 132851, 135364, 135391, 134097, 135717,~
## $ instnm         <chr> "Lynn University", "Quinebaug Valley Community College"~
## $ stabbr         <chr> "FL", "CT", "FL", "GA", "FL", "FL", "FL", "GA", "FL", "~
## $ grad_debt_mdn  <dbl> 17556, NA, 13140, 29875, 10413, 19002, 9500, 27000, 920~
## $ control        <chr> "Private", "Public", "Public", "Private", "Public", "Pu~
## $ region         <chr> "Soutwest", "New England", "Soutwest", "Soutwest", "Sou~
## $ preddeg        <chr> "Bachelor's", "Associate", "Associate", "Bachelor's", "~
## $ openadmp       <int> 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2~
## $ adm_rate       <dbl> 0.7045, NA, NA, 0.5909, NA, 0.3678, NA, 0.5185, NA, 0.4~
## $ ccbasic        <int> 18, 1, 23, 24, 23, 15, 23, 16, 14, 18, 14, 15, 20, 19, ~
## $ sat_avg        <dbl> 1086, NA, NA, NA, NA, 1289, NA, 1032, NA, 1189, NA, 126~
## $ md_earn_wne_p6 <int> 33400, 22300, 24700, 37000, 25600, 36500, 25700, 26700,~
## $ ugds           <int> 2194, 1236, 5711, 293, 8203, 32072, 49443, 3320, 38107,~
## $ selective      <dbl> 0, NA, NA, 0, NA, 0, NA, 0, NA, 0, NA, 0, 0, 0, 0, 0, 0~
## $ research_u     <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0~
```

Filter, Select, Arrange

In exploring data, many times we want to look at smaller parts of the dataset. There are three commands we'll use today that help with this.

-`filter` selects only those cases or rows that meet some logical criteria.

-`select` selects only those variables or columns that meet some criteria

-`arrange` arranges the rows of a dataset in the way we want.

For more on these, please see this vignette.

We can look at the first 5 rows:

```
head(df)
```

```
## # A tibble: 6 x 15
##   unitid instnm    stabbr grad_debt_mdn control region preddeg openadmp adm_rate
##    <int> <chr>     <chr>          <dbl> <chr>   <chr>  <chr>      <int>    <dbl>
## 1 132657 Lynn Uni~ FL             17556 Private Soutw~ Bachel~        2    0.704
## 2 130217 Quinebau~ CT                NA Public  New E~ Associ~        1   NA
## 3 132851 College ~ FL             13140 Public  Soutw~ Associ~        1   NA
## 4 135364 Luther R~ GA             29875 Private Soutw~ Bachel~        2    0.591
## 5 135391 State Co~ FL             10413 Public  Soutw~ Associ~        1   NA
## 6 134097 Florida ~ FL             19002 Public  Soutw~ Bachel~        2    0.368
## # ... with 6 more variables: ccbasic <int>, sat_avg <dbl>,
## #   md_earn_wne_p6 <int>, ugds <int>, selective <dbl>, research_u <dbl>
```

Or the last 5 rows:

```
tail(df)
```

```
## # A tibble: 6 x 15
##   unitid instnm    stabbr grad_debt_mdn control region preddeg openadmp adm_rate
##    <int> <chr>     <chr>          <dbl> <chr>   <chr>  <chr>      <int>    <dbl>
## 1 174066 Universi~ MN             19500 Public  Plains Bachel~        2    0.519
## 2 218229 Lander U~ SC             26000 Public  Soutw~ Bachel~        2    0.426
```

```
## 3 217040 Yeshivat~ PA               NA Private North~ Bachel~        2    0.909
## 4 217059 York Col~ PA            26950 Private North~ Bachel~        2    0.697
## 5 218353 Midlands~ SC            11707 Public  Soutw~ Associ~        1   NA
## 6 218399 Morris C~ SC            31775 Private Soutw~ Bachel~        1   NA
## # ... with 6 more variables: ccbasic <int>, sat_avg <dbl>,
## #   md_earn_wne_p6 <int>, ugds <int>, selective <dbl>, research_u <dbl>
```

Using filter in combination with other commands

`filter` can be used with any command that retruns true or false. This can be really powerful, for instance the command `str_detect` "detects" the relevant string in the data, so we can look for any college with the word "Colorado" in its name.

```
df%>%
  filter(str_detect(instnm,"Colorado"))%>%
  select(instnm,adm_rate,sat_avg)
```

```
## # A tibble: 12 x 3
##    instnm                                              adm_rate sat_avg
##    <chr>                                                  <dbl>   <dbl>
##  1 University of Colorado Boulder                         0.815    1281
##  2 Western Colorado University                            0.889    1095
##  3 University of Northern Colorado                        0.908    1099
##  4 Colorado State University-Global Campus                 NA        NA
##  5 Colorado School of Mines                               0.492    1383
##  6 Colorado State University-Fort Collins                 0.839    1197
##  7 University of Colorado Denver/Anschutz Medical Campus  0.636    1132
##  8 Colorado College                                       0.15       NA
##  9 Colorado State University-Pueblo                       0.952    1040
## 10 University of Colorado Colorado Springs                0.912    1123
## 11 Colorado Christian University                          NA        NA
## 12 Colorado Mesa University                               0.807    1022
```

Reminder: logical operators

- `>`, `<`: greater than, less than
- `>=`, `<=`: greater than or equal to, less than or equal to
- `!` :not, as in `!=` not equal to
- `&` AND
- `|` OR

Quick Exercise: Select Colleges with either Colorado OR California in their names

Extending Select

Select can also be used with other characteristics.

For quick guide on this: https://dplyr.tidyverse.org/reference/select.html

```
df%>%
  select(contains("region"))
```

```
## # A tibble: 2,555 x 1
##    region
##    <chr>
##  1 Soutwest
##  2 New England
```

```
##  3 Soutwest
##  4 Soutwest
##  5 Soutwest
##  6 Soutwest
##  7 Soutwest
##  8 Soutwest
##  9 Soutwest
## 10 Soutwest
## # ... with 2,545 more rows
```

To select only numeric variables

```
df%>%
  select(where(is.numeric))
```

```
## # A tibble: 2,555 x 10
##    unitid grad_debt_mdn openadmp adm_rate ccbasic sat_avg md_earn_wne_p6  ugds
##     <int>         <dbl>    <int>    <dbl>   <int>   <dbl>          <int> <int>
##  1 132657         17556        2    0.704      18    1086          33400  2194
##  2 130217            NA        1   NA           1      NA          22300  1236
##  3 132851         13140        1   NA          23      NA          24700  5711
##  4 135364         29875        2    0.591      24      NA          37000   293
##  5 135391         10413        1   NA          23      NA          25600  8203
##  6 134097         19002        2    0.368      15    1289          36500 32072
##  7 135717          9500        1   NA          23      NA          25700 49443
##  8 138947         27000        2    0.518      16    1032          26700  3320
##  9 138187          9208        1   NA          14      NA          26100 38107
## 10 138354         17250        2    0.424      18    1189          31500  9371
## # ... with 2,545 more rows, and 2 more variables: selective <dbl>,
## #   research_u <dbl>
```

Quick Exercise Use the same setup to select only character variables (`is.character`)

Summarizing Data

To summarize data, we use the summarize command. Inside that command, we tell R two things: what to call the new object (a data frame, really) that we're creating, and what numerical summary we would like. The code below summarizes median debt for the colleges in the dataset by calculating the average of median debt for all institutions.

```
df%>%
  summarize(mean_debt=mean(grad_debt_mdn,na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   mean_debt
##       <dbl>
## 1    19662.
```

```
df%>%
  summarize(median_debt=median(grad_debt_mdn,na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   median_debt
##         <dbl>
## 1       21500
```

Quick Exercise Summarize the average entering SAT scores in this dataset.

Combining Commands

We can also combine commands, so that summaries are done on only a part of the dataset. Below, we summarize median debt for selective schools, and not very selective schools.

```
df%>%
  filter(stabbr=="CA")%>%
  summarize(mean_adm_rate=mean(adm_rate,na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   mean_adm_rate
##           <dbl>
## 1         0.577
```

Quick Exercise Calculate average earnings for schools where SAT>1200 & the admissions rate is between 10 and 20 percent.

Mutate

`mutate` is the verb for changing variables in R. Let's say we want to create a variable that's set to 1 if the college admits less than 10 percent of the students who apply.

```
df<-df%>%
  mutate(selective=ifelse(adm_rate<=10,1,0))
```

Or what if we want to create another new variable that changes the admissions rate from its current proportion to a percent?

```
df<-df%>%
  mutate(adm_rate_pct=adm_rate*100)
```

To figure out if that worked we can use `summarize`

```
df%>%
  summarize(mean_adm_rate_pct=mean(adm_rate_pct,na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   mean_adm_rate_pct
##               <dbl>
## 1              66.7
```

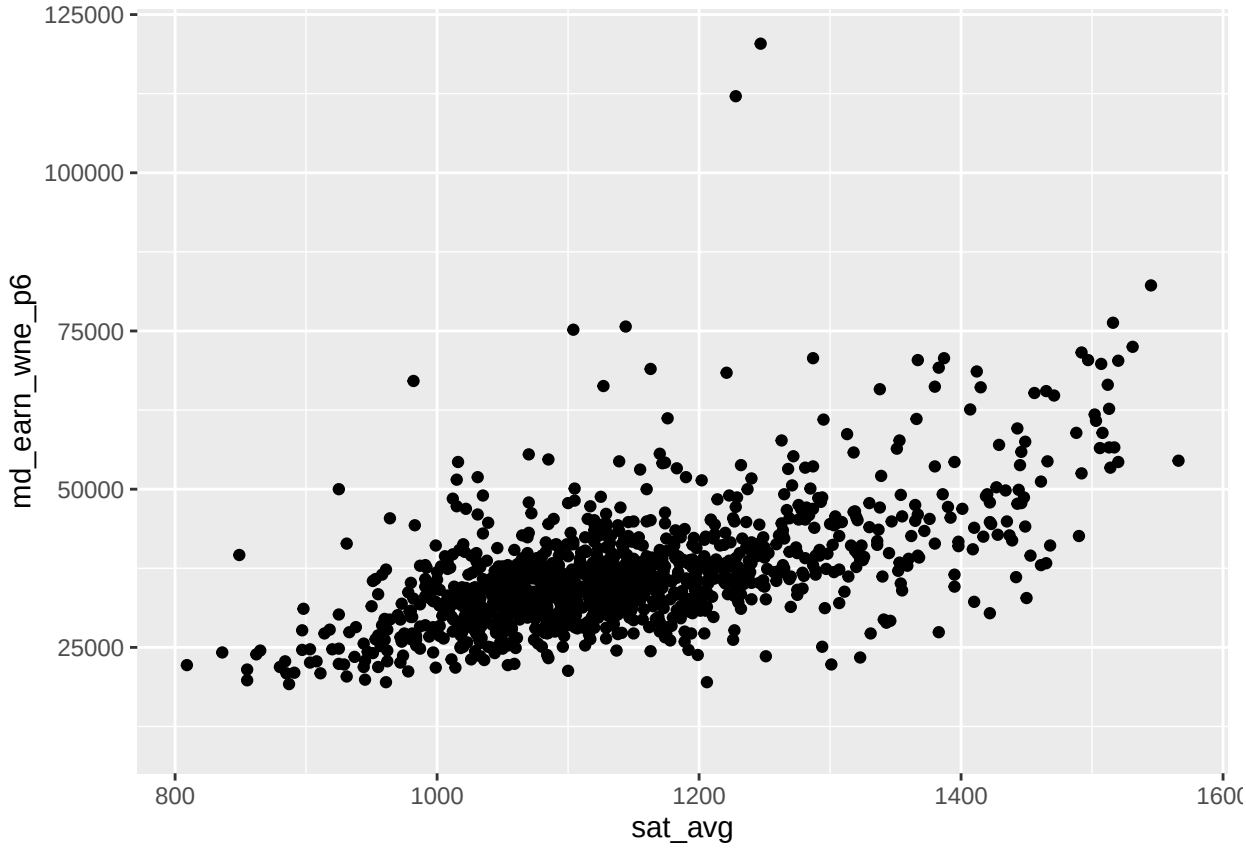
Quick exercise: create a new variable that's set to 1 if the college has more than 10,000 undergraduate students

Plotting Data

Below is a plot of earnings of by SAT scores

```
## Plotting: bivariate
df%>%
  ggplot(aes(x=sat_avg,y=md_earn_wne_p6))+
  geom_point()
```

```
## Warning: Removed 1355 rows containing missing values (geom_point).
```



Quick exercise: plot earnings by admission rate only for schools in California

Quick exercise Replicate the above plots, but put debt level on the y axis.