Advanced dplyr and More on RMarkdown

Daniel Anderson

Reviewing what we know

Talk with your neighbor: How would you do each of the following using *dplyr*?

- Extract only the first and third through ninth columns
- Extract the third column and any column that ends with "response"
- Reorder columns
- Extract only cases from students classified as English Language Learners (ELL)
- Create a new variable that represents the proportion of ELL students by grade
- Order the data in ascending or descending order
- Center a variable (i.e., subtract the mean from each observation)
- Compute the mean and standard deviation of a variable by one or more groups (e.g., mean math score for each school)

First, let's load in some data

```
library(rio)
mcrc <- import("../../data/mcrc.csv", setclass = "tbl_df", na = ".") %>%
  janitor::clean_names()
mcrc
```

```
## # A tibble: 72,948 x 47
##       id schid                        race           ethnicity   ell
##    <int> <int>                       <chr>               <chr> <chr>
##  1  8307   381                       White Not Hispanic/Latino
##  2  8308   449                             Not Hispanic/Latino
##  3  8309   817                       White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                       White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino     Y
##  7  8313   529                       White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                     Unknown             Unknown
## # ... with 72,938 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
```

- Extract only the first and third through ninth columns

```
library(dplyr)
mcrc %>%
  select(1, 3:9)
```

```
## # A tibble: 72,948 x 8
##       id                        race           ethnicity   ell disability
##    <int>                       <chr>               <chr> <chr>      <chr>
##  1  8307                       White Not Hispanic/Latino
##  2  8308                             Not Hispanic/Latino
##  3  8309                       White Not Hispanic/Latino
##  4  8310 Black or African American Not Hispanic/Latino
##  5  8311                       White Not Hispanic/Latino
##  6  8312 Black or African American     Hispanic/Latino     Y
##  7  8313                       White Not Hispanic/Latino
##  8  8314 Black or African American Not Hispanic/Latino
##  9  8315 Black or African American Not Hispanic/Latino
## 10  8316                    Unknown             Unknown
## # ... with 72,938 more rows, and 3 more variables: score <int>,
## #   item_1_response <int>, item_1_correct <chr>
```

- Extract the third column and any column that ends with "response"

```
mcrc %>%
  select(3, ends_with("response"))
```

```
## # A tibble: 72,948 x 21
##                          race item_1_response item_2_response
##                         <chr>           <int>           <int>
##  1                      White               3               1
##  2                                          3               2
##  3                      White               3               1
##  4 Black or African American               3               1
##  5                      White               3               1
##  6 Black or African American               3               1
##  7                      White               3               1
##  8 Black or African American               1               1
##  9 Black or African American               1               1
## 10                    Unknown               3               1
## # ... with 72,938 more rows, and 18 more variables: item_3_response <int>,
## #   item_4_response <int>, item_5_response <int>, item_6_response <int>,
## #   item_7_response <int>, item_8_response <int>, item_9_response <int>,
## #   item_10_response <int>, item_11_response <int>,
## #   item_12_response <int>, item_13_response <int>,
```

- Reorder columns

```
mcrc %>%
  select(1:6, ends_with("correct"), ends_with("response"))
```

```
## # A tibble: 72,948 x 46
##       id schid                      race           ethnicity   ell
##    <int> <int>                     <chr>               <chr> <chr>
##  1  8307   381                     White Not Hispanic/Latino
##  2  8308   449                           Not Hispanic/Latino
##  3  8309   817                     White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                     White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino     Y
##  7  8313   529                     White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                   Unknown             Unknown
## # ... with 72,938 more rows, and 41 more variables: disability <chr>,
## #   item_1_correct <chr>, item_2_correct <chr>, item_3_correct <chr>,
## #   item_4_correct <chr>, item_5_correct <chr>, item_6_correct <chr>,
## #   item_7_correct <chr>, item_8_correct <chr>, item_9_correct <chr>,
## #   item_10_correct <chr>, item_11_correct <chr>, item_12_correct <chr>,
```

- Extract only cases from students classified as English Language Learners

```
mcrc %>%
  filter(ell == "Y")
```

```
## # A tibble: 5,687 x 47
##       id schid                                race           ethnicity   ell
##    <int> <int>                                <chr>              <chr> <chr>
##  1  8312   410         Black or African American     Hispanic/Latino     Y
##  2  8323    32                                White Not Hispanic/Latino     Y
##  3  8326   520                                White     Hispanic/Latino     Y
##  4  8341   196 American Indian or Alaskan Native     Hispanic/Latino     Y
##  5  8388   620                                White     Hispanic/Latino     Y
##  6  8405   844                                White     Hispanic/Latino     Y
##  7  8409   585                                White Not Hispanic/Latino     Y
##  8  8417   390                                White Not Hispanic/Latino     Y
##  9  8442   851                                      Not Hispanic/Latino     Y
## 10  8459   374                                White Not Hispanic/Latino     Y
## # ... with 5,677 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
## #   item_5_response <int>, item_5_correct <chr>, item_6_response <int>,
```

- Create a new variable that represents the proportion of ELL students by grade

```
mcrc %>%
  group_by(schid) %>%
  mutate(prop_ell = mean(ell == "Y", na.rm = TRUE)) %>%
  select(1:2, prop_ell)
```

```
## # A tibble: 72,948 x 3
## # Groups:   schid [1,000]
##       id schid   prop_ell
##    <int> <int>      <dbl>
##  1  8307   381 0.07500000
##  2  8308   449 0.05747126
##  3  8309   817 0.07352941
##  4  8310   979 0.01408451
##  5  8311    13 0.08450704
##  6  8312   410 0.05128205
##  7  8313   529 0.16923077
##  8  8314   967 0.06818182
##  9  8315   589 0.07407407
## 10  8316   257 0.04347826
## # ... with 72,938 more rows
```

- Order the data in ascending order

```
mcrc %>%
  group_by(schid) %>%
  mutate(prop_ell = mean(ell == "Y", na.rm = TRUE)) %>%
  select(1:2, prop_ell) %>%
  arrange(schid)
```

```
## # A tibble: 72,948 x 3
## # Groups:   schid [1,000]
##       id schid   prop_ell
##    <int> <int>      <dbl>
##  1  6644     1 0.01408451
##  2 20444     1 0.01408451
##  3  2065     1 0.01408451
##  4 21933     1 0.01408451
##  5 22696     1 0.01408451
##  6 11418     1 0.01408451
##  7  6254     1 0.01408451
##  8 25539     1 0.01408451
##  9 26493     1 0.01408451
## 10 25447     1 0.01408451
## # ... with 72,938 more rows
```

- Order the data in descending order

```
mcrc %>%
  group_by(schid) %>%
  mutate(prop_ell = mean(ell == "Y", na.rm = TRUE)) %>%
  select(1:2, prop_ell) %>%
  arrange(desc(prop_ell))
```

```
## # A tibble: 72,948 x 3
## # Groups:   schid [1,000]
##       id schid  prop_ell
##    <int> <int>     <dbl>
##  1 20102   369 0.2083333
##  2 20183   369 0.2083333
##  3  9954   369 0.2083333
##  4 10028   369 0.2083333
##  5   848   369 0.2083333
##  6  4032   369 0.2083333
##  7  2038   369 0.2083333
##  8  2318   369 0.2083333
##  9 22285   369 0.2083333
## 10 11200   369 0.2083333
## # ... with 72,938 more rows
```

Center a variable (i.e., subtract the mean from each observation)

```
mcrc %>%
  mutate(score = scale(score, scale = FALSE)) %>%
  select(1:2, score)
```

```
## # A tibble: 72,948 x 3
##       id schid      score
##    <int> <int>      <dbl>
##  1  8307   381 -0.3585705
##  2  8308   449  1.6414295
##  3  8309   817  3.6414295
##  4  8310   979 -2.3585705
##  5  8311    13  3.6414295
##  6  8312   410  3.6414295
##  7  8313   529  0.6414295
##  8  8314   967 -5.3585705
##  9  8315   589 -6.3585705
## 10  8316   257  2.6414295
## # ... with 72,938 more rows
```

Quick note on `mutate()`

You can create a variable with `mutate()`, and then immediately call upon it for an additional calculation on the next line.

```
mcrc %>%
  group_by(schid) %>%
  mutate(sch_mean = mean(score, na.rm = TRUE),
         group_centered = score - sch_mean) %>%
  select(1:2, sch_mean, score, group_centered)
```

```
## # A tibble: 72,948 x 5
## # Groups:   schid [1,000]
##       id schid sch_mean score group_centered
##    <int> <int>    <dbl> <int>          <dbl>
##  1  8307   381 13.96250    13      -0.962500
##  2  8308   449 13.67816    15       1.321839
##  3  8309   817 13.38235    17       3.617647
##  4  8310   979 13.12676    11      -2.126761
##  5  8311    13 13.33803    17       3.661972
##  6  8312   410 12.92308    17       4.076923
##  7  8313   529 13.20000    14       0.800000
##  8  8314   967 13.43182     8      -5.431818
```

Compute the mean and standard deviation of a variable by one or more groups (e.g., mean math score for each school)
(I randomly assigned school ids, so the fact they are very similar is not terribly surprising)

```
mcrc %>%
  group_by(schid) %>%
  summarize(mean = mean(score, na.rm = TRUE),
            sd = sd(score, na.rm = TRUE))
```

```
## # A tibble: 1,000 x 3
##    schid     mean       sd
##    <int>    <dbl>    <dbl>
##  1     1 13.45070 3.790170
##  2     2 13.47945 4.140282
##  3     3 13.48438 3.690366
##  4     4 12.27419 4.316075
##  5     5 13.36000 4.138285
##  6     6 13.36709 3.800058
##  7     7 13.81081 3.662755
##  8     8 13.62069 4.020995
##  9     9 13.88889 3.605768
## 10    10 13.61290 3.856198
## # ... with 990 more rows
```

Other dplyr features

slice()

Extracts rows, like `filter()`, but does it by indexing

- Extract the first five rows

```
mcrc %>%
  slice(1:5)
```

```
## # A tibble: 5 x 47
##      id schid                          race           ethnicity   ell
##   <int> <int>                         <chr>               <chr> <chr>
## 1  8307   381                         White Not Hispanic/Latino
## 2  8308   449                               Not Hispanic/Latino
## 3  8309   817                         White Not Hispanic/Latino
## 4  8310   979 Black or African American Not Hispanic/Latino
## 5  8311    13                         White Not Hispanic/Latino
## # ... with 42 more variables: disability <chr>, score <int>,
## #   item_1_response <int>, item_1_correct <chr>, item_2_response <int>,
## #   item_2_correct <chr>, item_3_response <int>, item_3_correct <chr>,
## #   item_4_response <int>, item_4_correct <chr>, item_5_response <int>,
## #   item_5_correct <chr>, item_6_response <int>, item_6_correct <chr>,
## #   item_7_response <int>, item_7_correct <chr>, item_8_response <int>,
```

- Remove the first five rows
- Extract the first five rows

```
mcrc %>%
  slice(-1:-5)
```

```
## # A tibble: 72,943 x 47
##       id schid                      race           ethnicity   ell
##    <int> <int>                     <chr>               <chr> <chr>
##  1  8312   410 Black or African American     Hispanic/Latino     Y
##  2  8313   529                     White Not Hispanic/Latino
##  3  8314   967 Black or African American Not Hispanic/Latino
##  4  8315   589 Black or African American Not Hispanic/Latino
##  5  8316   257                   Unknown             Unknown
##  6  8317   277                     White Not Hispanic/Latino
##  7  8318    55                     White Not Hispanic/Latino
##  8  8319   253                     White Not Hispanic/Latino
##  9  8320   633                     White Not Hispanic/Latino
## 10  8321   215                     White Not Hispanic/Latino
## # ... with 72,933 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
```

distinct()

Select rows that are distinct, according to a specific variable

```
mcrc %>%
  distinct(schid)
```

```
## # A tibble: 1,000 x 1
##    schid
##    <int>
##  1   381
##  2   449
##  3   817
##  4   979
##  5    13
##  6   410
##  7   529
##  8   967
##  9   589
## 10   257
## # ... with 990 more rows
```

Why is this helpful

- Probably most frequently used for removing duplicate records, or for joins, which we haven't talked about yet.
- Also can be used to understand your data better
  - Number of rows from you output will be the number of distinct values (e.g., number of schools)
  - Can use distinct on multiple variable
- How many unique race/ethnicity combinations are there?

```
mcrc %>%
  distinct(race, ethnicity)
```

```
## # A tibble: 32 x 2
##                                    race           ethnicity
##                                   <chr>               <chr>
##  1                                White Not Hispanic/Latino
##  2                                      Not Hispanic/Latino
##  3         Black or African American Not Hispanic/Latino
##  4         Black or African American     Hispanic/Latino
##  5                           Unknown             Unknown
```

Why is this helpful

- Remove duplicate records (primary reason I find it helpful)
    - optional `.keep_all` argument

```
nrow(mcrc)
```

```
## [1] 72948
```

```
mcrc %>%
  distinct(id, .keep_all = TRUE)
```

```
## # A tibble: 36,474 x 47
##       id schid                      race           ethnicity   ell
##    <int> <int>                     <chr>               <chr> <chr>
##  1  8307   381                     White Not Hispanic/Latino
##  2  8308   449                           Not Hispanic/Latino
##  3  8309   817                     White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                     White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino     Y
##  7  8313   529                     White Not Hispanic/Latino
```

Keep all unique combinations of student/school ids

```
nrow(mcrc)
```

```
## [1] 72948
```

```
mcrc %>%
  distinct(id, schid, .keep_all = TRUE)
```

```
## # A tibble: 72,916 x 47
##       id schid                      race           ethnicity   ell
##    <int> <int>                     <chr>               <chr> <chr>
##  1  8307   381                     White Not Hispanic/Latino
##  2  8308   449                           Not Hispanic/Latino
##  3  8309   817                     White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                     White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino     Y
##  7  8313   529                     White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                   Unknown             Unknown
## # ... with 72,906 more rows, and 42 more variables: disability <chr>,
```

`count()`

If want to know how many cases are within a grouping, use `count` (almost always in conjunction with `group_by`)

```
mcrc %>%
  count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 72948
```

```
mcrc %>%
  group_by(schid) %>%
  count()
```

```
## # A tibble: 1,000 x 2
## # Groups:   schid [1,000]
##    schid     n
##    <int> <int>
##  1     1    71
##  2     2    73
##  3     3    64
##  4     4    62
##  5     5    75
##  6     6    79
##  7     7    74
##  8     8    87
##  9     9    72
## 10    10    93
## # ... with 990 more rows
```

What are the number of ELL and non-ELL student for each school?

```
mcrc %>%
  group_by(schid, ell) %>%
  count()
```

```
## # A tibble: 1,997 x 3
## # Groups:   schid, ell [1,997]
##    schid   ell     n
##    <int> <chr> <int>
##  1     1           70
##  2     1     Y      1
##  3     2           69
##  4     2     Y      4
##  5     3           59
##  6     3     Y      5
##  7     4           56
##  8     4     Y      6
##  9     5           70
## 10     5     Y      5
## # ... with 1,987 more rows
```

Use tools to make it more readable

```
mcrc %>%
  mutate(ell = ifelse(ell == "", "N", ell)) %>%
  group_by(schid, ell) %>%
  count() %>%
  tidyr::spread(ell, n)
```

```
## # A tibble: 1,000 x 3
## # Groups:   schid [1,000]
##    schid     N     Y
##  * <int> <int> <int>
##  1     1    70     1
##  2     2    69     4
##  3     3    59     5
##  4     4    56     6
##  5     5    70     5
##  6     6    72     7
##  7     7    67     7
##  8     8    81     6
##  9     9    69     3
## 10    10    89     4
## # ... with 990 more rows
```

Random Sampling

Randomly sample 100 observations

```
mcrc %>%
  sample_n(100)
```

```
## # A tibble: 100 x 47
##       id schid                                race           ethnicity   ell
##    <int> <int>                                <chr>              <chr> <chr>
##  1 19848   224 American Indian or Alaskan Native     Hispanic/Latino
##  2 11436   676         Black or African American Not Hispanic/Latino
##  3 11165   171                                White     Hispanic/Latino     Y
##  4 33492   488         Black or African American Not Hispanic/Latino
##  5 23018   148                            Unknown              Unknown
##  6 21701   901                                White Not Hispanic/Latino
##  7 30001   606                                White Not Hispanic/Latino
##  8 27983   776                                White Not Hispanic/Latino
##  9  1261   178                                White     Hispanic/Latino
## 10 18959   394                                White Not Hispanic/Latino
## # ... with 90 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
```

Randomly sample 5 observations from each school

```
mcrc %>%
  group_by(schid) %>%
  sample_n(5)
```

```
## # A tibble: 5,000 x 47
## # Groups:   schid [1,000]
##       id schid                                race           ethnicity   ell
##    <int> <int>                                <chr>              <chr> <chr>
##  1 24677     1                                White Not Hispanic/Latino
##  2 36036     1                                White     Hispanic/Latino     Y
##  3 27080     1 American Indian or Alaskan Native Not Hispanic/Latino
##  4 10263     1                                White              Unknown
##  5 27977     1                              Unknown              Unknown
##  6 11802     2                                White Not Hispanic/Latino
##  7  2773     2                              Unknown     Hispanic/Latino
##  8 10515     2                  Two or more races     Hispanic/Latino
##  9 10251     2                                White Not Hispanic/Latino
## 10 24026     2                                White Not Hispanic/Latino
## # ... with 4,990 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
```

Slightly more compilcated

How do we randomly select schools? (and keep all the data from each school)

- One method

```
schools <- mcrc %>%
  distinct(schid) %>%
  sample_n(10)
schools
```

```
## # A tibble: 10 x 1
##    schid
##    <int>
##  1    24
##  2   555
##  3   736
##  4   243
##  5   767
##  6   820
##  7    92
##  8   782
```

(there's actually a better way to do this, through a `semi_join`, but again, we haven't talked about joins yet)

```
mcrc %>%
  filter(schid %in% schools$schid)
```

```
## # A tibble: 732 x 47
##       id schid                                race           ethnicity   ell
##    <int> <int>                                <chr>              <chr> <chr>
##  1  8387   232         Black or African American Not Hispanic/Latino
##  2  8437   736                                White Not Hispanic/Latino
##  3  8499   555                Two or more races Not Hispanic/Latino
##  4  8788   782                                White             Unknown
##  5  8810   243                                White Not Hispanic/Latino
##  6  8843   820                                White Not Hispanic/Latino
##  7  8849   555                                Asian Not Hispanic/Latino
##  8  9360   232 American Indian or Alaskan Native             Unknown
##  9  9380    92                              Unknown             Unknown
## 10  9410   232                                White Not Hispanic/Latino
## # ... with 722 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
## #   item_5_response <int>, item_5_correct <chr>, item_6_response <int>,
```

Other helper functions used within `summarize`

- We've discussed `n()` - number of rows
- `n_distinct()` provides number of distinct values
- `first()` provides the first value
- `last()` provides the last value
- `nth()` provides the nth row

```
mcrc %>%
  group_by(schid) %>%
  summarize(n_students = n(),
            n_race_cats = n_distinct(race),
            n_eth_cats = n_distinct(ethnicity))
```

```
## # A tibble: 1,000 x 4
##    schid n_students n_race_cats n_eth_cats
##    <int>      <int>       <int>      <int>
##  1     1         71           6          4
##  2     2         73           7          4
##  3     3         64           6          4
##  4     4         62           7          4
##  5     5         75           7          4
##  6     6         79           7          4
##  7     7         74           6          3
##  8     8         87           8          4
##  9     9         72           7          4
## 10    10         93           7          4
## # ... with 990 more rows
```

Using the economics dataset from the package, identify the population from the first, middle, and last observations of each year

```
ggplot2::economics %>%
  arrange(date) %>%
  tidyr::separate(date, c("year", "month", "day"), sep = "-") %>%
  group_by(year) %>%
  summarize(n_obs = n(),
            first_ob = first(pop),
            middle_ob = nth(pop, round(n_obs / 2)),
            last_ob = last(pop))
```

```
## # A tibble: 49 x 5
##     year n_obs first_ob middle_ob last_ob
##    <chr> <int>    <int>     <int>   <int>
##  1  1967     6   198712    199113  199657
##  2  1968    12   199808    200536  201621
##  3  1969    12   201760    202507  203675
##  4  1970    12   203849    204830  206238
##  5  1971    12   206466    207462  208740
##  6  1972    12   208917    209725  210821
##  7  1973    12   210985    211746  212785
##  8  1974    12   212932    213686  214782
##  9  1975    12   214931    215768  216931
```

We've seen both of these, but to revisit

`recode()` and `rename()`

(note, with recode I had troubles trying to use `"" = "N"`)

```
mcrc %>%
  rename(student_id = id) %>%
  mutate(ethnicity = recode(ethnicity, "Unknown" = "NA"),
         ell = recode(ell, "Y" = "Yes", .default = "No"))
```

```
## # A tibble: 72,948 x 47
##    student_id schid                    race           ethnicity   ell
##         <int> <int>                   <chr>               <chr> <chr>
##  1       8307   381                   White Not Hispanic/Latino    No
##  2       8308   449                         Not Hispanic/Latino    No
##  3       8309   817                   White Not Hispanic/Latino    No
##  4       8310   979 Black or African American Not Hispanic/Latino    No
##  5       8311    13                   White Not Hispanic/Latino    No
##  6       8312   410 Black or African American     Hispanic/Latino   Yes
##  7       8313   529                   White Not Hispanic/Latino    No
##  8       8314   967 Black or African American Not Hispanic/Latino    No
##  9       8315   589 Black or African American Not Hispanic/Latino    No
## 10       8316   257                 Unknown                  NA    No
```

Conditional verbs

`mutate` If there are systematic manipulations, or many manipulation you need to make, use the conditional versions.

```
mcrc %>%
  mutate_if(is.character, as.factor)
```

```
## # A tibble: 72,948 x 47
##       id schid                        race           ethnicity    ell
##    <int> <int>                      <fctr>              <fctr> <fctr>
##  1  8307   381                       White Not Hispanic/Latino
##  2  8308   449                             Not Hispanic/Latino
##  3  8309   817                       White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                       White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino      Y
##  7  8313   529                       White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                     Unknown             Unknown
## # ... with 72,938 more rows, and 42 more variables: disability <fctr>,
## #   score <int>, item_1_response <int>, item_1_correct <fctr>,
```

```
mcrc %>%
  mutate_if(is.numeric, scale)
```

```
## # A tibble: 72,948 x 47
##            id      schid                      race           ethnicity
##         <dbl>      <dbl>                     <chr>               <chr>
##  1 -0.9431383 -0.4078646                     White Not Hispanic/Latino
##  2 -0.9430433 -0.1730446                           Not Hispanic/Latino
##  3 -0.9429483  1.0977462                     White Not Hispanic/Latino
##  4 -0.9428533  1.6571704 Black or African American Not Hispanic/Latino
##  5 -0.9427584 -1.6786554                     White Not Hispanic/Latino
##  6 -0.9426634 -0.3077208 Black or African American     Hispanic/Latino
##  7 -0.9425684  0.1032143                     White Not Hispanic/Latino
##  8 -0.9424735  1.6157316 Black or African American Not Hispanic/Latino
##  9 -0.9423785  0.3104084 Black or African American Not Hispanic/Latino
## 10 -0.9422835 -0.8360659                   Unknown             Unknown
## # ... with 72,938 more rows, and 43 more variables: ell <chr>,
## #   disability <chr>, score <dbl>, item_1_response <dbl>,
## #   item_1_correct <chr>, item_2_response <dbl>, item_2_correct <chr>,
## #   item_3_response <dbl>, item_3_correct <chr>, item_4_response <dbl>,
## #   item_4_correct <chr>, item_5_response <dbl>, item_5_correct <chr>,
## #   item_6_response <dbl>, item_6_correct <chr>, item_7_response <dbl>,
## #   item_7_correct <chr>, item_8_response <dbl>, item_8_correct <chr>,
```

```
mcrc %>%
  mutate_all(as.character)
```

```
## # A tibble: 72,948 x 47
##       id schid                          race           ethnicity   ell
##    <chr> <chr>                         <chr>               <chr> <chr>
##  1  8307   381                         White Not Hispanic/Latino
##  2  8308   449                               Not Hispanic/Latino
##  3  8309   817                         White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                         White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino     Y
##  7  8313   529                         White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                      Unknown             Unknown
## # ... with 72,938 more rows, and 42 more variables: disability <chr>,
## #   score <chr>, item_1_response <chr>, item_1_correct <chr>,
## #   item_2_response <chr>, item_2_correct <chr>, item_3_response <chr>,
## #   item_3_correct <chr>, item_4_response <chr>, item_4_correct <chr>,
## #   item_5_response <chr>, item_5_correct <chr>, item_6_response <chr>,
## #   item_6_correct <chr>, item_7_response <chr>, item_7_correct <chr>,
## #   item_8_response <chr>, item_8_correct <chr>, item_9_response <chr>,
```

```
mcrc %>%
  mutate_at(vars(race, ethnicity, ell), as.factor)
```

```
## # A tibble: 72,948 x 47
##       id schid                        race           ethnicity    ell
##    <int> <int>                      <fctr>              <fctr> <fctr>
##  1  8307   381                       White Not Hispanic/Latino
##  2  8308   449                             Not Hispanic/Latino
##  3  8309   817                       White Not Hispanic/Latino
##  4  8310   979 Black or African American Not Hispanic/Latino
##  5  8311    13                       White Not Hispanic/Latino
##  6  8312   410 Black or African American     Hispanic/Latino      Y
##  7  8313   529                       White Not Hispanic/Latino
##  8  8314   967 Black or African American Not Hispanic/Latino
##  9  8315   589 Black or African American Not Hispanic/Latino
## 10  8316   257                     Unknown             Unknown
## # ... with 72,938 more rows, and 42 more variables: disability <chr>,
## #   score <int>, item_1_response <int>, item_1_correct <chr>,
## #   item_2_response <int>, item_2_correct <chr>, item_3_response <int>,
## #   item_3_correct <chr>, item_4_response <int>, item_4_correct <chr>,
## #   item_5_response <int>, item_5_correct <chr>, item_6_response <int>,
## #   item_6_correct <chr>, item_7_response <int>, item_7_correct <chr>,
## #   item_8_response <int>, item_8_correct <chr>, item_9_response <int>,
```

group_by_if

```
mcrc %>%
  select(-starts_with("item")) %>%
  group_by_if(is.character) %>%
  summarize(mean = mean(score, na.rm = TRUE))
```

```
## # A tibble: 116 x 5
## # Groups:   race, ethnicity, ell [?]
##     race           ethnicity   ell disability     mean
##    <chr>               <chr> <chr>      <chr>    <dbl>
##  1                                          13.28889
##  2                                        Y 12.83333
##  3                           Y              10.33333
##  4                           Y            Y  8.00000
##  5          Hispanic/Latino                 13.31176
##  6          Hispanic/Latino               Y 11.04762
##  7          Hispanic/Latino     Y           11.61364
##  8          Hispanic/Latino     Y         Y 10.36364
##  9       Not Hispanic/Latino                14.92762
## 10       Not Hispanic/Latino              Y 11.62121
## # ... with 106 more rows
```

summarize_versions

```
mcrc %>%
  summarize_if(is.numeric, funs(mean, sd))
```

```
## # A tibble: 1 x 46
##   id_mean schid_mean score_mean item_1_response_mean item_2_response_mean
##     <dbl>      <dbl>      <dbl>                <dbl>                <dbl>
## 1 18237.5   499.1108   13.35857                   NA                   NA
## # ... with 41 more variables: item_3_response_mean <dbl>,
## #   item_4_response_mean <dbl>, item_5_response_mean <dbl>,
## #   item_6_response_mean <dbl>, item_7_response_mean <dbl>,
## #   item_8_response_mean <dbl>, item_9_response_mean <dbl>,
## #   item_10_response_mean <dbl>, item_11_response_mean <dbl>,
## #   item_12_response_mean <dbl>, item_13_response_mean <dbl>,
## #   item_14_response_mean <dbl>, item_15_response_mean <dbl>,
## #   item_16_response_mean <dbl>, item_17_response_mean <dbl>,
## #   item_18_response_mean <dbl>, item_19_response_mean <dbl>,
## #   item_20_response_mean <dbl>, id_sd <dbl>, schid_sd <dbl>,
## #   score_sd <dbl>, item_1_response_sd <dbl>, item_2_response_sd <dbl>,
## #   item_3_response_sd <dbl>, item_4_response_sd <dbl>,
## #   item_5_response_sd <dbl>, item_6_response_sd <dbl>,
## #   item_7_response_sd <dbl>, item_8_response_sd <dbl>,
```

```
mcrc %>%
  select(contains("correct")) %>%
  mutate_all(funs(ifelse(. == "Y", 1, 0))) %>%
  summarize_all(funs(mean, sd))
```

```
## # A tibble: 1 x 40
##   item_1_correct_mean item_2_correct_mean item_3_correct_mean
##                 <dbl>               <dbl>               <dbl>
## 1           0.8860284           0.8216675           0.8473707
## # ... with 37 more variables: item_4_correct_mean <dbl>,
## #   item_5_correct_mean <dbl>, item_6_correct_mean <dbl>,
## #   item_7_correct_mean <dbl>, item_8_correct_mean <dbl>,
## #   item_9_correct_mean <dbl>, item_10_correct_mean <dbl>,
## #   item_11_correct_mean <dbl>, item_12_correct_mean <dbl>,
## #   item_13_correct_mean <dbl>, item_14_correct_mean <dbl>,
## #   item_15_correct_mean <dbl>, item_16_correct_mean <dbl>,
## #   item_17_correct_mean <dbl>, item_18_correct_mean <dbl>,
## #   item_19_correct_mean <dbl>, item_20_correct_mean <dbl>,
## #   item_1_correct_sd <dbl>, item_2_correct_sd <dbl>,
## #   item_3_correct_sd <dbl>, item_4_correct_sd <dbl>,
## #   item_5_correct_sd <dbl>, item_6_correct_sd <dbl>,
## #   item_7_correct_sd <dbl>, item_8_correct_sd <dbl>,
## #   item_9_correct_sd <dbl>, item_10_correct_sd <dbl>,
```

Challege (more from last class)

- Make the output more readable (tidy)
- Produce a plot

What I asked for

```
library(tidyverse)
mcrc %>%
  select(contains("correct")) %>%
  mutate_all(funs(ifelse(. == "Y", 1, 0))) %>%
  summarize_all(funs(mean, sd)) %>%
  gather(var, val) %>%
  separate(var, c("dis1", "item", "dis2", "stat"), sep = "_", convert = TRUE) %>%
  select(-contains("dis")) %>%
  spread(stat, val)
```

```
## # A tibble: 20 x 3
##     item      mean        sd
##  * <int>     <dbl>     <dbl>
##  1     1 0.8860284 0.3177789
##  2     2 0.8216675 0.3827950
##  3     3 0.8473707 0.3596322
##  4     4 0.3674261 0.4821072
##  5     5 0.8232714 0.3814415
##  6     6 0.7587871 0.4278221
##  7     7 0.6976202 0.4592920
##  8     8 0.8444234 0.3624559
##  9     9 0.8109064 0.3915856
```

Better approach - tidy first

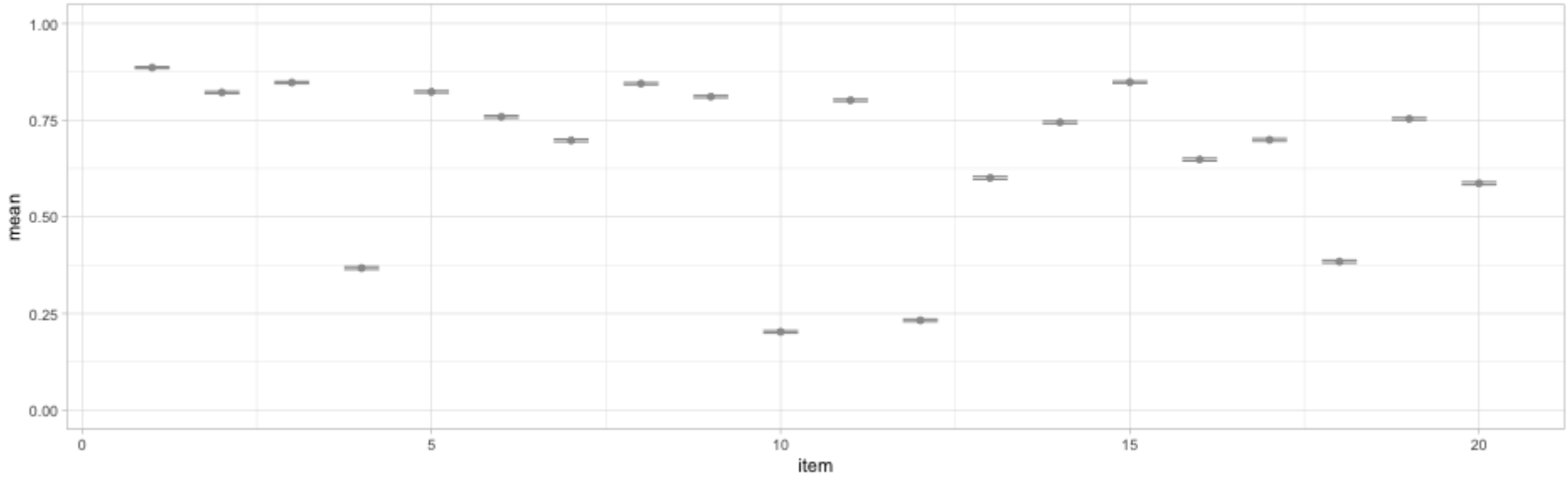
```
itm_summary <- mcrc %>%
  select(contains("correct")) %>%
  gather(item, response) %>%
  mutate(item = parse_number(item),
         response = ifelse(response == "Y", 1, 0)) %>%
  group_by(item) %>%
  summarize(mean = mean(response),
            se = sundry::se(response))
itm_summary
```

```
## # A tibble: 20 x 3
##     item      mean          se
##    <dbl>     <dbl>       <dbl>
##  1     1 0.8860284 0.001176572
##  2     2 0.8216675 0.001417293
##  3     3 0.8473707 0.001331533
##  4     4 0.3674261 0.001784995
##  5     5 0.8232714 0.001412281
##  6     6 0.7587871 0.001584005
##  7     7 0.6976202 0.001700522
##  8     8 0.8444234 0.001341988
##  9     9 0.8109064 0.001449840
```

plot

One potential

```
theme_set(theme_light())
ggplot(itm_summary, aes(x = item, y = mean)) +
  geom_point(color = "gray60") +
  ylim(0, 1) +
  geom_errorbar(aes(ymin = mean - 2*se, ymax = mean + 2*se), width = 0.5, color = "gray60")
```



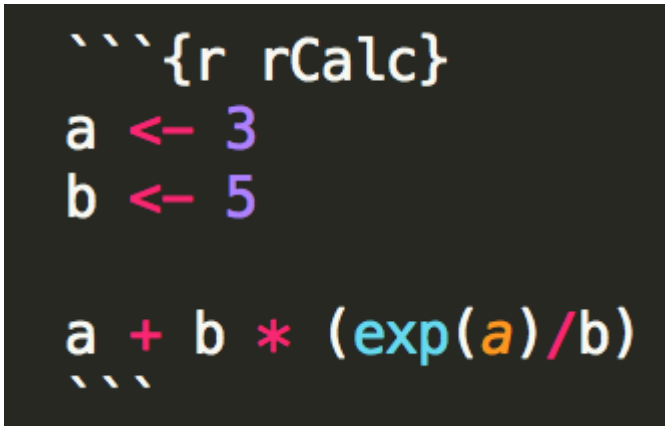
RMarkdown

Things I want to cover

- Chunk options
- Setting global options
- Inline code evaluation
- Citations
- Caching (briefly)
- Brief look at the package
- Practice (if time allows)

Code chunks

Start a code chunk with ```{r chunkName, chunkOptions}, then produce some r code, then close the chunk with three additional back ticks ```.




```
a <- 3
b <- 5

a + b * (exp(a)/b)
```

```
## [1] 23.08554
```

A few select chunk options

OPTIONS	ARGUMENTS	DEFAULT	RESULT
eval	logical	TRUE	Evaluate the code?
echo	logical	TRUE	Show the code?
results	markup, asis, hold, hide	markup	Render the results
warning	logical	TRUE	Print warnings?
error	logical	TRUE	Preserve errors? (if FALSE, quit)
message	logical	TRUE	Print any messages?
include	logical	TRUE	Include any of the code or output or code?
tidy	logical	FALSE	Tidy code? (see formatR package)

(and a few more)

	OPTIONS	ARGUMENTS	DEFAULT	RESULT
9	cache	logical, 0:3	FALSE	Cache code chunks?
10	cache.comments	logical	NULL	Cache invalidated by comment changes?
11	dependson	char, num	NULL	Current chunk depend on prior cached chunks?
12	autodep	logical	FALSE	Should dependencies be determined automatically? (if TRUE, no need for dependson)
13	fig.height/fig.width	numeric	7, 7	Height and width of figure
14	fig.show	asis, hold, animate, hide	asis	How the figure should be displayed
15	interval	numeric	1	Interval (speed) When fig.show = 'animate'

For complete documentation, see http://yihui.name/knitr/options/

echo and eval

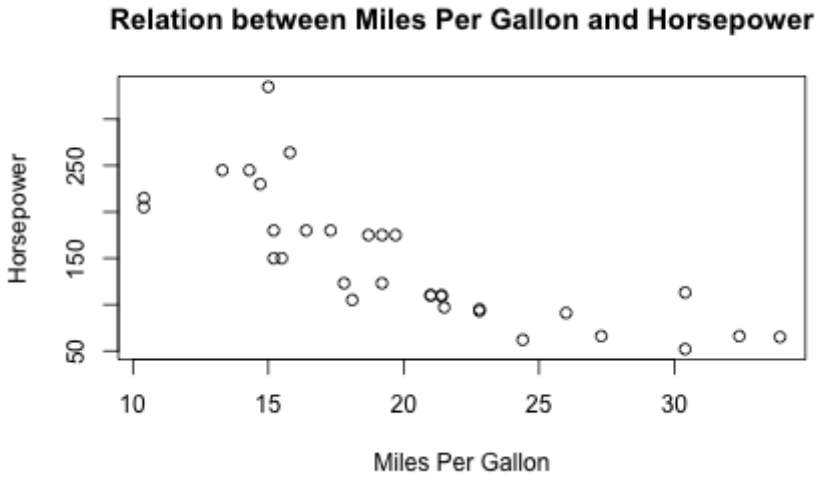
You can show code without evaluating it, using `eval = FALSE`.

````
```{r ex_rCalc2, eval = FALSE}
a + b * (exp(a)/b)
```
````

```
a + b * (exp(a)/b)
```

Alternatively, you can evaluate the code without displaying it, using `echo = FALSE`.

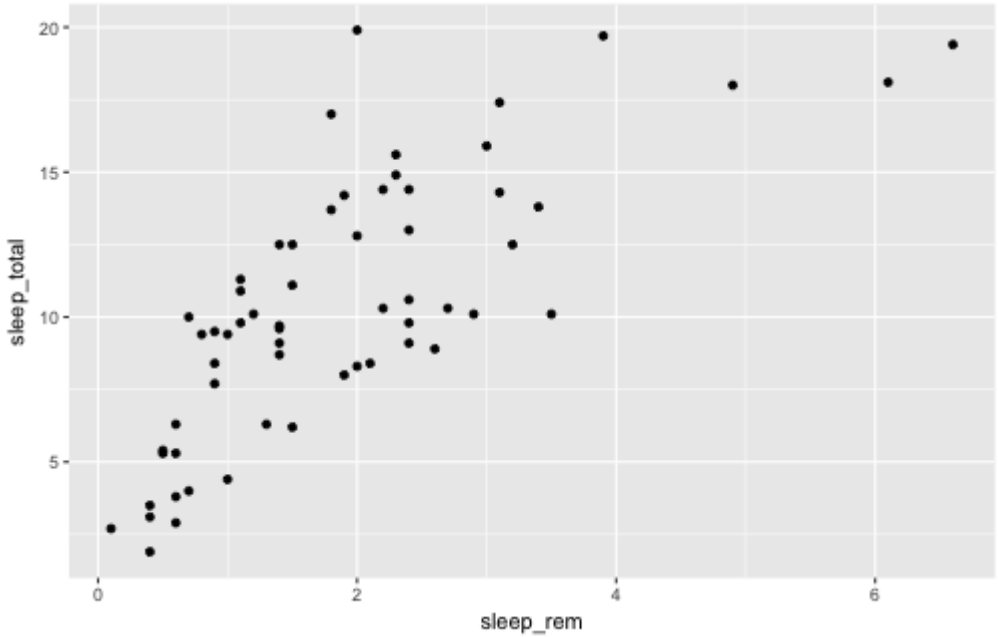
````
```{r plotExample, echo = FALSE, fig.width = 6, fig.height = 3.8}
data(mtcars)
with(mtcars, plot(mpg, hp,
  xlab = "Miles Per Gallon",
  ylab = "Horsepower",
  main = "Relation between Miles Per Gallon and Horsepower"))
```
````



warning

Warning = FALSE

```
ggplot(msleep,
  aes(sleep_rem, sleep_total)) +
  geom_point()
```

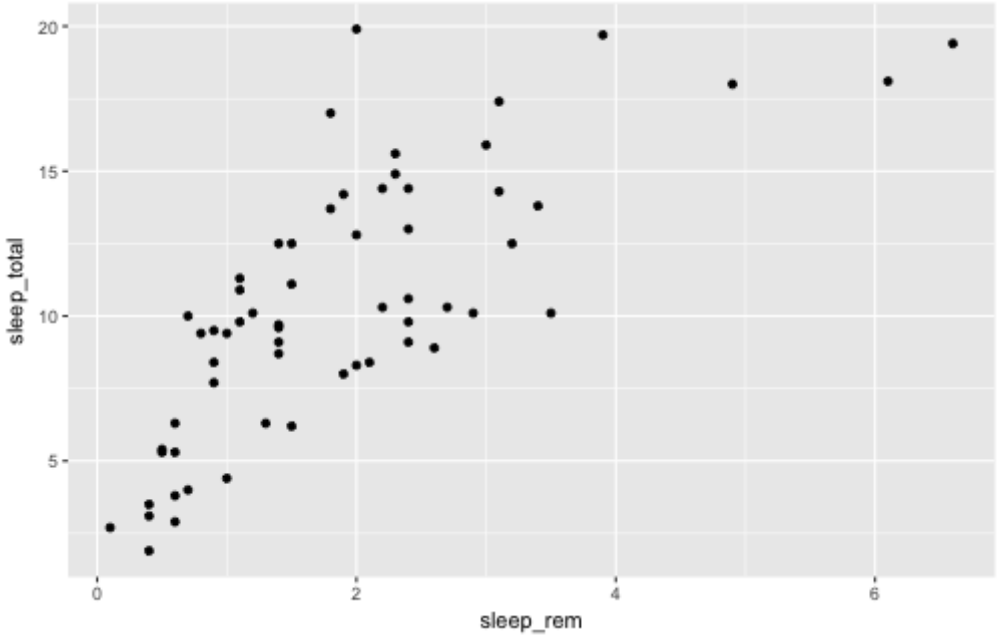


Warning is printed to the console when rendering.

Warning = TRUE

```
ggplot(msleep,
  aes(sleep_rem, sleep_total)) +
  geom_point()
```

```
## Warning: Removed 22 rows containing missing
```



Show errors

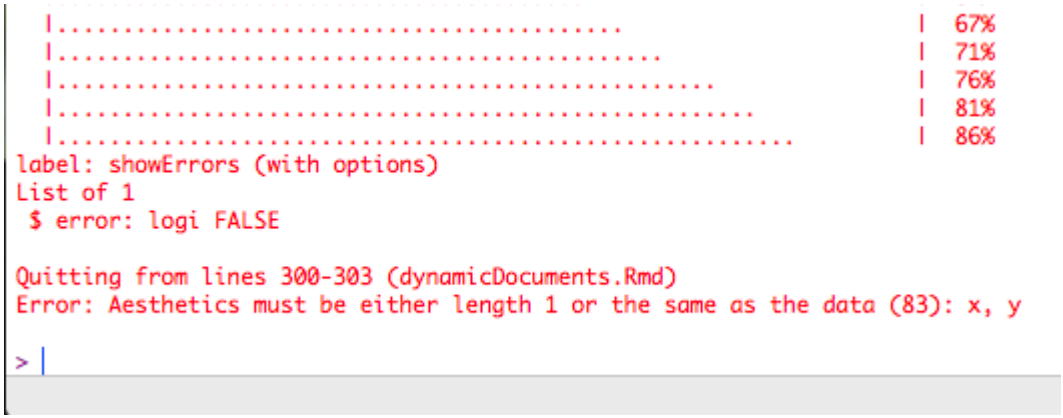
Default

```
ggplot(msleep,
  aes(sleep, sleep_total)) +
  geom_point()
```

```
## Don't know how to automatically pick scale for object of type data.frame. Defaulting to cor
```

```
## Error: Aesthetics must be either length 1 or the same as the data (83): x, y
```

If `error = FALSE`, the document won't render if it encounters an error.

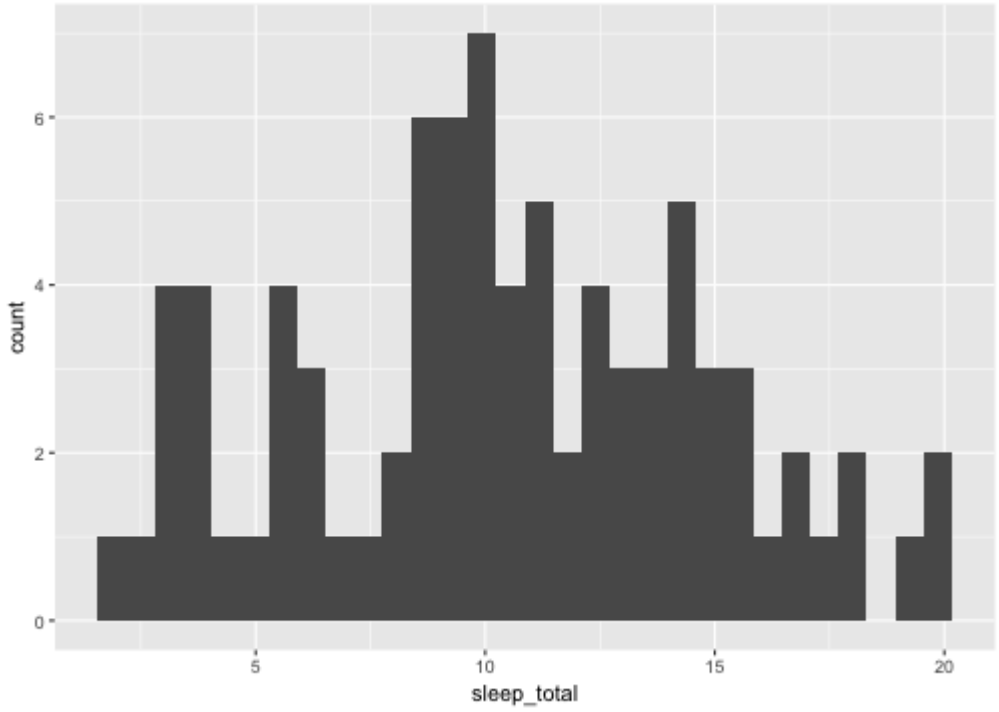


Message

Some functions will return messages. You may want to suppress these.

message = FALSE

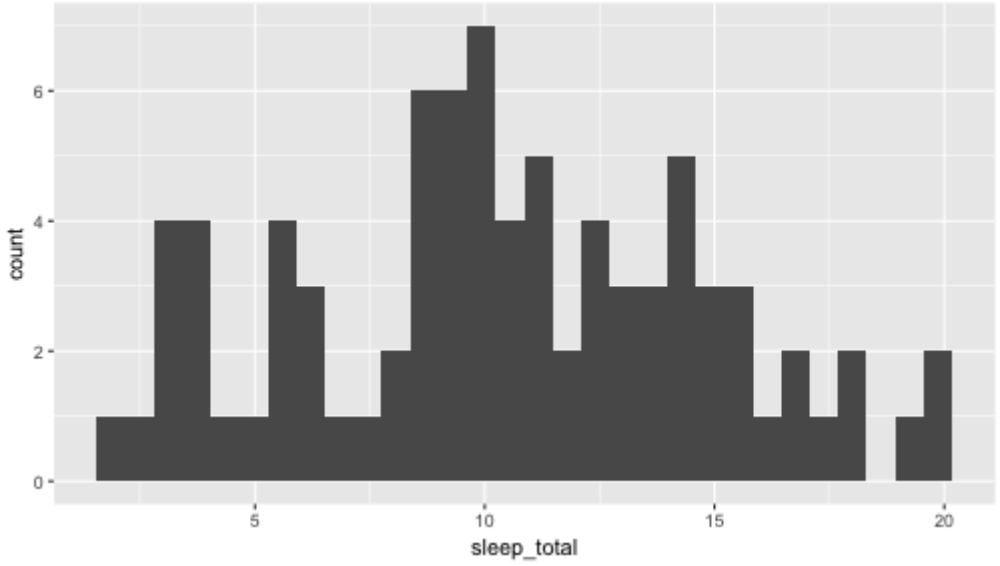
```
ggplot(msleep,
  aes(sleep_total)) +
  geom_histogram()
```



message = TRUE

```
ggplot(msleep,
  aes(sleep_total)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better
```



tidy

Tidy = FALSE

```
matRow<-matrix(c(10,11,12,13,20,21,22,23,
  30,31,32,33),nrow=3,ncol=4,byrow=TRUE)

matRow<-matrix(c(10,11,12,13,
                 20,21,22,23,
                 30,31,32,33),
nrow=3,ncol=4,byrow=TRUE)
```

Tidy = TRUE

```
matRow <- matrix(c(10, 11, 12, 13, 20, 21, 22, 
    ncol = 4, byrow = TRUE)

matRow <- matrix(c(10, 11, 12, 13, 20, 21, 22, 
    ncol = 4, byrow = TRUE)
```

(It can only do so much, and sometimes ends up looking worse. Follow a style! Don't rely on this.)

include

````
```{r setup, include = FALSE}
library(knitr)

# Set global chunk options
opts_chunk$set(cache = TRUE, cache.comments = FALSE, autodep = T

dep_auto()
```
````

The `include` argument is used to evaluate code that is not included in the document at all. For example, when setting up your global options.

Other options for results: `hold`

```
m1 <- lm(mpg ~ ., data = mtcars)
coef(m1)
coef(summary(m1))[, "Std. Error"]
arm::display(m1)
```

```
## (Intercept)         cyl        disp          hp        drat          wt
## 12.30337416 -0.11144048  0.01333524 -0.02148212  0.78711097 -3.71530393
##        qsec          vs          am        gear        carb
##  0.82104075  0.31776281  2.52022689  0.65541302 -0.19941925
## (Intercept)         cyl        disp          hp        drat          wt
## 18.71788443  1.04502336  0.01785750  0.02176858  1.63537307  1.89441430
##        qsec          vs          am        gear        carb
##  0.73084480  2.10450861  2.05665055  1.49325996  0.82875250
## lm(formula = mpg ~ ., data = mtcars)
##             coef.est coef.se
## (Intercept) 12.30    18.72
## cyl         -0.11     1.05
## disp         0.01     0.02
## hp          -0.02     0.02
## drat         0.79     1.64
## wt          -3.72     1.89
```

Same chunk, no hold

```
m1 <- lm(mpg ~ ., data = mtcars)
coef(m1)
```

```
## (Intercept)         cyl        disp          hp        drat          wt
## 12.30337416 -0.11144048  0.01333524 -0.02148212  0.78711097 -3.71530393
##        qsec          vs          am        gear        carb
##  0.82104075  0.31776281  2.52022689  0.65541302 -0.19941925
```

```
coef(summary(m1))[, "Std. Error"]
```

```
## (Intercept)         cyl        disp          hp        drat          wt
## 18.71788443  1.04502336  0.01785750  0.02176858  1.63537307  1.89441430
##        qsec          vs          am        gear        carb
##  0.73084480  2.10450861  2.05665055  1.49325996  0.82875250
```

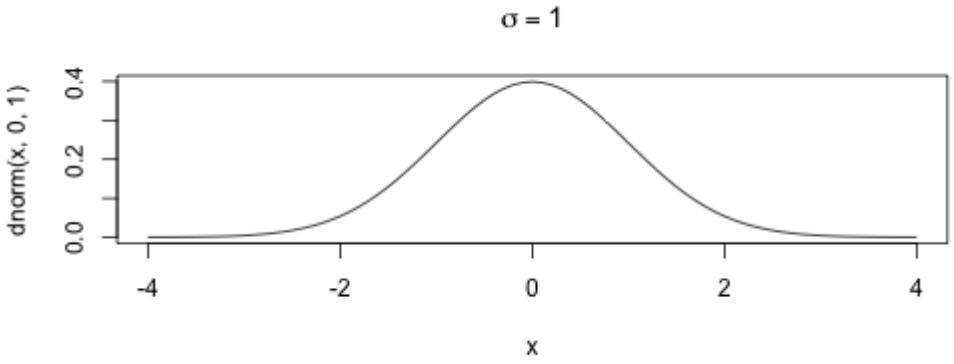
```
arm::display(m1)
```

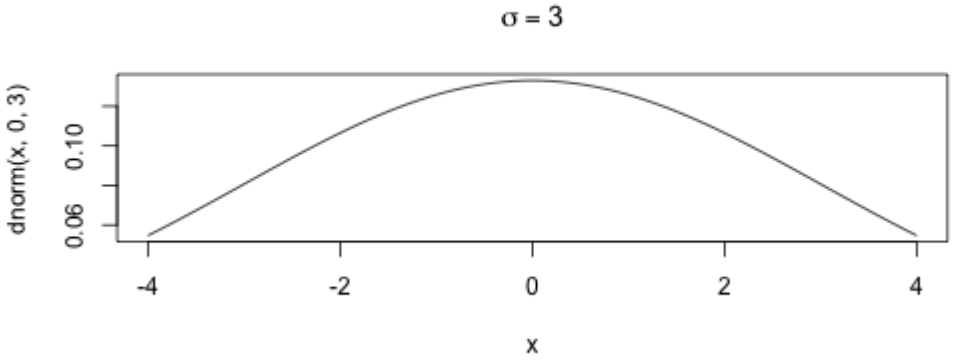
```
## lm(formula = mpg ~ ., data = mtcars)
##             coef.est coef.se
```

Hold figures

`fig.show = "hold"`

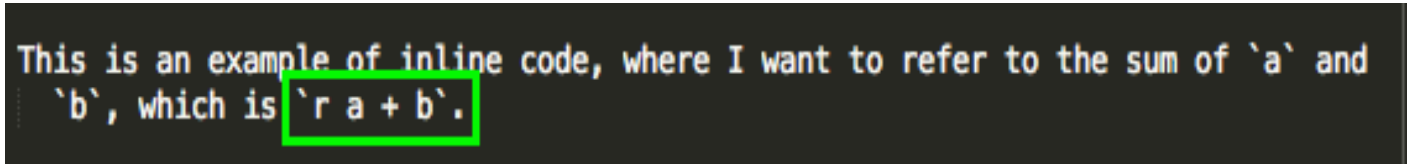
```
x <- seq(-4, 4, 0.1)
plot(x, dnorm(x, 0, 1), type = "l", main = expression(sigma == 1))
plot(x, dnorm(x, 0, 3), type = "l", main = expression(sigma == 3))
```





Inline code

A single back tick followed by `r` prooduces inline code to be evaluated.



This is an example of inline code, where I want to refer to the sum of `a` and `b`, which is 8.

This is useful in writing reports. Never have to update any numbers in text, regardless of changes to your models or data (if you are careful about it).

Real example

```
```{r ell_grp_means}
ell_means <- with(d, tapply(reading, ell, mean, na.rm = TRUE))
```

Figure 2 represents a example of why the PP plot is such a powerful visualization of group differences. As previously stated, the *Monitor* group had the highest average achievement, scoring `r round(ell_means[2] - ell_means[3], 2)` points higher than *Non-ELL* students and `r round(ell_means[2] - ell_means[1], 2)` points higher than *Active* students, on average. However, the data visualization provides a more nuanced picture of the achievement differences. The *Monitor* curve is below the reference line on the lower end of the scale, but above the reference line at the upper end of the scale. In other words, students in the *Monitor* group only scored higher than students in the *Non-ELL* group at the bottom of the scale. At the upper end of the scale the effect was reversed, and students in the *Non-ELL* group had the higher achievement. In this case, the line crosses at essentially the 50th percentile of achievement for non-ELL students. Observing these achievement differences may help lead us to more refined research questions and research hypotheses. For example, a reasonable hypothesis stemming from Figure 2 may be that low-performing students are in need of additional attention, and that providing additional attention is beneficial to their academic achievement even if the intervention is not strictly focused on academics. Interestingly, although not reported here, when the equivalent plot is produced with the mathematics outcome, the overall group differences appear very similar at the bottom of the scale, but the groups are essentially indistinguishable above the 50th percentile.
```

Setting global options

Change the default behavior

```
opts_chunk$set(options)
```

For example, you can set `echo = FALSE` and `fig.width = 6.5` and `fig.height = 8` with the following code.

```
opts_chunk$set(echo = FALSE, fig.width = 6.5, fig.height = 8)
```

This is most useful when producing a report for somebody who doesn't use R and has no use or knowledge of the code.

You can always override the defaults (global options) within a particular chunk, e.g.

```{r, chunkName, echo = TRUE}

```

Other things to consider setting globally:

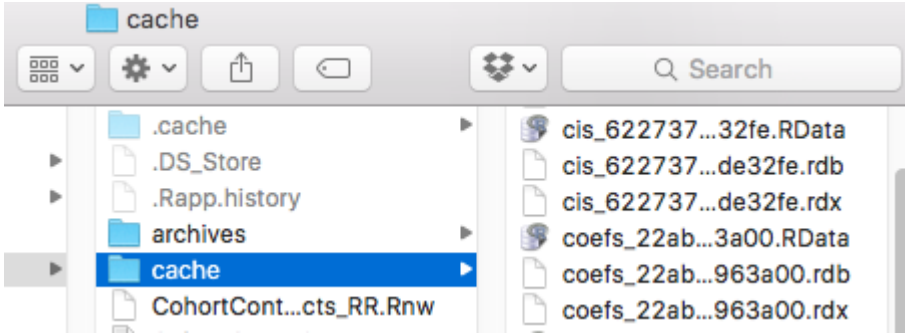
- `warnings = FALSE`
- `message = FALSE`
- `errors = TRUE`
- `echo = FALSE`
- Caching options (next slides)

Caching (briefly)

Somewhat complicated

- When chunks take a long time to process, it is usually a good idea to them.
  - Create temporary files with the results of the chunk
  - These files are then called when the document is rendered, and do not need to be re-run, provided **nothing in the chunk has changed**.
  - Often, chunks may depend on the results of previous chunks. These are .
    All dependencies must then be updated, if a chunk with dependencies is updated.
  - You can declare dependencies manually or automatically

cache



- Cache files are named according to your chunk names (why it's important to name your chunks)
- Note that some packages that use (i.e., , which was used to produce these slides), will cache for you automatically. And the slidify cache is in a hidden folder (which can be really annoying)

Declaring dependencies manually

````
```{r data}
boys <- c(25, 32, 11, 54)
girls <- c(30, 29, 22, 43)
mean(boys)
mean(girls)
```
As can be seen, boys scored `r mean(boys) - mean(girls)` points different than
girls. Below is a histogram of each.

```{r histograms, dependson = data}
par(mfrow = c(1,2))
hist(boys)
hist(girls)
```
````

Inline code
(which we'll talk
about momentarily)

```
boys <- c(25, 32, 11, 54)
girls <- c(30, 29, 22, 43)
mean(boys)
```

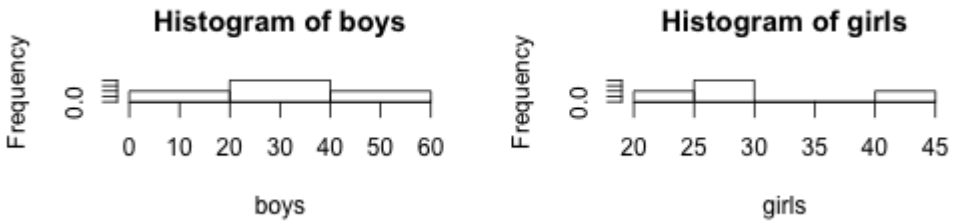
```
## [1] 30.5
```

```
mean(girls)
```

```
## [1] 31
```

As can be seen, boys scored -0.5 points different than girls. Below is a histogram of each.

```
par(mfrow = c(1,2))
hist(boys)
hist(girls)
```



Better option (generally)

Set caching to be automatic, with the dependencies automatically determined. Use global options.

```
opts_chunk$set(cache = TRUE, autodep = TRUE)

dep_auto()
```

Note that `dep_auto()` is a function that must be run on its own (which finds the dependencies).

Confused?

Don't worry, be happy

- If much of the caching part makes little to no sense to you, don't worry, it's not that big of a deal.
- Getting your document to render is the biggest deal. Caching can just help speed things up some (and a lot in some cases).

Citations

Citations

To include references in your paper, you must:

- Create an external .bib file using LaTeX formatting (we'll get to this)
- Include `bibliography: nameOfYourBibFile.bib` in your YAML front matter.
- Refer to the citations in text using `@`

Creating a .bib doc

The persistence of school-level value-added

DC Briggs, JP Weeks - Journal of Educational and Behavioral ..., 2011 - jeb.sagepub.com

Abstract Using longitudinal data for an entire state from 2004 to 2008, this article describes the results from an empirical investigation of the persistence of value-added school effects on student achievement in reading and math. It shows that when schools are the principal ...

Cited by 20 Related articles All 8 versions Web of Science: 4 Cite Save More

- Cite

Copy and paste a formatted citation or use one of the links to import into a bibliography manager.

MLA Briggs, Derek C., and Jonathan P. Weeks. "The persistence of school-lev value-added." *Journal of Educational and Behavioral Statistics* 36.5 (2011 616-637.

APA Briggs, D. C., & Weeks, J. P. (2011). The persistence of school-level valu added. *Journal of Educational and Behavioral Statistics*, *36*(5), 616-637.

Chicago Briggs, Derek C., and Jonathan P. Weeks. "The persistence of school-lev value-added." *Journal of Educational and Behavioral Statistics* 36, no. 5 (2011): 616-637.

Harvard Briggs, D.C. and Weeks, J.P., 2011. The persistence of school-level value-added. *Journal of Educational and Behavioral Statistics*, *36*(5), pp.616-637.

Vancouver Briggs DC, Weeks JP. The persistence of school-level value-added. Journal of Educational and Behavioral Statistics. 2011 Oct 1;36(5):616-37.

BibTeX EndNote RefMan RefWorks

```
@article{briggs2011persistence,
  title={The persistence of school-level value-added},
  author={Briggs, Derek C and Weeks, Jonathan P},
  journal={Journal of Educational and Behavioral Statistics},
  volume={36},
  number={5},
  pages={616--637},
  year={2011},
  publisher={SAGE Publications}
}
```

```
@article{Briggs11,
  title={The persistence of school-level value-added},
  author={Briggs, Derek C and Weeks, Jonathan P},
  journal={Journal of Educational and Behavioral Statistics},
  volume={36},
  number={5},
  pages={616--637},
  year={2011},
  publisher={SAGE Publications}
}
```

Tag for in-text referencing

In text citations

CITATION STYLE	OUTPUT
@Briggs11	Briggs and Weeks (2011)
[see @Baldwin2014; @Caruso2000]	(see Baldwin et al. 2014; Caruso 2000)
[@Linn02, p. 9]	(Linn and Haug 2002, 9)
[-@Goldhaber08]	(2008)

Note this is not APA. However, references are included automatically at the end of the document. Include `# References` as the last line of your document to give it a title.

A few real examples

```
A more flexible approach, discussed by @ho_09 and others [@ho_12;
@reardon_15], is to construct a probability-probability (PP) plot and compute
the area under the PP curve (see Figure 1). Note that the PP curve is
```

```
 normal distributions separated by standard deviation units [@ho_09]. The *V*
 statistic assumes respective normality "where two distributions that may not
 be normal may nonetheless be normal with respect to each other, under a shared
 transformation" [@ho_09, p. 217]. The assumptions underlying *V* are thus less
```

```
where $ref$ and $foc$ represent the reference and focal groups, respectively
[see @lakens_13]. The magnitude of the effect size is primarily driven by the
```

References

References

Baldwin, Scott A, Zac E Imel, Scott R Braithwaite, and David C Atkins. 2014. "Analyzing Multiple Outcomes in Clinical Research Using Multivariate Multilevel Models." *Journal of Consulting and Clinical Psychology* 82 (5). American Psychological Association: 920.

Briggs, Derek C, and Jonathan P Weeks. 2011. "The Persistence of School-Level Value-Added." *Journal of Educational and Behavioral Statistics* 36 (5). SAGE Publications: 616–37.

Caruso, John C. 2000. "Reliability Generalization of the NEO Personality Scales." *Educational and Psychological Measurement* 60 (2). Sage Publications: 236–54.

Goldhaber, D., and M. Hansen. 2008. "Is It Just a Bad Class? Assessing the Stability of Measured Teacher Performance. CPRE Working Paper No. 2008-5, University of Washington." Report.

Linn, R. L., and C. Haug. 2002. "Stability of School-Building Accountability Scores and Gains." Journal Article. *Educational Evaluation and Policy Analysis* 24: 29–36. doi:10.3102/01623737024001029.

Let's try!

[demo first]

- Open R Studio
    - Open a new Rmd document
- In your YAML, add a new line for `bibliography:`. Follow this with the name of your bibliography file, which should be in the same directory, and should be the bare name (not in quotes).
- Open a text file, save it with a `.bib` extension.
- Go to google Scholar and find a reference to cite. Copy and paste the citation into your .bib file using google's cite option.
- Include a reference to the citation in your Rmd file and render it. Include a page break so your citations are on a separate page.

Other potential option

- Add a GUI-like interface with
    - written by the same author who wrote the APA R Markdown documentation

See https://github.com/crsh/citr

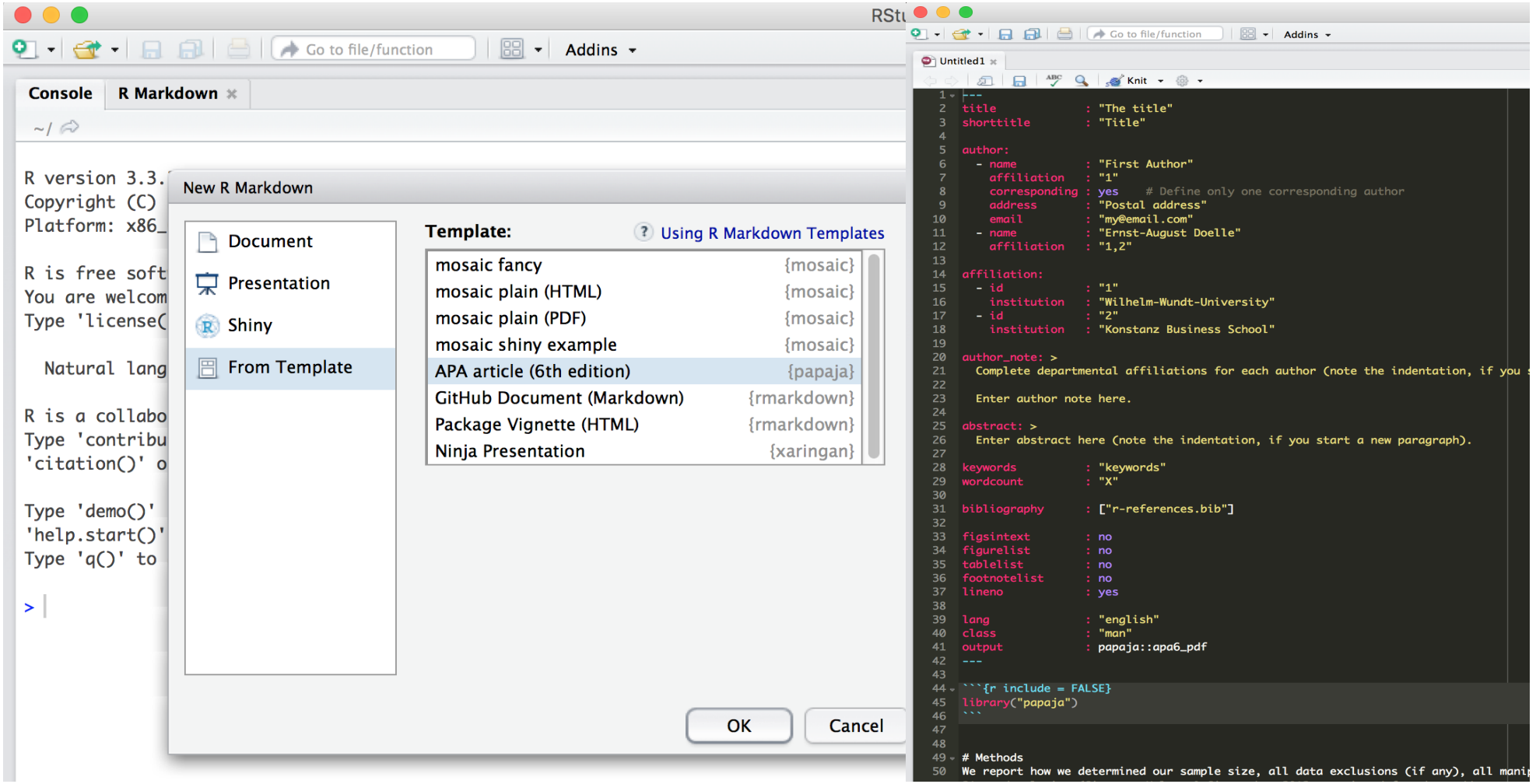
Writing a paper in apa style

Use the package

```
library(devtools)
install_github("crsh/papaja")
```

- Full Documentation available here: https://crsh.github.io/papaja_man/index.html

Template


Console
R Markdown
New R Markdown
Document
Presentation
Shiny
From Template
Template:
Using R Markdown Templates
mosaic fancy {mosaic}
mosaic plain (HTML) {mosaic}
mosaic plain (PDF) {mosaic}
mosaic shiny example {mosaic}
APA article (6th edition) {papaja}
GitHub Document (Markdown) {rmarkdown}
Package Vignette (HTML) {rmarkdown}
Ninja Presentation {xaringan}
OK
Cancel
Untitled1
Knit
Addins
Go to file/function

APA Tables

```
papaja::apa_table(head(mtcars))
```

Table 1

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.00	6.00	160.00	110.00	3.90	2.62	16.46	0.00	1.00	4.00	4.00
Mazda RX4 Wag	21.00	6.00	160.00	110.00	3.90	2.88	17.02	0.00	1.00	4.00	4.00
Datsun 710	22.80	4.00	108.00	93.00	3.85	2.32	18.61	1.00	1.00	4.00	1.00
Hornet 4 Drive	21.40	6.00	258.00	110.00	3.08	3.21	19.44	1.00	0.00	3.00	1.00
Hornet Sportabout	18.70	8.00	360.00	175.00	3.15	3.44	17.02	0.00	0.00	3.00	2.00
Valiant	18.10	6.00	225.00	105.00	2.76	3.46	20.22	1.00	0.00	3.00	1.00

Let's play!

- Install papaja
- Start a new document from a template
- Render it!
- Add a bibliography!
- Just for fun, try changing the `class` to `jou` instead of `man`

Note: I'm not sure what the `r_refs(file = "r-references.bib")` part is all about. I just delete it and use the normal method. Let me know if you figure it out!

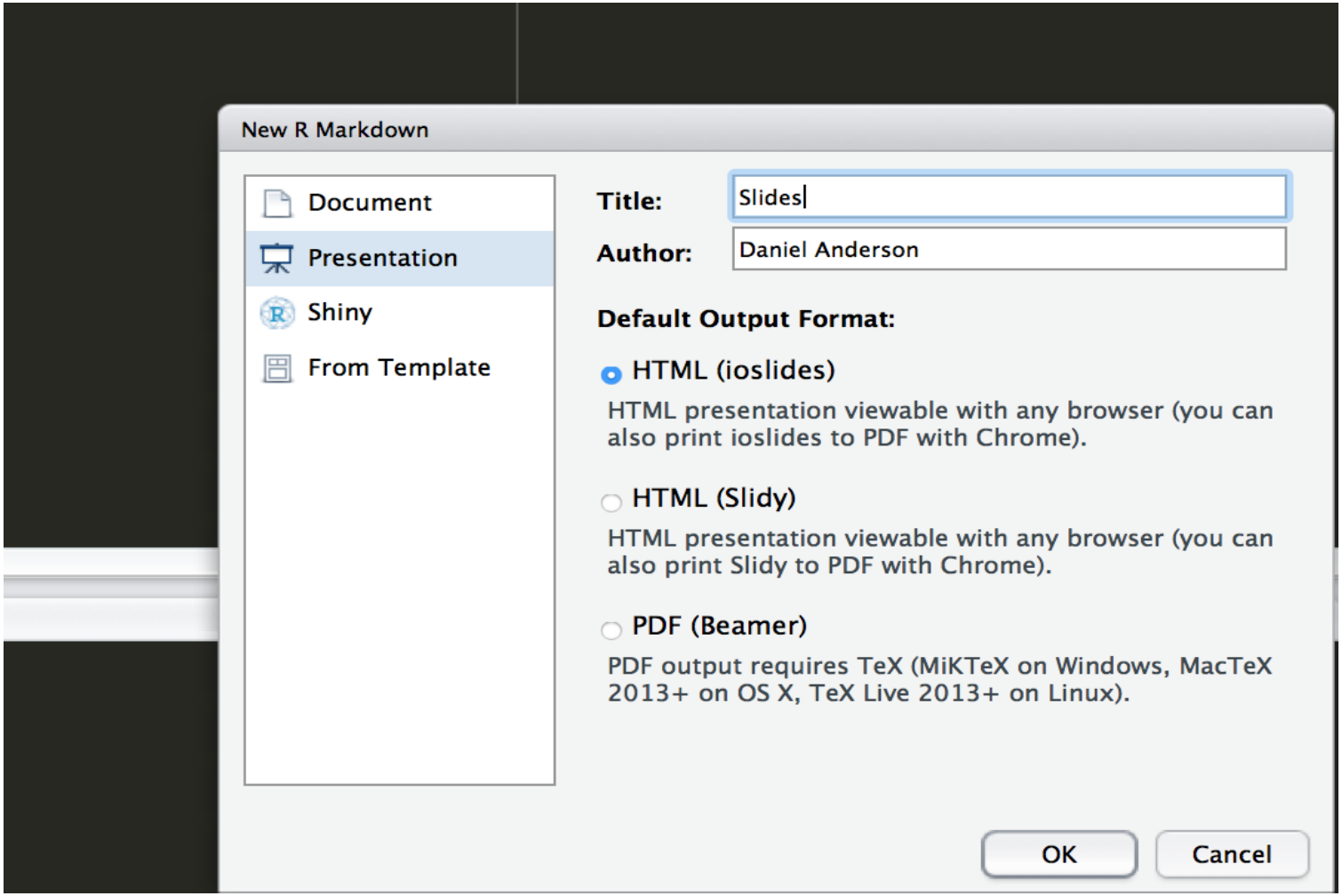
Other output options

Slides

- I've mentioned a number of times that these slides were produced with a varient of RMarkdown. The varient I use is called
    - see http://slidify.org

Probably easier way

- RStudio Presentations



Knit

```
---
title: "slides"
author: "Daniel Anderson"
date: "4/29/2017"
output: ioslides_presentation
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

R Markdown

This is an R Markdown presentation. Markdown is a simple formatting syntax
details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includ
code chunks within the document.

Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Slide with R Output

```{r cars, echo = TRUE}
summary(cars)
```

Slide with Plot

```{r pressure}
plot(pressure)
```
```

Slide with R Output

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Want to customize further?

The YAML will control a lot of how a document looks. For example, if you wanted to render with a different syntax highlighter:

Standard Rmd

```
---
title: "Doc Title"
output:
  pdf_document:
    highlight: kate
---
```

papaja

```
---
title: "Doc Title"
output:
  papaja::apa6_pdf:
    highlight: kate
---
```

Last note

- For the homework, I'll ask you to create a pdf.
- Install a TeX distribution for it to work
  - Mac download: http://tug.org/mactex/
  - Windows download: https://miktex.org
- Note the distributions are pretty big (couple gigs), but you'll need it.

Any time left?

Other formats

- blogdown: https://bookdown.org/yihui/blogdown/
- bookdown: https://bookdown.org/yihui/bookdown/

(lots of others but these are cool)

Include animations

- htmlwidgets: http://www.htmlwidgets.org

plotly

- You can use plotly along with ggplot to create interactive graphics (see https://plot.ly/ggplot2/)
- Easiest example: Use the `ggplotly` function

```
p <- ggplot(mpg, aes(manufacturer, hwy, fill = class)) + geom_bar(stat = "identity")

library(plotly)
ggplotly(p)
```

Floating TOC

Example from some consulting I did recently

```
---
title: "Exploring Treatment Effect Variance"
author: "Daniel Anderson"
date: "February 24, 2017"
output:
  html_document:
    toc: true
    toc_float: true
    code_folding: "hide"
---
```