Lab 9

incent Miceli

11:59PM April 14, 2019

"data wrangling / munging / carpentry" with dplyr.

First load `dplyr`, `tidyr`, `magrittr` and `lubridate` in one line.

```
pacman::p_load(dplyr, tidyr, magrittr, lubridate)
```

Load the `storms` dataset from the `dplyr` package and investigate it using `str` and `summary` and `head`. Which two columns should be converted to type factor? Do so below using the `mutate` and the overwrite pipe operator `%<>%`. Verify.

```
data("storms")
str(storms)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    10010 obs. of  13 variables:
##  $ name       : chr  "Amy" "Amy" "Amy" "Amy" ...
##  $ year       : num  1975 1975 1975 1975 1975 ...
##  $ month      : num  6 6 6 6 6 6 6 6 6 6 ...
##  $ day        : int  27 27 27 27 28 28 28 28 29 29 ...
##  $ hour       : num  0 6 12 18 0 6 12 18 0 6 ...
##  $ lat        : num  27.5 28.5 29.5 30.5 31.5 32.4 33.3 34 34.4 34 ...
##  $ long       : num  -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
##  $ status     : chr  "tropical depression" "tropical depression" "tropical depression" "tropical dep
##  $ category   : Ord.factor w/ 7 levels "-1"<"0"<"1"<"2"<..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ wind       : int  25 25 25 25 25 25 25 30 35 40 ...
##  $ pressure   : int  1013 1013 1013 1013 1012 1012 1011 1006 1004 1002 ...
##  $ ts_diameter: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ hu_diameter: num  NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(storms)
```

```
##      name                year          month             day        
##  Length:10010       Min.   :1975   Min.   : 1.000   Min.   : 1.00  
##  Class :character   1st Qu.:1990   1st Qu.: 8.000   1st Qu.: 8.00  
##  Mode  :character   Median :1999   Median : 9.000   Median :16.00  
##                     Mean   :1998   Mean   : 8.779   Mean   :15.86  
##                     3rd Qu.:2006   3rd Qu.: 9.000   3rd Qu.:24.00  
##                     Max.   :2015   Max.   :12.000   Max.   :31.00  
##                                                                    
##       hour             lat             long             status         
##  Min.   : 0.000   Min.   : 7.20   Min.   :-109.30   Length:10010      
##  1st Qu.: 6.000   1st Qu.:17.50   1st Qu.: -80.70   Class :character  
##  Median :12.000   Median :24.40   Median : -64.50   Mode  :character  
##  Mean   : 9.114   Mean   :24.76   Mean   : -64.23                     
##  3rd Qu.:18.000   3rd Qu.:31.30   3rd Qu.: -48.60
```

```
##  Max.   :23.000   Max.   :51.90   Max.   :  -6.00
##
##  category       wind           pressure       ts_diameter
##  -1:2545   Min.   : 10.00   Min.   : 882.0   Min.   :   0.00
##  0 :4373   1st Qu.: 30.00   1st Qu.: 985.0   1st Qu.:  69.05
##  1 :1685   Median : 45.00   Median : 999.0   Median : 138.09
##  2 : 628   Mean   : 53.49   Mean   : 992.1   Mean   : 166.76
##  3 : 363   3rd Qu.: 65.00   3rd Qu.:1006.0   3rd Qu.: 241.66
##  4 : 348   Max.   :160.00   Max.   :1022.0   Max.   :1001.18
##  5 :  68                                     NA's   :6528
##   hu_diameter
##  Min.   :  0.00
##  1st Qu.:  0.00
##  Median :  0.00
##  Mean   : 21.41
##  3rd Qu.: 28.77
##  Max.   :345.23
##  NA's   :6528
```

```
head(storms)
```

```
## # A tibble: 6 x 13
##   name   year month   day  hour   lat  long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropi~ -1          25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropi~ -1          25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropi~ -1          25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropi~ -1          25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi~ -1          25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropi~ -1          25     1012
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

```
storms %<>%
  mutate(name = factor(name), status = factor(status))
str(storms)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    10010 obs. of  13 variables:
##  $ name       : Factor w/ 198 levels "AL011993","AL012000",..: 44 44 44 44 44 44 44 44 44 44 ...
##  $ year       : num  1975 1975 1975 1975 1975 ...
##  $ month      : num  6 6 6 6 6 6 6 6 6 6 ...
##  $ day        : int  27 27 27 27 28 28 28 28 29 29 ...
##  $ hour       : num  0 6 12 18 0 6 12 18 0 6 ...
##  $ lat        : num  27.5 28.5 29.5 30.5 31.5 32.4 33.3 34 34.4 34 ...
##  $ long       : num  -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
##  $ status     : Factor w/ 3 levels "hurricane","tropical depression",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ category   : Ord.factor w/ 7 levels "-1"<"0"<"1"<"2"<..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ wind       : int  25 25 25 25 25 25 25 30 35 40 ...
##  $ pressure   : int  1013 1013 1013 1013 1012 1012 1011 1006 1004 1002 ...
##  $ ts_diameter: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ hu_diameter: num  NA NA NA NA NA NA NA NA NA NA ...
```

```
head(storms)
```

```
## # A tibble: 6 x 13
##   name   year month   day  hour   lat  long status category  wind pressure
##   <fct> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <fct>  <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropi~ -1          25     1013
## 2 Amy    1975     6    27     6  28.5 -79   tropi~ -1          25     1013
## 3 Amy    1975     6    27    12  29.5 -79   tropi~ -1          25     1013
## 4 Amy    1975     6    27    18  30.5 -79   tropi~ -1          25     1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi~ -1          25     1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropi~ -1          25     1012
## # ... with 2 more variables: ts_diameter <dbl>, hu_diameter <dbl>
```

Reorder the columns so name is first, status is second, category is third and the rest are the same. Verify.

```
storms %<>%
  select(name, status, category, everything())
storms
```

```
## # A tibble: 10,010 x 13
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Amy   tropi~ -1        1975     6    27     0  27.5 -79      25     1013
##  2 Amy   tropi~ -1        1975     6    27     6  28.5 -79      25     1013
##  3 Amy   tropi~ -1        1975     6    27    12  29.5 -79      25     1013
##  4 Amy   tropi~ -1        1975     6    27    18  30.5 -79      25     1013
##  5 Amy   tropi~ -1        1975     6    28     0  31.5 -78.8    25     1012
##  6 Amy   tropi~ -1        1975     6    28     6  32.4 -78.7    25     1012
##  7 Amy   tropi~ -1        1975     6    28    12  33.3 -78      25     1011
##  8 Amy   tropi~ -1        1975     6    28    18  34   -77      30     1006
##  9 Amy   tropi~ 0         1975     6    29     0  34.4 -75.8    35     1004
## 10 Amy   tropi~ 0         1975     6    29     6  34   -74.8    40     1002
## # ... with 10,000 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Sort the dataframe by year (most recent first) then category of the storm (most sever first). Verify.

```
storms %<>%
  arrange(desc(year), desc(category))
storms
```

```
## # A tibble: 10,010 x 13
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12  23.1 -73.7   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18  23   -74.2   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0  22.9 -74.4   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6  23   -74.7   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12  23.4 -74.8   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0  24.3 -74.3   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6  24.8 -73.6   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12  25.4 -72.6   135      934
```

```
##  9 Joaq~ hurri~ 4         2015    10     3    18  26.3 -71     130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0  27.4 -69.5   115      941
## # ... with 10,000 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Create a new feature `wind_speed_per_unit_pressure`.

```
storms %<>%
  mutate(wind_speed_per_unit_pressure = wind / pressure)
storms
```

```
## # A tibble: 10,010 x 14
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12  23.1 -73.7   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18  23   -74.2   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0  22.9 -74.4   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6  23   -74.7   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12  23.4 -74.8   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0  24.3 -74.3   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6  24.8 -73.6   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12  25.4 -72.6   135      934
##  9 Joaq~ hurri~ 4         2015    10     3    18  26.3 -71     130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0  27.4 -69.5   115      941
## # ... with 10,000 more rows, and 3 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>
```

Create a new feature: `average_diameter` which averages the two diameters.

```
storms %<>%
  mutate(average_diameter = (ts_diameter + hu_diameter) / 2 )
storms
```

```
## # A tibble: 10,010 x 15
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12  23.1 -73.7   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18  23   -74.2   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0  22.9 -74.4   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6  23   -74.7   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12  23.4 -74.8   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0  24.3 -74.3   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6  24.8 -73.6   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12  25.4 -72.6   135      934
##  9 Joaq~ hurri~ 4         2015    10     3    18  26.3 -71     130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0  27.4 -69.5   115      941
## # ... with 10,000 more rows, and 4 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>,
## #   average_diameter <dbl>
```

At home: calculate the distance from each storm observation to Miami in a new variable `distance_to_miami`.

```
compute_glodbe_distance = function(destination, origin){
  rad_E = 3958.8
  dlon = destination[2] - origin[2]
  dlat = destination[1] - origin[1]
  a = (sin(dlat/2))^2 + cos(origin[1]) * cos(destination[1]) * sin(dlon/2)^2
  c = 2 * atan2(sqrt(a), sqrt(1-a))
  rad_E = 3958.8
  d = rad_E * c
  d
}

#storms %<>%
#  mutate(distance_to_miami = compute_glodbe_distance(miami_coords, origin_coords)
```

At home: convert year, month, day, hour into the variable `timestamp` using the `lubridate` package.

```
#TO-DO
```

At home: using the `lubridate` package, create new variables `day_of_week` which is a factor with levels "Sunday", "Monday", ... "Saturday" and `week_of_year` which is integer 1, 2, ..., 52.

```
#TO-DO
```

Create a new data frame `serious_storms` which are category 3 and above hurricanes.

```
serious_storms = storms %>%
  filter(category >= 3)
serious_storms
```

```
## # A tibble: 779 x 15
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12  23.1 -73.7   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18  23   -74.2   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0  22.9 -74.4   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6  23   -74.7   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12  23.4 -74.8   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0  24.3 -74.3   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6  24.8 -73.6   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12  25.4 -72.6   135      934
##  9 Joaq~ hurri~ 4         2015    10     3    18  26.3 -71     130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0  27.4 -69.5   115      941
## # ... with 769 more rows, and 4 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>,
## #   average_diameter <dbl>
```

In `serious_storms`, merge the variables lat and long together into `lat-long` with values `lat / long` as a string.

```
serious_storms %<>%
  unite(lat_long, lat, long, sep = " / ")
serious_storms
```

```
## # A tibble: 779 x 14
##    name  status category  year month   day  hour lat_long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <chr>    <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12 23.1 / ~   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18 23 / -7~   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0 22.9 / ~   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6 23 / -7~   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12 23.4 / ~   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0 24.3 / ~   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6 24.8 / ~   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12 25.4 / ~   135      934
##  9 Joaq~ hurri~ 4         2015    10     3    18 26.3 / ~   130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0 27.4 / ~   115      941
## # ... with 769 more rows, and 4 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>,
## #   average_diameter <dbl>
```

Back to the main dataframe `storms`, create a new feature `decile_windspeed` by binning wind speed into 10 bins.

```
storms %<>%
  mutate(decile_windspeed = factor(ntile(wind, 10)))
storms
```

```
## # A tibble: 10,010 x 16
##    name  status category  year month   day  hour   lat  long  wind pressure
##    <fct> <fct>  <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl> <int>    <int>
##  1 Joaq~ hurri~ 4         2015    10     1    12  23.1 -73.7   115      942
##  2 Joaq~ hurri~ 4         2015    10     1    18  23   -74.2   115      936
##  3 Joaq~ hurri~ 4         2015    10     2     0  22.9 -74.4   120      931
##  4 Joaq~ hurri~ 4         2015    10     2     6  23   -74.7   120      935
##  5 Joaq~ hurri~ 4         2015    10     2    12  23.4 -74.8   115      937
##  6 Joaq~ hurri~ 4         2015    10     3     0  24.3 -74.3   115      943
##  7 Joaq~ hurri~ 4         2015    10     3     6  24.8 -73.6   120      945
##  8 Joaq~ hurri~ 4         2015    10     3    12  25.4 -72.6   135      934
##  9 Joaq~ hurri~ 4         2015    10     3    18  26.3 -71     130      934
## 10 Joaq~ hurri~ 4         2015    10     4     0  27.4 -69.5   115      941
## # ... with 10,000 more rows, and 5 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>, wind_speed_per_unit_pressure <dbl>,
## #   average_diameter <dbl>, decile_windspeed <fct>
```

Let's summarize some data. Find the strongest storm by wind speed per year.

```
storms %<>%
  group_by(year) %>%
  summarize(max_wind_speed = max(wind))
  storms
```

```
## # A tibble: 41 x 2
##     year max_wind_speed
##    <dbl>          <dbl>
##  1  1975            100
##  2  1976            105
```

```
##  3  1977            150
##  4  1978             80
##  5  1979            150
##  6  1980             90
##  7  1981            115
##  8  1982            115
##  9  1983            100
## 10  1984            115
## # ... with 31 more rows
```

For each status, find the average category, wind speed, pressure and diameters (do not allow the average to be NA).

For each named storm, find its maximum category, wind speed, pressure and diameters (do not allow the max to be NA) and the number of readings (i.e. observations).

```
#TO-DO
```

For each category, find its average wind speed, pressure and diameters (do not allow the max to be NA).

```
#TO-DO
```

At home: for each named storm, find its duration in hours.

```
#TO-DO
```

For each named storm, find the distance from its starting position to ending position in kilometers.

```
#TO-DO
```

Now we want to transition to building real design matrices for prediction. We want to predict the following: given the first three readings of a storm, can you predict its maximum wind speed? Identify the `y` and identify which features you need $x_1, ...x_p$ and build that matrix with `dplyr` functions. This is not easy, but it is what it's all about. Feel free to "featurize" (as Dana Chandler spoke about) as creatively as you would like. You aren't going to overfit if you only build a few features relative to the total 198 storms.

```
#TO-DO
```

Interactions in linear models

Load the Boston Housing Data from package `MASS` and use `str` and `summary` to remind yourself of the features and their types and then use `?MASS::Boston` to read an English description of the features.

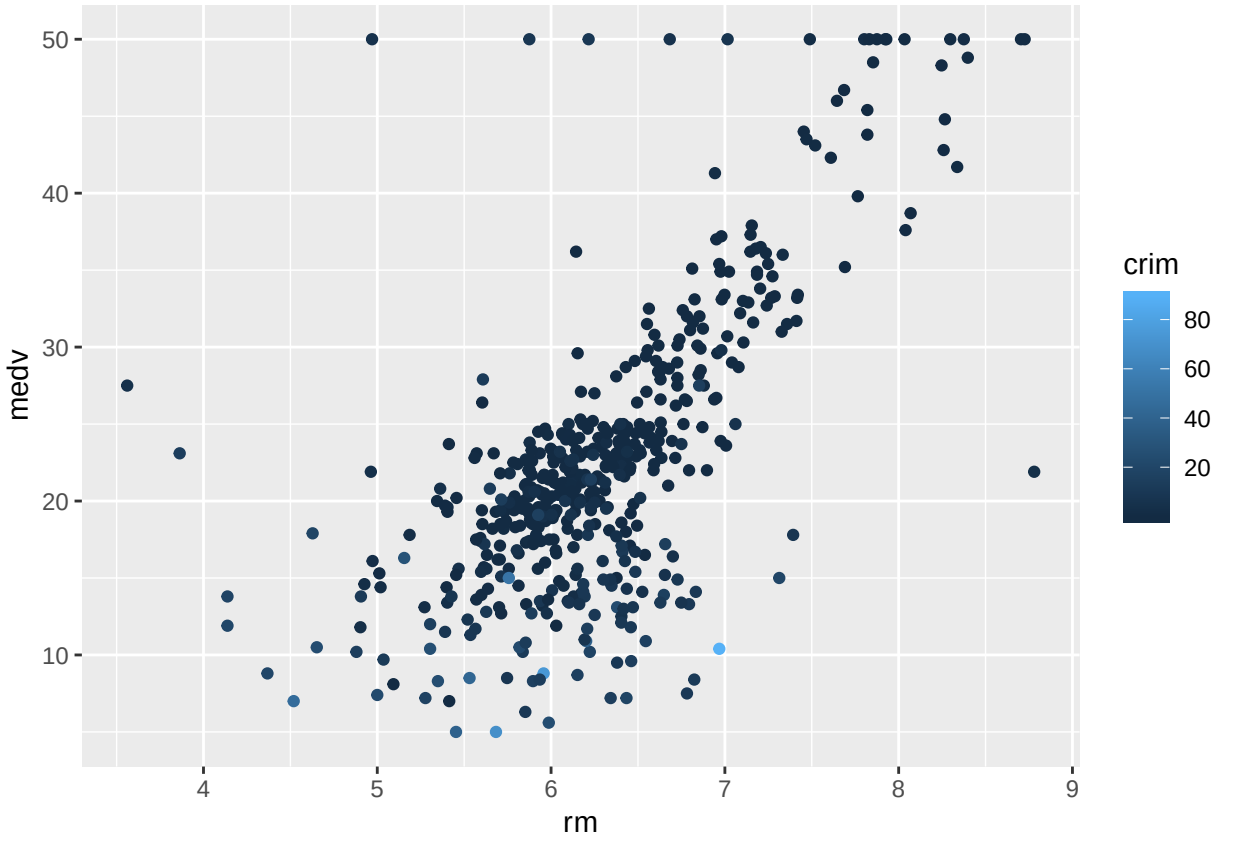
```
boston = MASS::Boston
summary(boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
```

```
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm              age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

Using your knowledge of the modeling problem, try to guess which features are interacting. Confirm using plots in `ggplot` that illustrate three (or more) features.

```
pacman::p_load(ggplot2)
base = ggplot(boston, aes(x = rm, y = medv))
base + geom_point(aes(col = crim))
```



Once an interaction has been located, confirm the "non-linear linear" model with the interaction term does better than just the vanilla linear model.

```
model = lm(medv ~ rm * crim, boston)
coef(model)
```

```
## (Intercept)          rm        crim     rm:crim
##  -37.257338    9.651470    1.462943   -0.287657
```

```
model_vanilla = lm(medv ~ rm + crim, boston)
summary(model_vanilla)$r.squared
```

```
## [1] 0.5419592
```

```
summary(model_vanilla)$sigma
```

```
## [1] 6.236844
```

```
summary(model)$r.squared
```

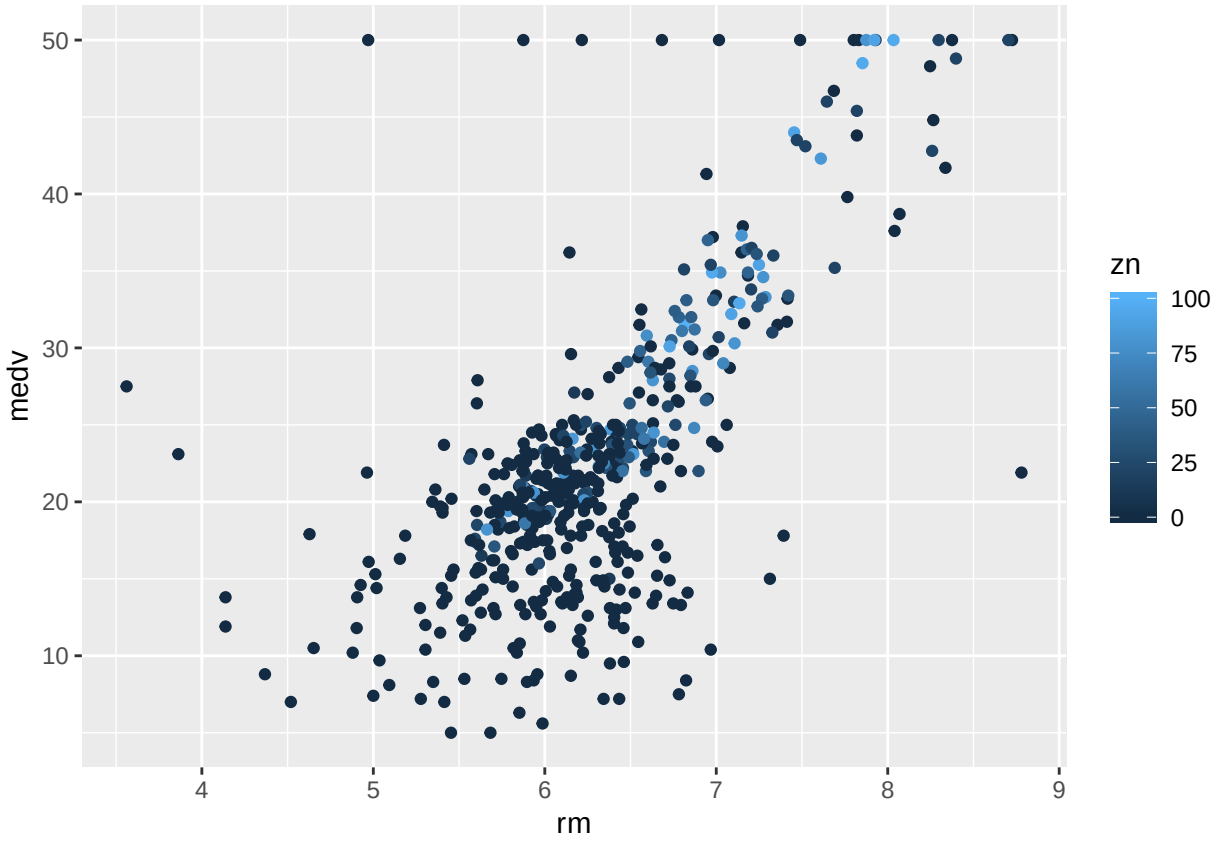
```
## [1] 0.5814763
```

```
summary(model)$sigma
```

```
## [1] 5.967672
```

Repeat this procedure for another interaction with two different features (not used in the previous interaction you found) and verify.

```
base + geom_point(aes(col = zn))
```




```
model = lm(medv ~ rm * zn, boston)
coef(model)
```

```
## (Intercept)          rm          zn       rm:zn
## -26.9934476   7.7661501  -0.4697937   0.0791624
```

```
model_vanilla = lm(medv ~ rm + zn, boston)
summary(model_vanilla)$r.squared
```

```
## [1] 0.5063381
```

```
summary(model_vanilla)$sigma
```

```
## [1] 6.474818
```

```r
summary(model)$r.squared
```

```
## [1] 0.5223732
```

```r
summary(model)$sigma
```

```
## [1] 6.375133
```

Fit a model using all possible first-order interactions. Verify it is "better" than the linear model. Do you think you overfit? Why or why not?

```r
#TO-DO
```

CV

Use 5-fold CV to estimate the generalization error of the model with all interactions.

```r
#TO-DO
```