Computer Assignment 0: Getting Started with R

[STOR 565](file:///C:\Users\murph\Downloads\xxx)

January, 2020

Table of Contents

[Credit 1](#_Toc27641604)

[Stuff to install 2](#_Toc27641605)

[Latex 2](#_Toc27641606)

[R 2](#_Toc27641607)

[R Studio 2](#_Toc27641608)

[Updating R/R Studio 3](#_Toc27641609)

[Basic R commands 3](#_Toc27641610)

[Packages 4](#_Toc27641611)

[IMBD 5](#_Toc27641612)

[Numerical summaries 5](#_Toc27641613)

[Vizualization 11](#_Toc27641614)

[Getting help 15](#_Toc27641615)

[Literate programming and R Markdown 15](#_Toc27641616)

[Additional references 16](#_Toc27641617)

# Credit

**This part of Homework 0 is point-free. You are only required to go through “Stuff to Install”, “Basic R Commands”, “Getting help”, and “Literate programming and R Markdown”. You are STRONGLY ENCOURAGED to use this document as a resource when using R for the remainder of this class. You will be using R Markdown a lot for homework, so it would be best to get comfortable with it now.**

Most of these notes were written by [Iain Carmichael](https://idc9.github.io) for the STOR data science course, and edited by [Alexander Murph](https://sirmurphalot.github.io/) for this semester’s machine learning course. The text below gives you the basics on what you need to get started for the course.

# Stuff to install

We will begin by walking you through the software you will need to install on your computer. Being comfortable with this software will be essential for your success in this class, and potentially for your success in your future career path. If you are having any issues with installation, please visit [Murph’s](https://sirmurphalot.github.io/) office hours as soon as possible for assistance.

## Latex

This part is only tangential to the course however it is required to compile various math formulae in the homework. Go to

[Latex project](https://www.latex-project.org/get/)

and install the Latex distribution appropriate for your computer. (For Windows we recommend MikTex, while for Macs there is only MacTex). You should do this before installing R and RStudio. If you already have R and Rstudio installed, that is ok, but later in the course as you do HWs and “knit” your HW, if you are running into issues, one method that seemed to work was reinstalling R and Rstudio.

## R

**Download the latest version of R from the** [**Comprehensive R Archive Network**](https://cran.rstudio.com/) (CRAN). [R](https://www.r-project.org/about.html) is a programming language built for statistical analysis.

## R Studio

**Download** [**R Studio**](https://www.rstudio.com/products/rstudio/#Desktop) which is an [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) built for R. While you can use R without R Studio, R Studio makes life much better.

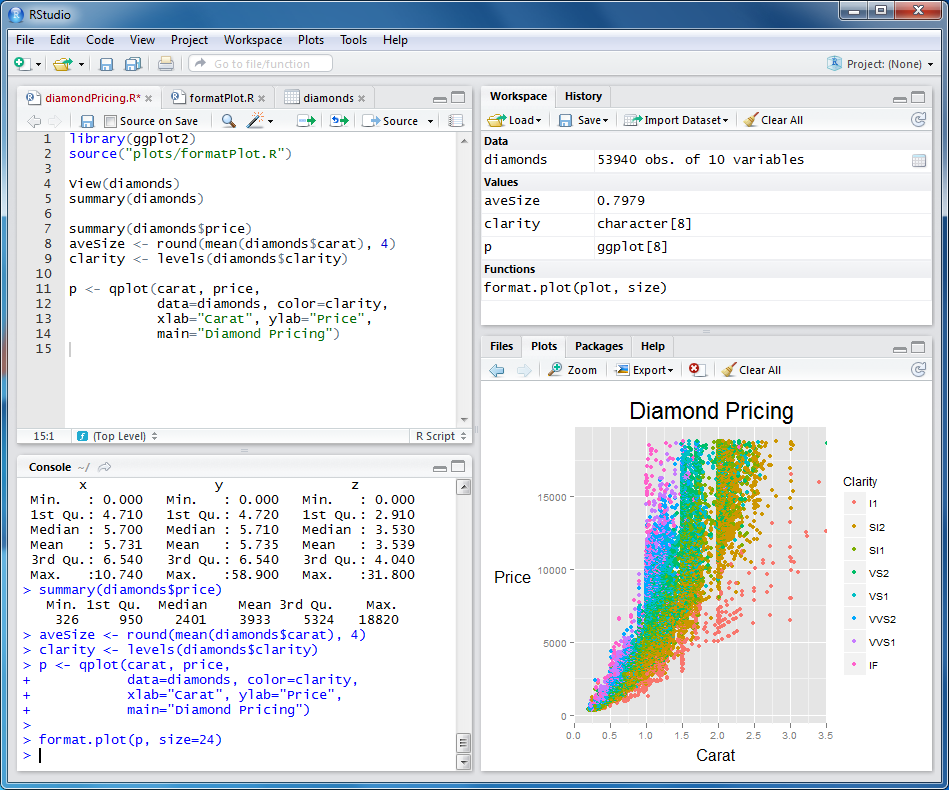


image from <http://rprogramming.net/download-and-install-rstudio/>

Watch the [video on this page](https://www.rstudio.com/products/RStudio/) for the basics parts of R Studio. The lower-left box entitled “Console” allows you to run commands in R (see the next section *Basic R Commands*), which you do by typing the command and pressing *enter*. You can also write commands in the top-left box and press *command+enter* to run them. Try running the following *Basic R Commands* using each of these described methods.

## Updating R/R Studio

If you already have R and R Studio please update both of them. For instructions see [this page](http://bioinfo.umassmed.edu/bootstrappers/bootstrappers-courses/courses/rCourse/Additional_Resources/Updating_R.html).

# Basic R commands

Once you have downloaded [base R](https://www.rstudio.com/wp-content/uploads/2016/05/base-r.pdf) from CRAN, run try running some commands using your R console. In the shaded section below, we write out a series of R commands that you should try out on your own console. In R, the hashtags “##” at the start of a line represent the start of code comments, which will do nothing if you attempt to run them. They simply exist to provide more information to humans who wish to read the code. In our code comments here, we have given the output you should expect when you run the code yourself.

1 + 1

## [1] 2

a <- 1  
b <- 2  
a + b

## [1] 3

If you are new to R we suggest reading through [before we start](http://www.datacarpentry.org/R-ecology-lesson/00-before-we-start.html) and [intro to R](http://www.datacarpentry.org/R-ecology-lesson/01-intro-to-R.html).

Being able to run R commands is the first, and most important, thing you should gain from this homework. We will expect from here on out that you are able to run code in R, and understand the output. If you need help, get help!

# Packages

The R language is particularly powerful due to the fact that it is constantly improving and expanding. R is an [open source](https://en.wikipedia.org/wiki/Open-source_software) language, meaning anyone can add new functionality to R by the way of “R packages”. This is particularly exciting, since it means that as a language R is constantly improving and adding new commands and functions. If there is something that R cannot currently do, any R programmer can write a package that can then be accessed and used by other users.

You can install a package from [CRAN](https://cran.r-project.org/) like this

install.packages("tidyverse")

To get started for the course you will need to install the following packages. (To do this, run the text below in the console in RStudio.)

install.packages("tidyverse")  
install.packages("devtools")  
devtools::install\_github("hadley/r4ds")  
install.packages("ISLR")  
install.packages("caret") # for machine learning tasks  
install.packages("leaps") # for subset selection in regression

You only need to install an R package once. You need to load an R package every time you want to use it, which you do with the “library()” command. To reiterate, installing means using the “install.packages()” command, while loading means using the “library()” command.

# ignore the warnings for now  
library(tidyverse)

## ── Attaching packages ───────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.0.0 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.6  
## ✔ tidyr 0.8.1 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

**Moving forward.** The rest of these notes introduce you to some basic commands in R using an example dataset. The machine learning course is **not** a programming course so don’t worry too much about knowing all the ins and outs of R. However, as we proceed with the class, you can pick up things as you need them from two good resources: Google and Stack Overflow.

# IMBD

Load the [movies data set](http://www2.stat.duke.edu/~mc301/data/movies.html) generously curated by [Mine Cetinkaya-Rundel](http://www2.stat.duke.edu/~mc301/). In this data, we have information about many different movies based on the database from *IMDB.com*.

# downloads data set and loads it into R  
load(url('https://stat.duke.edu/~mc301/data/movies.Rdata'))

The first thing you should do when you get a data set is [look at it](https://www.youtube.com/watch?v=EF8GhC-T_Mo)!

## Numerical summaries

By default, R often loads data into a R object called a data.frame. For the time being, we will have you play around with this object without fully explaining what it means to be a data frame. In Computer Assignment 1, we will review the different data objects in R, especially the data.frame object.

So, let’s examine our data. The str() command tells you about the data frame. The first thing to note is the dimension of the data frame (651 rows by 32 columns) and the column types.

str(movies)

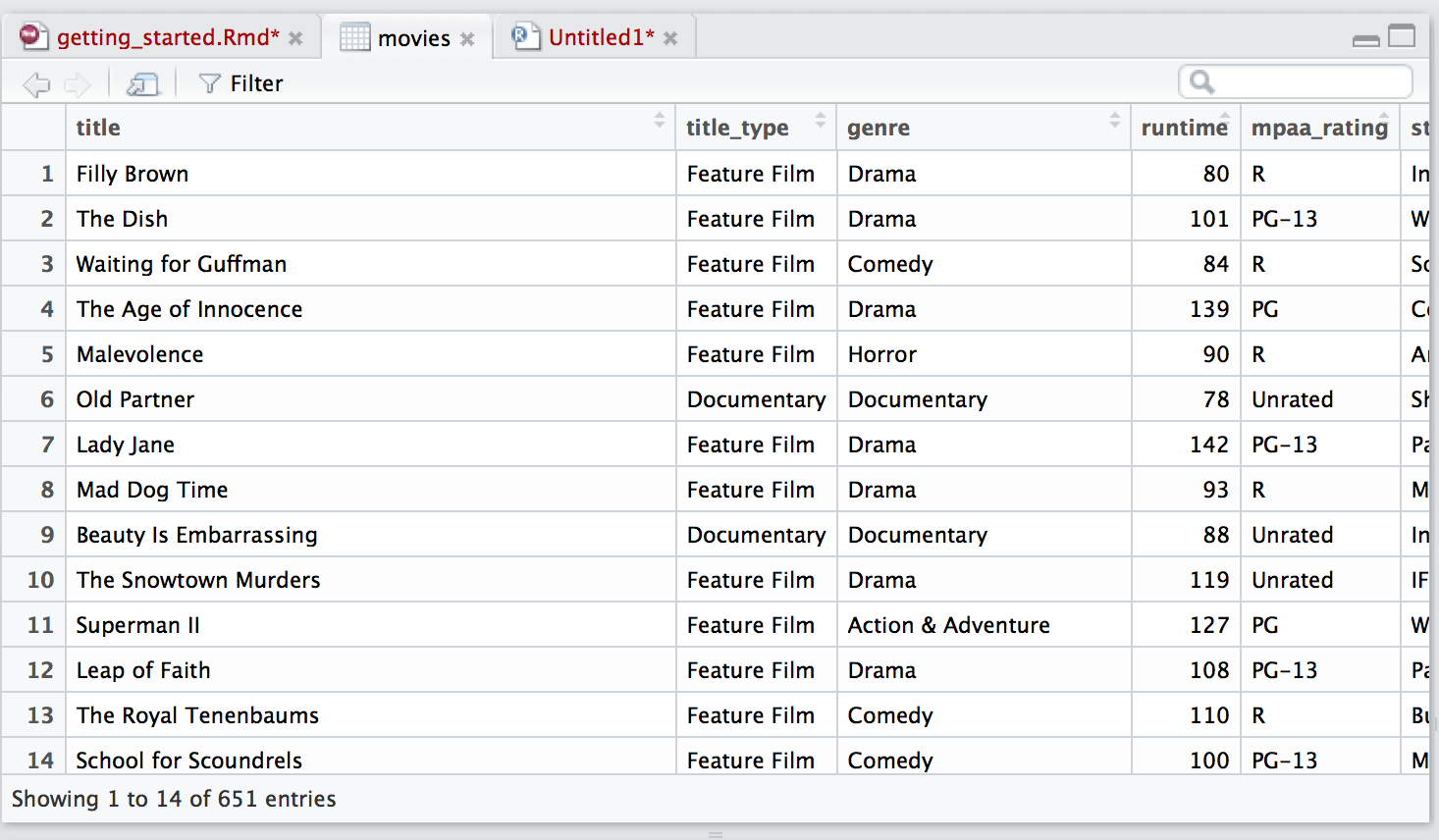
## Classes 'tbl\_df', 'tbl' and 'data.frame': 651 obs. of 32 variables:  
## $ title : chr "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocence" ...  
## $ title\_type : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...  
## $ genre : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...  
## $ runtime : num 80 101 84 139 90 78 142 93 88 119 ...  
## $ mpaa\_rating : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...  
## $ studio : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84 ...  
## $ thtr\_rel\_year : num 2013 2001 1996 1993 2004 ...  
## $ thtr\_rel\_month : num 4 3 8 10 9 1 1 11 9 3 ...  
## $ thtr\_rel\_day : num 19 14 21 1 10 15 1 8 7 2 ...  
## $ dvd\_rel\_year : num 2013 2001 2001 2001 2005 ...  
## $ dvd\_rel\_month : num 7 8 8 11 4 4 2 3 1 8 ...  
## $ dvd\_rel\_day : num 30 28 21 6 19 20 18 2 21 14 ...  
## $ imdb\_rating : num 5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...  
## $ imdb\_num\_votes : int 899 12285 22381 35096 2386 333 5016 2272 880 12496 ...  
## $ critics\_rating : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...  
## $ critics\_score : num 45 96 91 80 33 91 57 17 90 83 ...  
## $ audience\_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...  
## $ audience\_score : num 73 81 91 76 27 86 76 47 89 66 ...  
## $ best\_pic\_nom : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ best\_pic\_win : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ best\_actor\_win : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...  
## $ best\_actress\_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ best\_dir\_win : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...  
## $ top200\_box : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
## $ director : chr "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese" ...  
## $ actor1 : chr "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis" ...  
## $ actor2 : chr "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfeiffer" ...  
## $ actor3 : chr "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ryder" ...  
## $ actor4 : chr "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...  
## $ actor5 : chr "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen" ...  
## $ imdb\_url : chr "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/tt0205873/" "http://www.imdb.com/title/tt0118111/" "http://www.imdb.com/title/tt0106226/" ...  
## $ rt\_url : chr "//www.rottentomatoes.com/m/filly\_brown\_2012/" "//www.rottentomatoes.com/m/dish/" "//www.rottentomatoes.com/m/waiting\_for\_guffman/" "//www.rottentomatoes.com/m/age\_of\_innocence/" ...

There is a lot to unpack here! The first line tells you the dimension of the data: 651 rows and 32 columns. The following lines gives you information about each of the 32 columns. A “$” marks a new column, which is followed by the name of the column, a colon, then information concerning the type of data found in that column.

The head() command prints the first six rows of a data set (and as many columns that will fit on the screen). This is a great command that allows you to examine the data.

head(movies)

## # A tibble: 6 x 32  
## title title\_type genre runtime mpaa\_rating studio thtr\_rel\_year  
## <chr> <fct> <fct> <dbl> <fct> <fct> <dbl>  
## 1 Fill… Feature F… Drama 80 R Indom… 2013  
## 2 The … Feature F… Drama 101 PG-13 Warne… 2001  
## 3 Wait… Feature F… Come… 84 R Sony … 1996  
## 4 The … Feature F… Drama 139 PG Colum… 1993  
## 5 Male… Feature F… Horr… 90 R Ancho… 2004  
## 6 Old … Documenta… Docu… 78 Unrated Shcal… 2009  
## # ... with 25 more variables: thtr\_rel\_month <dbl>, thtr\_rel\_day <dbl>,  
## # dvd\_rel\_year <dbl>, dvd\_rel\_month <dbl>, dvd\_rel\_day <dbl>,  
## # imdb\_rating <dbl>, imdb\_num\_votes <int>, critics\_rating <fct>,  
## # critics\_score <dbl>, audience\_rating <fct>, audience\_score <dbl>,  
## # best\_pic\_nom <fct>, best\_pic\_win <fct>, best\_actor\_win <fct>,  
## # best\_actress\_win <fct>, best\_dir\_win <fct>, top200\_box <fct>,  
## # director <chr>, actor1 <chr>, actor2 <chr>, actor3 <chr>,  
## # actor4 <chr>, actor5 <chr>, imdb\_url <chr>, rt\_url <chr>

If you double click a data frame (which you can do from the top-right section of your RStudio console) it will pull up R’s built-in spreadsheet:

The summary() command prints out some descriptive statistics for each column

summary(movies)

## title title\_type genre   
## Length:651 Documentary : 55 Drama :305   
## Class :character Feature Film:591 Comedy : 87   
## Mode :character TV Movie : 5 Action & Adventure: 65   
## Mystery & Suspense: 59   
## Documentary : 52   
## Horror : 23   
## (Other) : 60   
## runtime mpaa\_rating studio   
## Min. : 39.0 G : 19 Paramount Pictures : 37   
## 1st Qu.: 92.0 NC-17 : 2 Warner Bros. Pictures : 30   
## Median :103.0 PG :118 Sony Pictures Home Entertainment: 27   
## Mean :105.8 PG-13 :133 Universal Pictures : 23   
## 3rd Qu.:115.8 R :329 Warner Home Video : 19   
## Max. :267.0 Unrated: 50 (Other) :507   
## NA's :1 NA's : 8   
## thtr\_rel\_year thtr\_rel\_month thtr\_rel\_day dvd\_rel\_year   
## Min. :1970 Min. : 1.00 Min. : 1.00 Min. :1991   
## 1st Qu.:1990 1st Qu.: 4.00 1st Qu.: 7.00 1st Qu.:2001   
## Median :2000 Median : 7.00 Median :15.00 Median :2004   
## Mean :1998 Mean : 6.74 Mean :14.42 Mean :2004   
## 3rd Qu.:2007 3rd Qu.:10.00 3rd Qu.:21.00 3rd Qu.:2008   
## Max. :2014 Max. :12.00 Max. :31.00 Max. :2015   
## NA's :8   
## dvd\_rel\_month dvd\_rel\_day imdb\_rating imdb\_num\_votes   
## Min. : 1.000 Min. : 1.00 Min. :1.900 Min. : 180   
## 1st Qu.: 3.000 1st Qu.: 7.00 1st Qu.:5.900 1st Qu.: 4546   
## Median : 6.000 Median :15.00 Median :6.600 Median : 15116   
## Mean : 6.333 Mean :15.01 Mean :6.493 Mean : 57533   
## 3rd Qu.: 9.000 3rd Qu.:23.00 3rd Qu.:7.300 3rd Qu.: 58300   
## Max. :12.000 Max. :31.00 Max. :9.000 Max. :893008   
## NA's :8 NA's :8   
## critics\_rating critics\_score audience\_rating audience\_score   
## Certified Fresh:135 Min. : 1.00 Spilled:275 Min. :11.00   
## Fresh :209 1st Qu.: 33.00 Upright:376 1st Qu.:46.00   
## Rotten :307 Median : 61.00 Median :65.00   
## Mean : 57.69 Mean :62.36   
## 3rd Qu.: 83.00 3rd Qu.:80.00   
## Max. :100.00 Max. :97.00   
##   
## best\_pic\_nom best\_pic\_win best\_actor\_win best\_actress\_win best\_dir\_win  
## no :629 no :644 no :558 no :579 no :608   
## yes: 22 yes: 7 yes: 93 yes: 72 yes: 43   
##   
##   
##   
##   
##   
## top200\_box director actor1 actor2   
## no :636 Length:651 Length:651 Length:651   
## yes: 15 Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## actor3 actor4 actor5   
## Length:651 Length:651 Length:651   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## imdb\_url rt\_url   
## Length:651 Length:651   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##   
##

Say you wanted to look at just one of the 32 columns. The $ sign after a data.frame, followed by the name of column, will print out the data in that column.

movies$imdb\_num\_votes

## [1] 899 12285 22381 35096 2386 333 5016 2272 880 12496  
## [11] 71979 9669 201779 25808 5544 240033 66489 6336 37769 21268  
## [21] 56201 3459 16717 9357 4541 1816 163490 19285 3688 3488  
## [31] 82851 56128 33101 259822 4908 290356 47297 2145 466400 69338  
## [41] 6228 75468 48519 13523 58668 9654 22079 5258 761 33839  
## [51] 183 225130 62241 19714 137126 10468 37770 1628 5587 6247  
## [61] 8646 8320 2362 1942 110238 21501 14359 12450 21704 4375  
## [71] 48324 4768 2944 6788 24783 3745 54726 1493 5616 10492  
## [81] 4451 12269 60335 5035 315051 82737 4944 8229 4516 7858  
## [91] 10938 279704 11192 3851 1838 6552 390 68429 182983 1147  
## [101] 35635 2569 13980 11477 9853 14970 9001 35868 41767 6954  
## [111] 893008 3467 117688 2817 32751 5704 318 123989 70209 541  
## [121] 132215 127458 27769 52635 12819 6054 9367 17798 26010 4515  
## [131] 535 71572 57933 1784 3883 2282 105745 104457 205065 562136  
## [141] 1778 7628 17934 2502 13092 1010 4687 48137 164112 30921  
## [151] 37640 95327 289825 15525 43268 3153 1887 3096 112216 154674  
## [161] 12535 3363 40659 82378 3135 1489 41385 88523 12221 6304  
## [171] 9565 15913 414687 33720 325 22245 17133 23821 13285 192052  
## [181] 211129 146518 2239 486 7658 1268 64119 24678 6114 9399  
## [191] 414650 15291 725 93331 9876 9525 44741 285 3673 79970  
## [201] 1058 60220 5591 340 50340 2934 5564 753592 1141 3967  
## [211] 287476 94983 56185 3138 25264 9424 16681 1308 87215 121245  
## [221] 71112 72295 2289 84191 235529 168032 56329 2849 26360 5762  
## [231] 8521 191935 872 8999 184656 2098 375820 19187 1406 16755  
## [241] 9003 5014 2818 38076 137405 56361 7244 2701 18712 25054  
## [251] 1510 8544 8561 151934 52449 11001 42613 3505 74294 1480  
## [261] 703 103789 42842 26731 26628 149437 9291 157701 14901 1361  
## [271] 47065 12606 246587 42208 9216 201787 19000 1799 12877 9025  
## [281] 5985 30886 4970 2732 3336 252661 30641 22601 7076 161601  
## [291] 285328 2598 108598 20655 12322 99192 44257 77762 368799 2830  
## [301] 2960 46233 9980 8016 37938 16480 63219 4251 23697 53675  
## [311] 9787 1978 9370 10380 73280 7656 11855 172765 23201 8604  
## [321] 14949 1995 15806 30694 56888 4821 3145 19115 70994 66233  
## [331] 2295 2698 34652 739 64489 49985 3649 3359 10020 78726  
## [341] 246907 49374 40001 18670 5149 35577 38076 87652 17329 16511  
## [351] 5002 749783 10599 3101 4180 9904 3428 2056 100447 37506  
## [361] 109633 21443 4031 47692 47343 24084 13215 105982 34461 15714  
## [371] 3358 275125 13280 2551 5863 73219 8319 265725 3859 53535  
## [381] 4907 318019 679 806911 3342 3649 27417 1815 9939 4857  
## [391] 1428 44248 18141 303529 86953 128361 297034 4143 3128 36909  
## [401] 1816 1043 54829 9725 490295 7881 5136 10055 2408 24472  
## [411] 115026 651 7710 5425 4550 16262 1571 10271 204042 16824  
## [421] 6061 10651 1346 1886 4874 4121 40133 3416 73617 183747  
## [431] 123588 124250 11103 33040 11236 9946 1680 122980 19603 1663  
## [441] 71141 13790 5374 14589 11259 6472 100416 3866 872 2928  
## [451] 3887 1607 4904 19937 17384 25683 3883 99582 2959 134031  
## [461] 17190 135840 2897 10126 19383 8030 3461 3970 572236 4072  
## [471] 126257 15491 51070 2530 30495 16955 797101 8059 60483 3602  
## [481] 34802 3730 30085 32737 34307 66171 8685 54597 7862 68871  
## [491] 582091 11156 6345 9990 3473 42295 329613 42408 137222 51366  
## [501] 21623 39320 1915 1674 2096 1935 10522 2380 78862 83724  
## [511] 34298 830 2869 134510 152216 54771 11838 110540 6343 309494  
## [521] 6909 1890 72176 2271 6804 161101 10535 448434 2931 21009  
## [531] 6811 1943 32338 19161 54871 2433 21924 128298 14559 34926  
## [541] 27097 4021 764 6418 70737 9656 193702 59076 154148 11377  
## [551] 3302 180 34253 17960 281 12498 86831 83424 1803 56919  
## [561] 10250 18005 62773 15444 48756 14986 13525 246343 290958 3146  
## [571] 10886 96471 17101 723 106171 88777 294683 51534 3487 5115  
## [581] 15449 2181 9832 247105 13614 78297 4369 6765 16137 101850  
## [591] 504 1935 15116 183717 64873 20738 123769 79866 160237 24595  
## [601] 16883 390 19539 48718 26301 26943 4077 63511 27601 756602  
## [611] 3998 19898 10786 2857 9675 7545 2113 3448 2441 12402  
## [621] 3373 6322 9906 15025 30826 309896 7284 58907 57251 3790  
## [631] 8818 11125 675907 2120 111132 103378 13682 63672 6946 3584  
## [641] 54363 11197 96787 16366 134270 11657 8345 46794 10087 66054  
## [651] 43574

The mean() function computes the mean of a vector. A vector is another data structure (like data.frames) that we will cover in Computer Assignment 1. Whenever you select a single column like we did above, it will be returned as a vector. So, if we wanted to know the mean of a column, we would run:

mean(movies$imdb\_num\_votes)

## [1] 57532.98

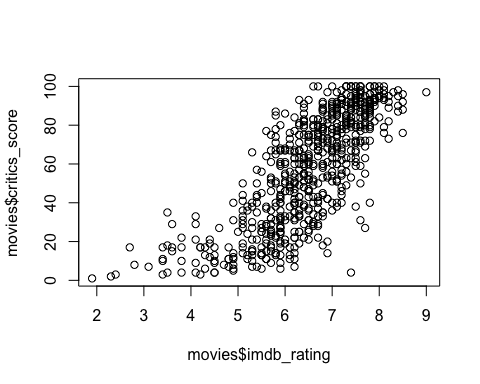
There are also functions for median, var, min, and max. Try them out!

## Visualization

You can only learn so much by looking lists of numbers. Let’s make some plots.

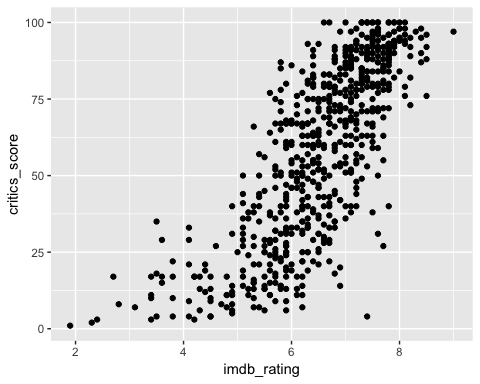
There are two popular plotting systems in R: the *base* R system and ggplot2. Using the base R system we can create a scatterplot of imdb ratings vs. critic scores as follows

plot(movies$imdb\_rating, movies$critics\_score)



Here’s how to do the same thing with ggplot.

# ggplot was loaded with tidyverse  
ggplot(data = movies) + geom\_point(mapping = aes(x = imdb\_rating, y = critics\_score))

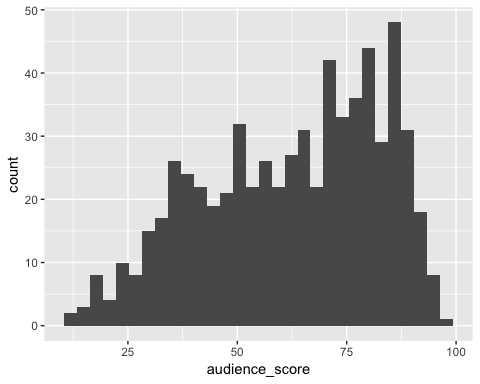


We will often use ggplot2 in this course but it is up to you what you want to use. ggplot2 is significantly more advanced, but comes with a sharper learning curve. See the readings below about ggplot2 vs base plotting in R. Using ggplot2 can be a bit intimidating at first, so there will be a lot of guidance throughout this course on how to draw the plots we want in R. For now, try playing around with the given code for the plots.

Below is a histogram of the IMBD data we used earlier. With more experience, you can add titles, colors, and custom labels to plots like this one.

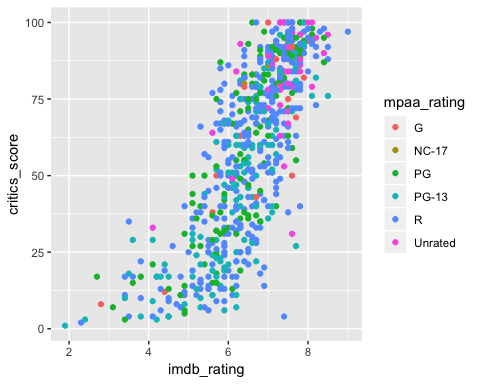
ggplot(data=movies, aes(x=audience\_score)) + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot2 has a ton of functionality built in and you will learn to love it when you get used to it. Observe below for a more sophisticated scatterplot of the IMBD data.

ggplot(data = movies) + geom\_point(mapping = aes(x = imdb\_rating, y = critics\_score, color=mpaa\_rating))



# Getting help

The main online source of help for the computing portion of the course is [R for Data Science](http://r4ds.had.co.nz/data-visualisation.html) written by Hadley Wickham (and free online). In addition, there is also a [long list of alternative resources](https://idc9.github.io/stor390/course_info/references.html) (textbooks, coursera courses, etc.) compiled by Iain Carmichael.

Google and [StackOverflow](http://stackoverflow.com/) are good general resources for R related issues. If you have a question, chances are someone has already asked and answered it. If R gives you an error message you don’t understand Google it – someone else has probably figured it out and posted it online.

# Literate programming and R Markdown

[Literate Programming](https://en.wikipedia.org/wiki/Literate_programming) is a concept introduced by Donald Knuth saying you should *write code that communicates primarily to humans, not computers*. Here are some examples:

* [Jane Austen and R](https://cran.r-project.org/web/packages/tidytext/vignettes/tidytext.html)
* [Document Clustering with Python](http://brandonrose.org/clustering)

[R Markdown](http://rmarkdown.rstudio.com/) allows you easily write documents that contain R code, text, images, links, etc. It may sound bland at first, but R Markdown is super useful. Open a new R Markdown document and play around with it. You can read more [about R Markdown in r4ds](http://r4ds.had.co.nz/r-markdown.html#r-markdown-basics). This webpage may also be helpful in getting started with R Markdown: <http://stat545.com/block007_first-use-rmarkdown.html>

# Additional references

* ggplot2 vs. base plotting
  + [Why I don’t use ggplot2](http://simplystatistics.org/2016/02/11/why-i-dont-use-ggplot2/)
  + [Why I use ggplot2](http://varianceexplained.org/r/why-I-use-ggplot2/)