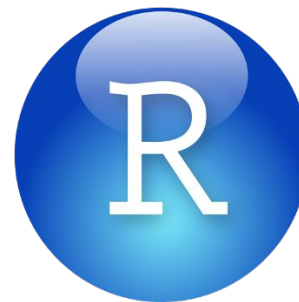




A Short Introduction into Data Science with R



Robert Ladwig
University of Wisconsin-Madison



Disclaimer

- material is based on the **AWESOME** workshop by Rachel Pilla (now postdoc at Oak Ridge National Lab, @rmpilla) “Introduction to R” (*Computer Science in Modern Biology* at Miami University, Ohio)
- with help by Andrew Cannizzarro, Alva Strand and Nicole Berry

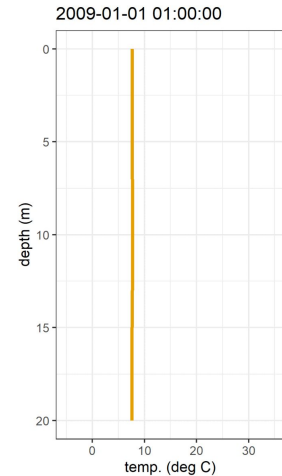


About myself

- **Physical limnologist**
 - use mathematical models to explore mixing dynamics and water quality in lakes
- developing and applying open-source and open-access software in **R**, **Python** and **Matlab**
- background in civil engineering and geology

Aquatic ecosystem modeling

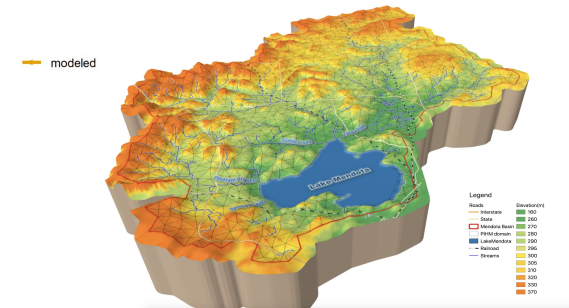
mechanistic understanding of cause and effects



$$\frac{\partial T}{\partial t} = \frac{1}{A} \frac{\partial}{\partial z} \left(A (v_t' + v') \frac{\partial T}{\partial z} \right) + \frac{1}{\rho_0 c_p} \frac{\partial H_{sol}}{\partial z} + \frac{dA}{dz} \frac{H_{geo}}{A \rho_0 c_p}$$

$$\frac{\partial [O_2]}{\partial t} = NEP = GPP - ER \pm D$$

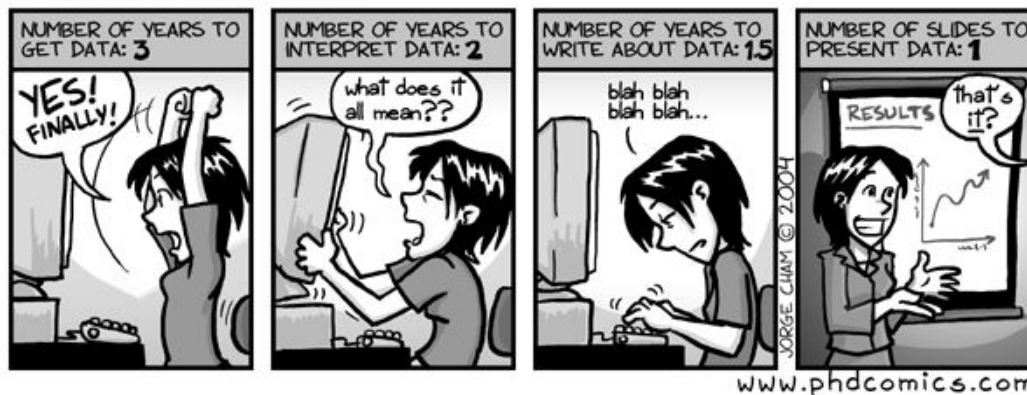
$$\frac{\partial P}{\partial t} = D \frac{\partial^2 P}{\partial z^2} + \lambda P \left(1 - \frac{P}{K} \right)$$



What's data science?

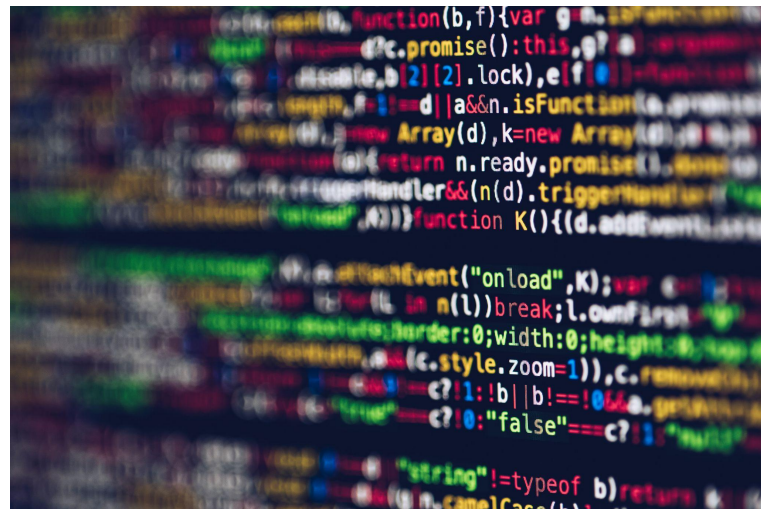
- combining math & statistics, programming and machine learning to explore any kind of data
- data is often noisy or unstructured → cleaning is big part of the job
- buzzwords like big data

DATA: BY THE NUMBERS



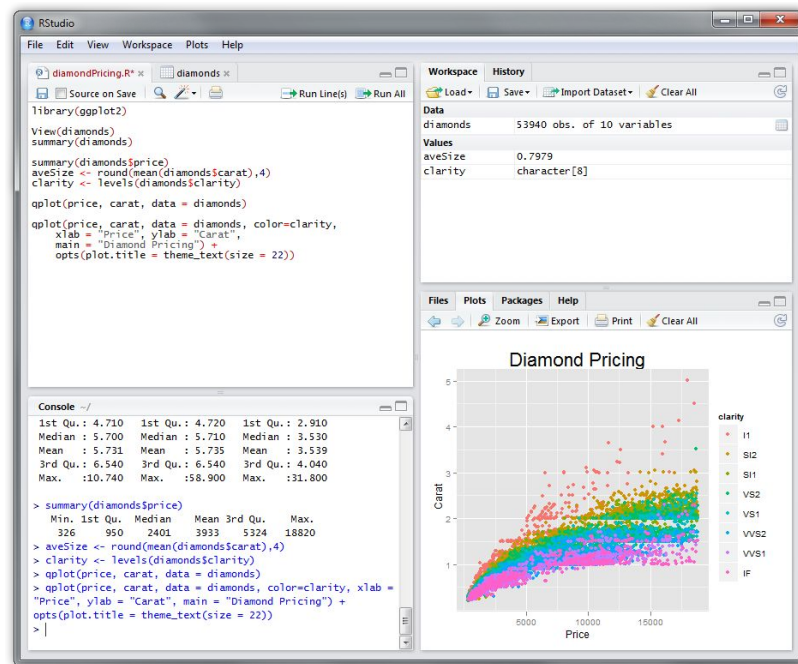
What's the plan for today?

- short introduction into the R scripting language
 - all material on github:
<https://github.com/robertladwig/intro-atasciencer>
 - GitHub: uses Git (version control), fantastic for coding and collaborations
- hands-on coding to introduce you to data analysis
- **at the end:** able to know basic R commands, load data, analyze data, plot data



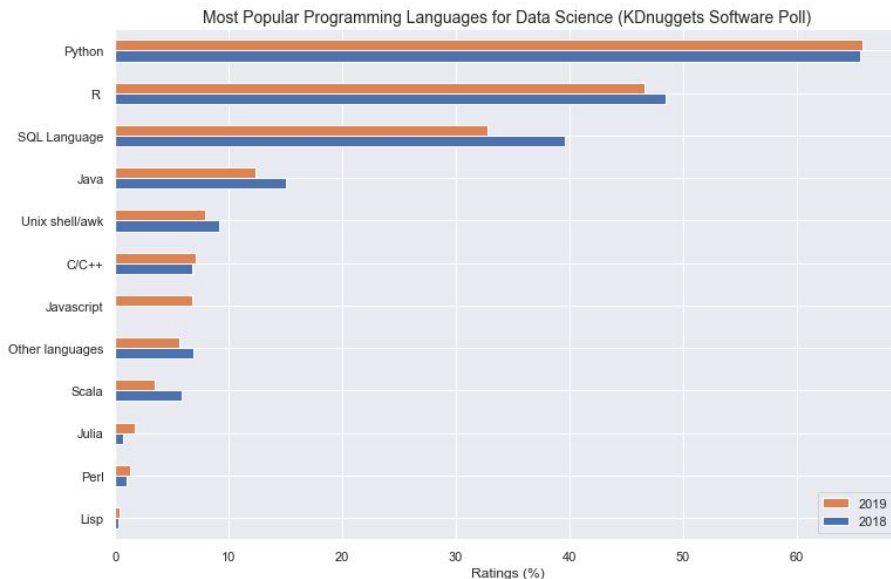
Install R and RStudio

- Install R from here (language):
<https://www.r-project.org/>
- Install RStudio (editor and GUI):
<https://rstudio.com/products/rstudio/download/#download>



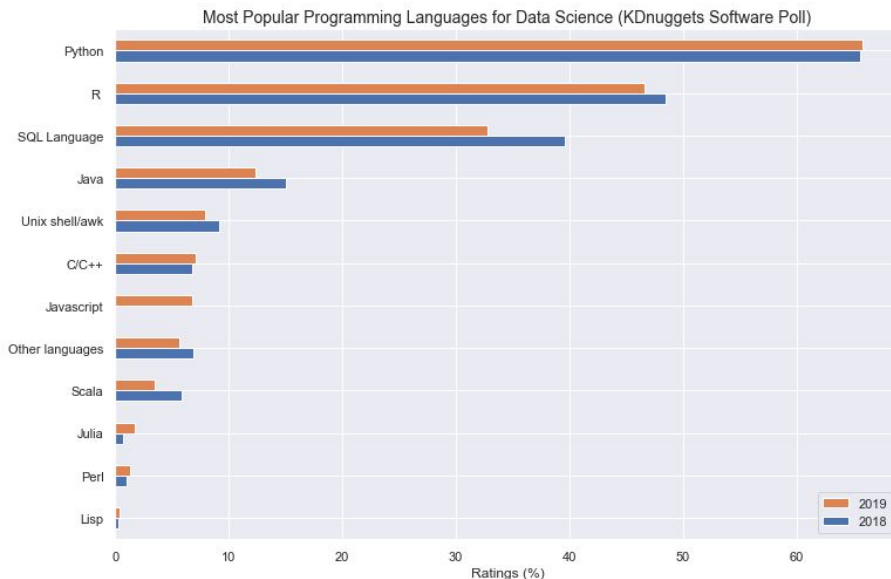
What's R?

- FREE and OPEN SOURCE statistical and computational software



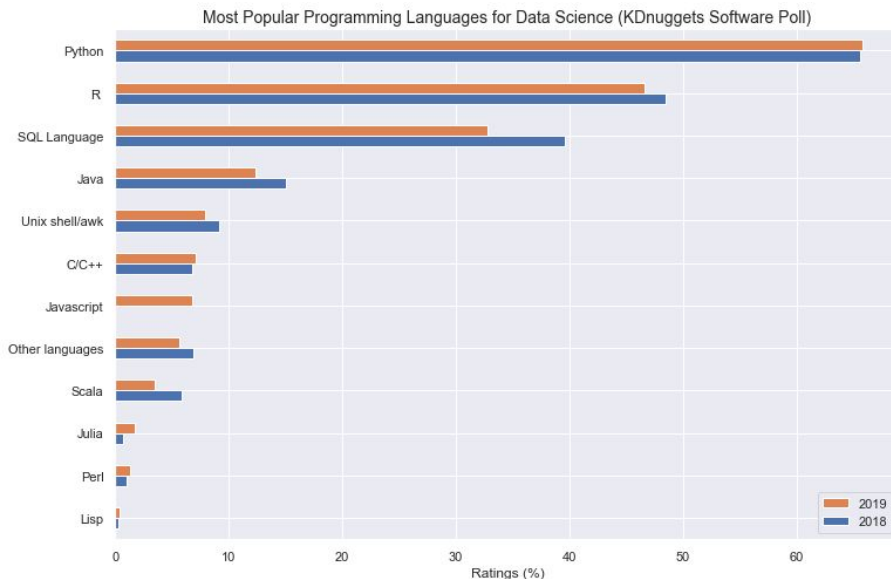
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online



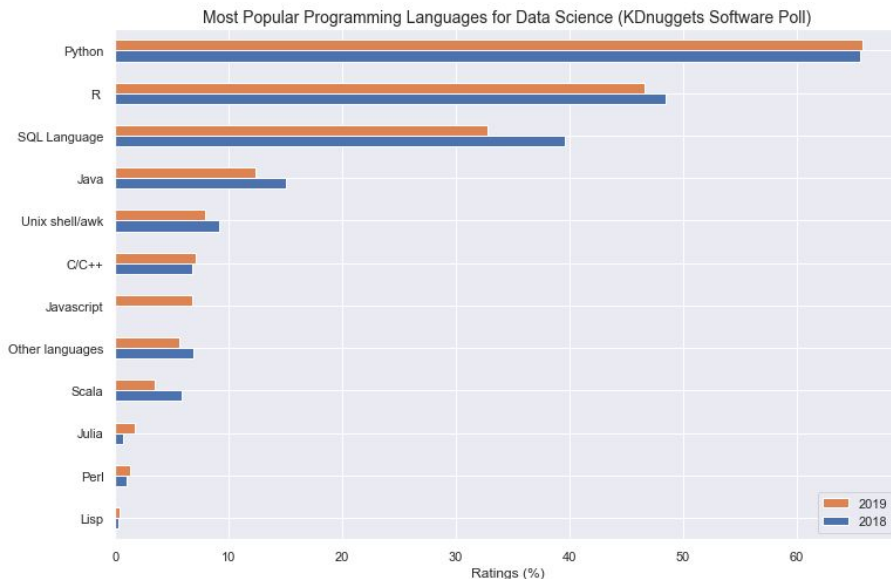
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity



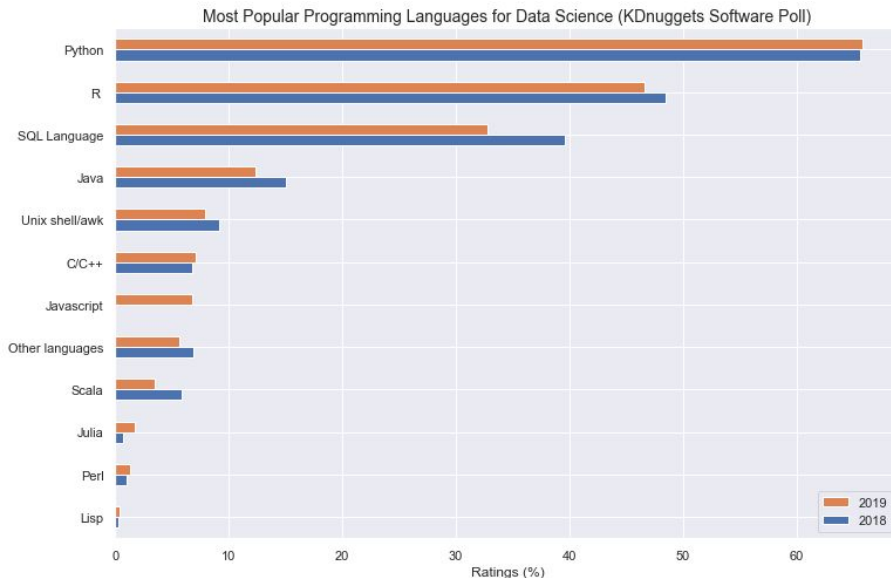
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity
- can handle more advanced computations, statistical analyses and bigger data files than Excel

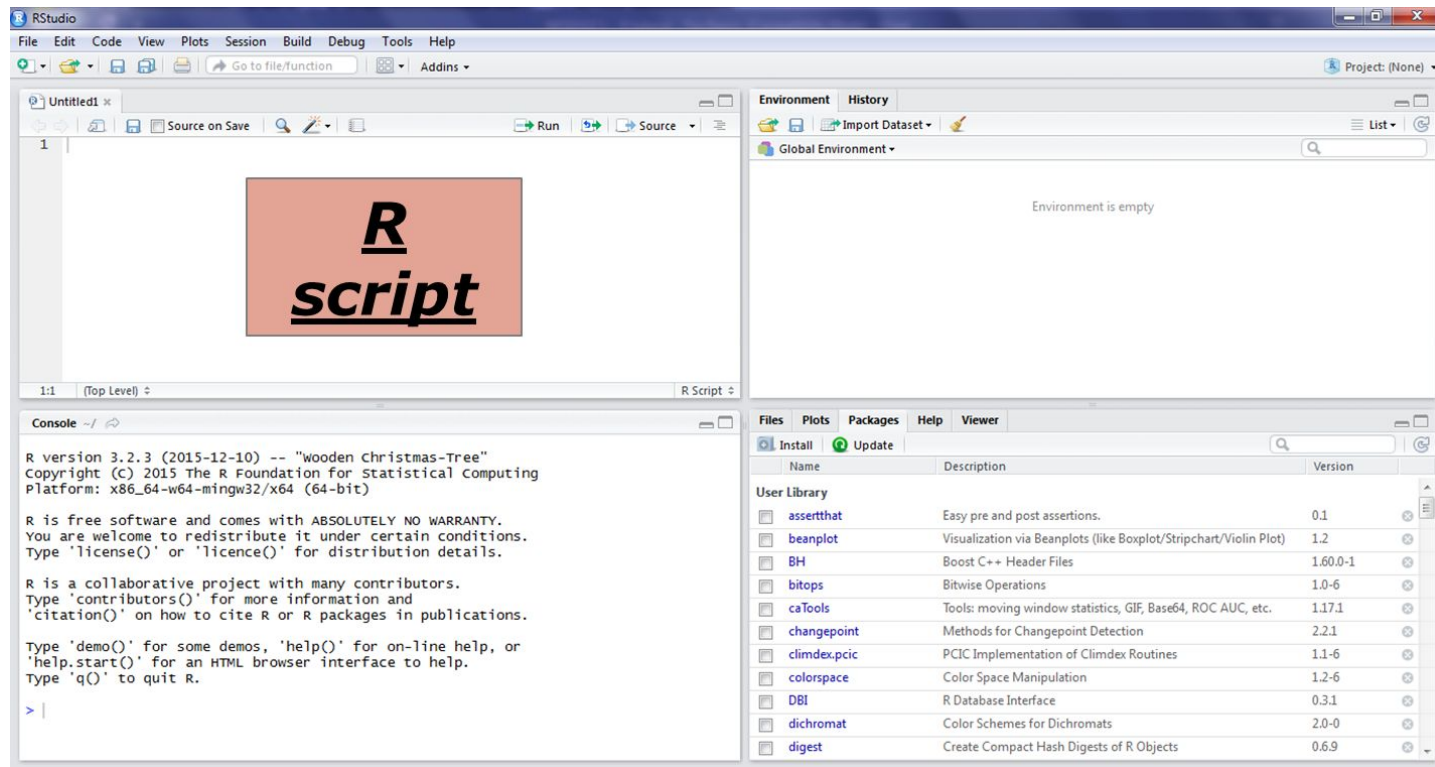


What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity
- can handle more advanced computations, statistical analyses and bigger data files than Excel
- lots of styles for coding



Open RStudio on your laptop now



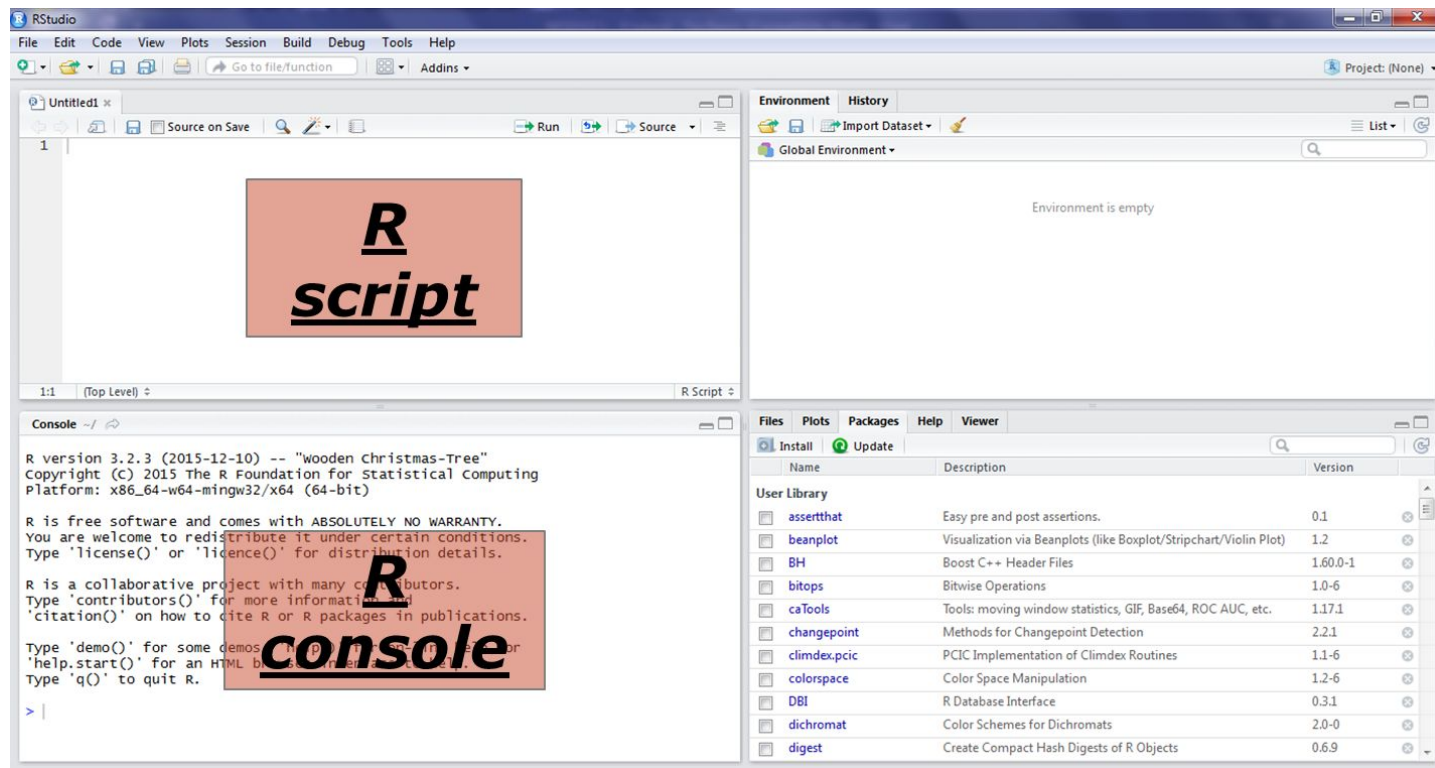
R script:

write and edit
code

color-coding for
easier
identification

automatically fills
in parentheses
and quotations

Open RStudio on your laptop now

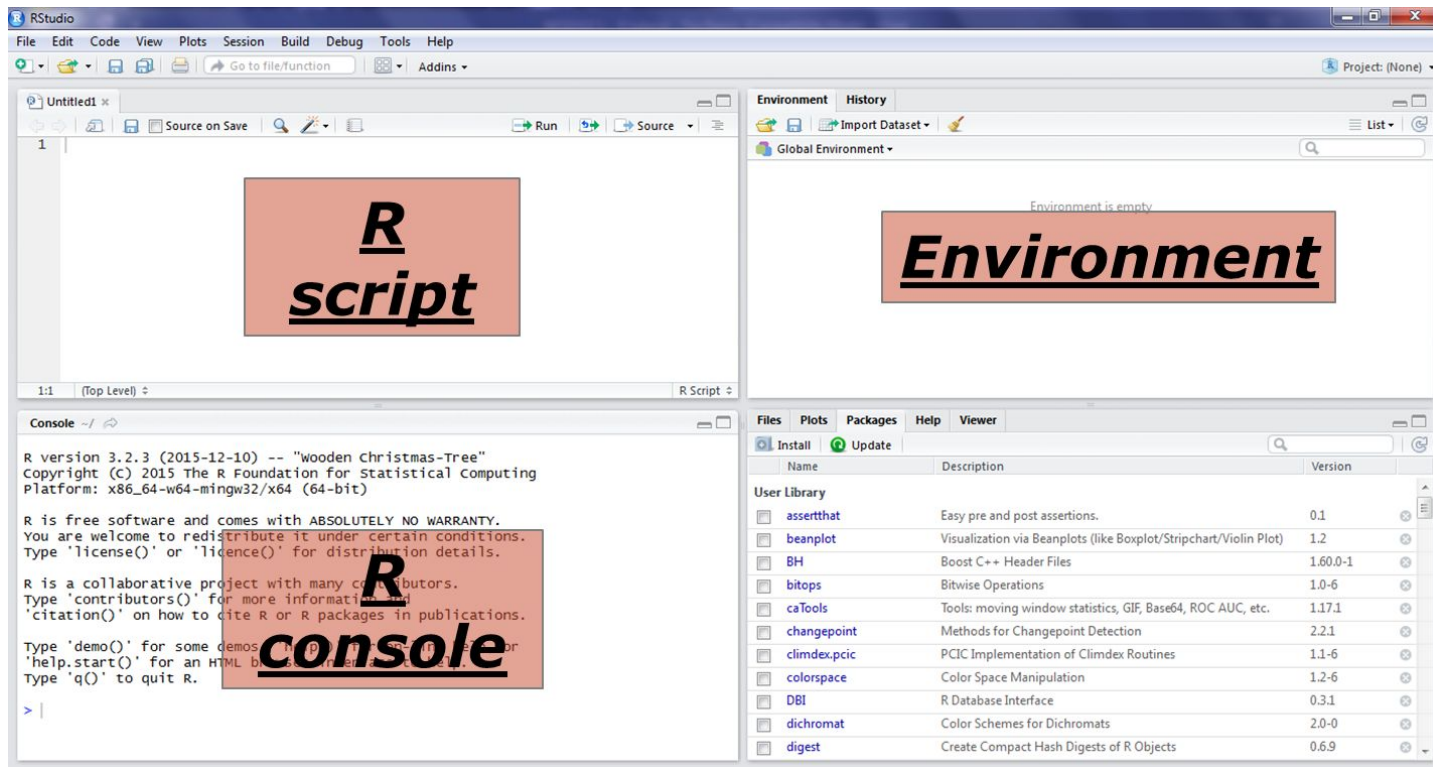


R console:

commands are
run and results
appear

> means 'ready to
work'

Open RStudio on your laptop now

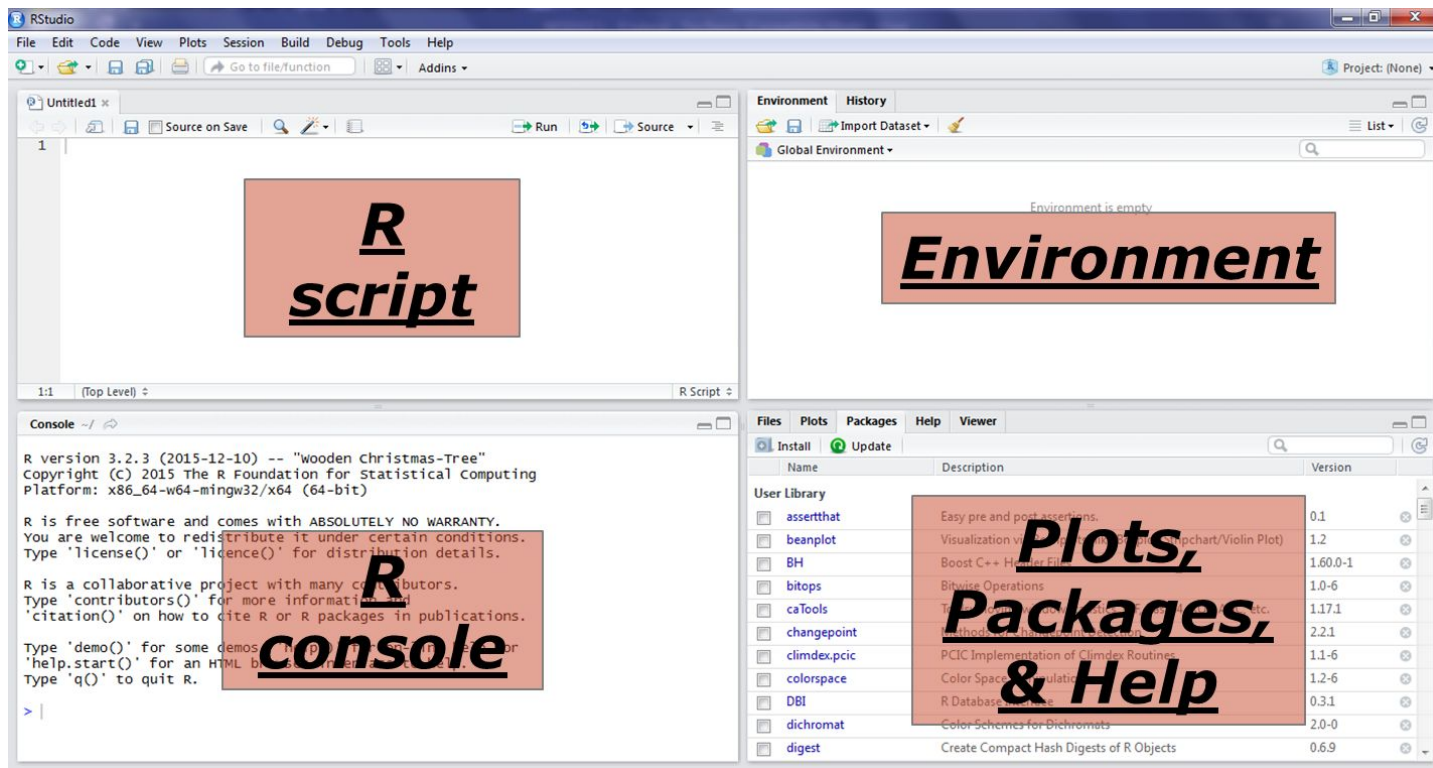


Environment:

see which objects were created

type of objects (double, integers, etc.), size and dimensions

Open RStudio on your laptop now



**Plots/Packages/
Help:**

plots will appear

see which
packages are
installed

help functions

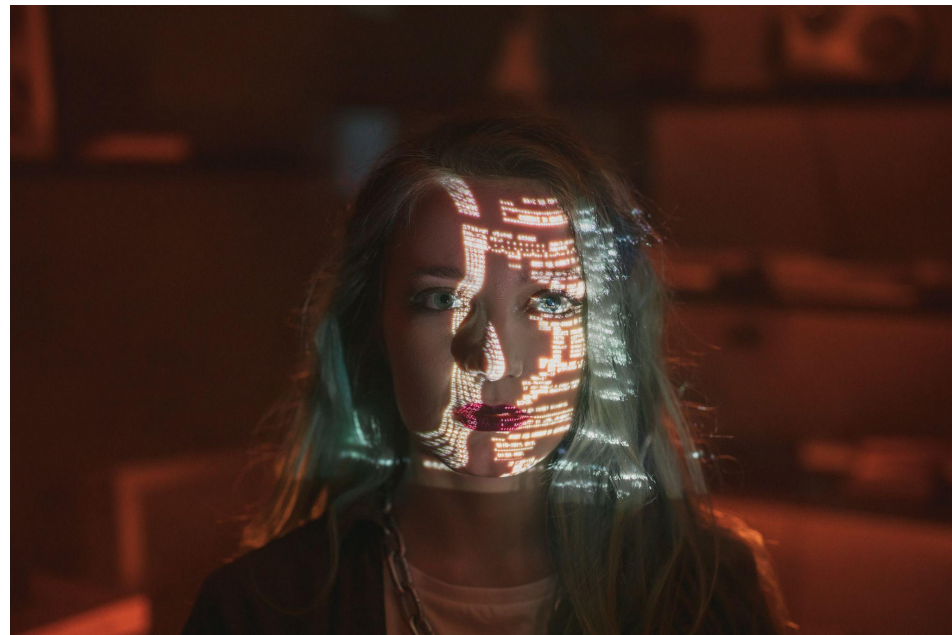
What are packages?

- bundles of tools and functions that others have developed to be used in R



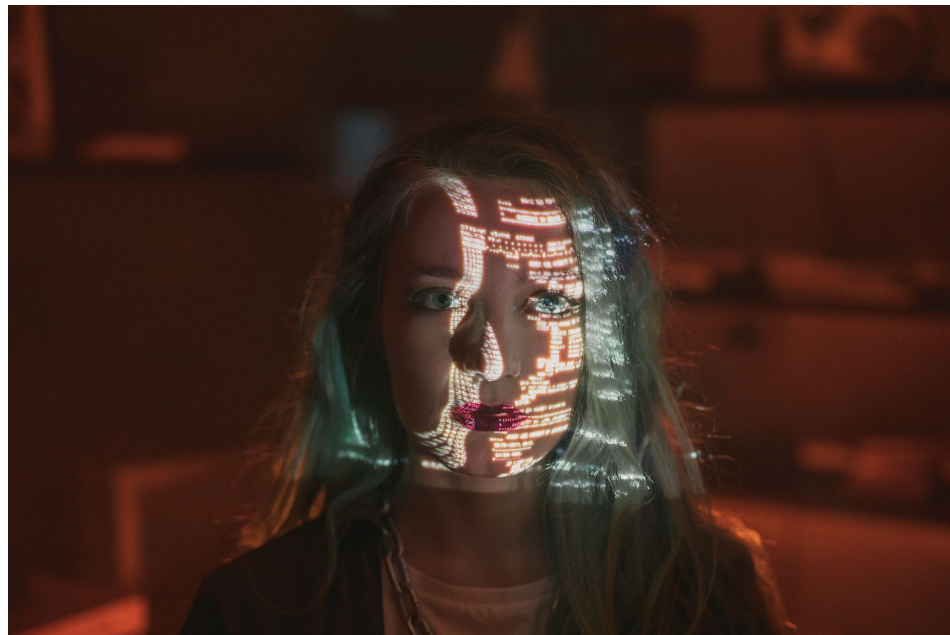
What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets



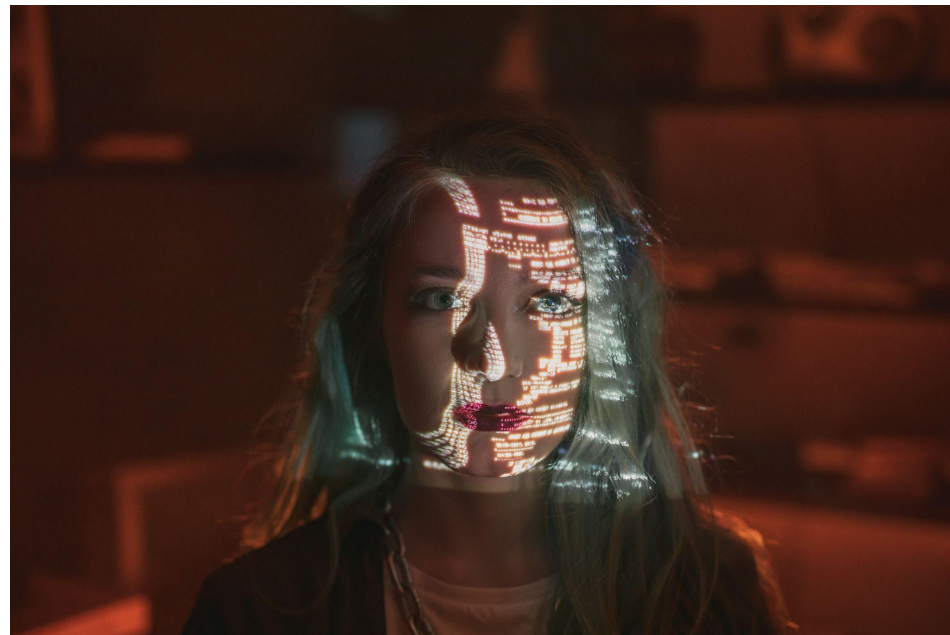
What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets
- you can also create your own packages for specific tasks!



What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets
- you can also create your own packages for specific tasks!
- currently, above 16,000 packages are available



Install one package

The screenshot shows the RStudio interface with the 'Tools' menu open, highlighting 'Install Packages...'. The console window at the bottom shows the following output:

```
> install.packages("rLakeAnalyzer")
Installing package into 'C:/Users/Rachel Pilla/documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/rLakeAnalyzer_1.7.6.zip'
Content type 'application/zip' length 516163 bytes (504 KB)
downloaded 504 KB

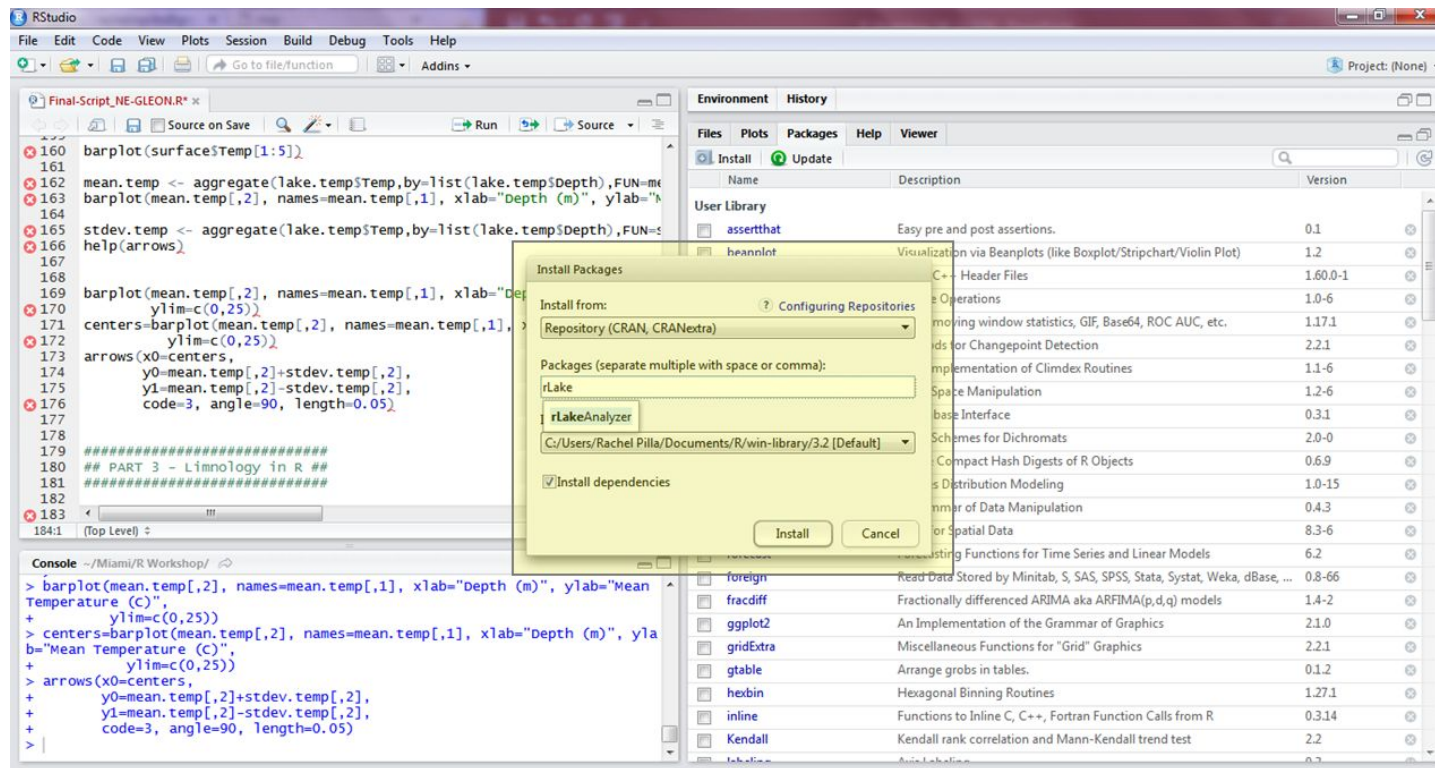
package 'rLakeAnalyzer' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\Rachel Pilla\AppData\Local\Temp\RtmpmmkDw7\downloaded_packages
> |
```

we need:

tidyverse

Install one package



The screenshot shows the RStudio interface with the following components:

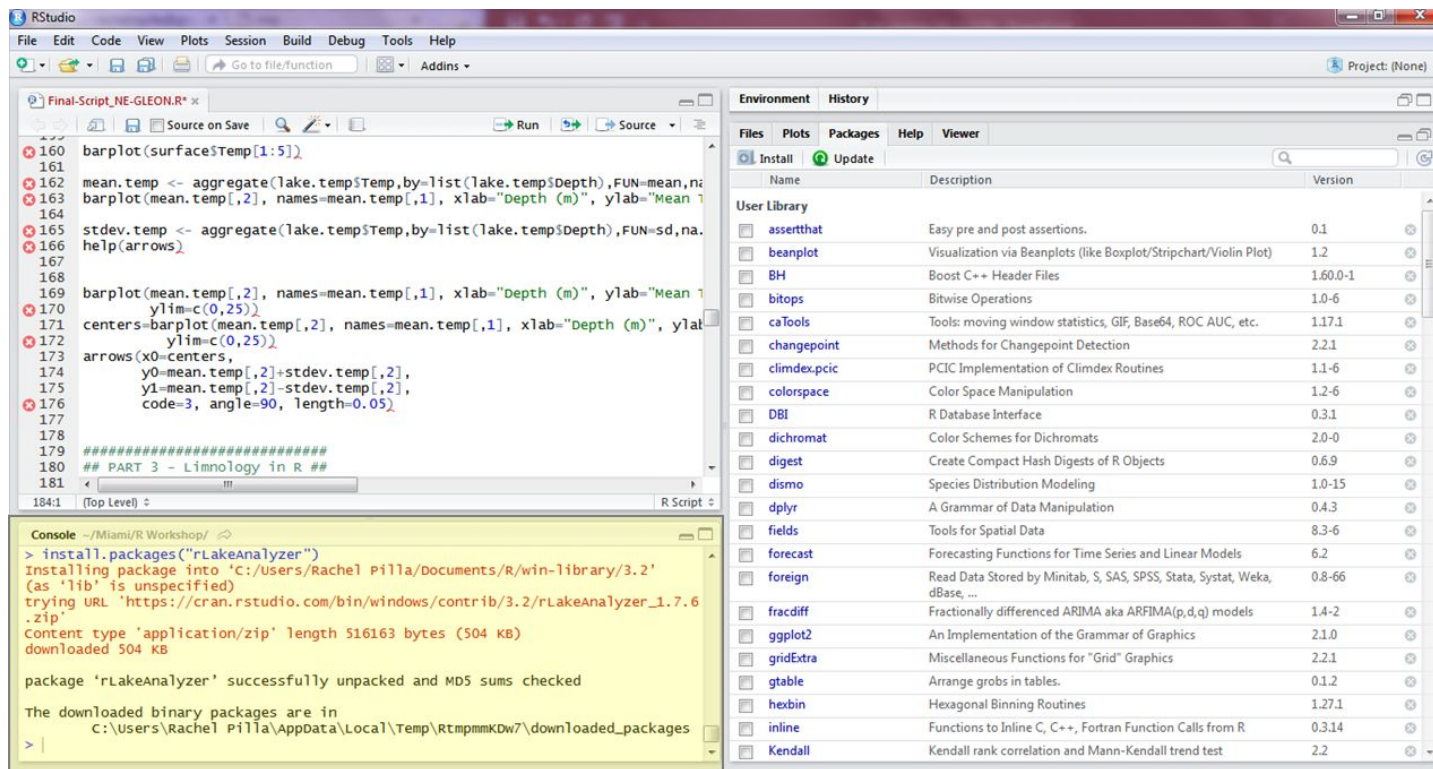
- Source Editor:** Contains R code for creating a barplot and arrows. The code is as follows:

```
barplot(surface$Temp[1:5])  
  
mean.temp <- aggregate(lake.temp$Temp, by=list(lake.temp$Depth), FUN=mean)  
barplot(mean.temp[,2], names=mean.temp[,1], xlab="Depth (m)", ylab="Mean Temperature (C)")  
  
stdev.temp <- aggregate(lake.temp$Temp, by=list(lake.temp$Depth), FUN=sd)  
help(arrows)  
  
barplot(mean.temp[,2], names=mean.temp[,1], xlab="Depth (m)", ylab="Mean Temperature (C)",  
        ylim=c(0,25))  
centers=barplot(mean.temp[,2], names=mean.temp[,1], xlab="Depth (m)", ylab="Mean Temperature (C)",  
               ylim=c(0,25))  
arrows(x0=centers[,1], y0=mean.temp[,2]+stdev.temp[,2],  
       x1=centers[,1], y1=mean.temp[,2]-stdev.temp[,2],  
       code=3, angle=90, length=0.05)
```
- Environment Pane:** Shows the User Library with various installed packages. The packages listed include: assertthat, beanint, C++ Header Files, C++ Operations, moving window statistics, GIF, Base64, ROC AUC, etc., vds for Changepoint Detection, implementation of Climdex Routines, Space Manipulation, base Interface, Schemes for Dichromats, Compact Hash Digests of R Objects, s Distribution Modeling, nmer of Data Manipulation, or Spatial Data, and Testing Functions for Time Series and Linear Models.
- Install Packages Dialog:** A dialog box is open, showing the repository (CRAN) and the package name (rLakeAnalyzer). The dialog also shows the path to the package (C:/Users/Rachel Pilla/Documents/R/win-library/3.2 [Default]) and the option to install dependencies.

we need:

tidyverse

Install one package



The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for plotting data. The code includes comments and function calls like `barplot`, `aggregate`, and `sd`.
- Environment/History:** Shows the installed packages and their versions.
- Console:** Displays the output of the `install.packages("rLakeAnalyzer")` command, showing the package being installed into the user's library and the successful unpacking of the package.

Environment/History Panel:

Name	Description	Version
assertthat	Easy pre and post assertions.	0.1
beanplot	Visualization via Beanplots (like Boxplot/Stripchart/Violin Plot)	1.2
BH	Boost C++ Header Files	1.60.0-1
bitops	Bitwise Operations	1.0-6
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
changepoint	Methods for Changepoint Detection	2.2.1
climdex.pcic	PCIC Implementation of Climdex Routines	1.1-6
colorspace	Color Space Manipulation	1.2-6
DBI	R Database Interface	0.3.1
dichromat	Color Schemes for Dichromats	2.0-0
digest	Create Compact Hash Digests of R Objects	0.6.9
dismo	Species Distribution Modeling	1.0-15
dplyr	A Grammar of Data Manipulation	0.4.3
fields	Tools for Spatial Data	8.3-6
forecast	Forecasting Functions for Time Series and Linear Models	6.2
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...	0.8-66
fracdiff	Fractionally differenced ARIMA aka ARFIMA(p,d,q) models	1.4-2
ggplot2	An Implementation of the Grammar of Graphics	2.1.0
gridExtra	Miscellaneous Functions for "Grid" Graphics	2.2.1
gtable	Arrange grobs in tables.	0.1.2
hexbin	Hexagonal Binning Routines	1.27.1
inline	Functions to Inline C, C++, Fortran Function Calls from R	0.3.14
Kendall	Kendall rank correlation and Mann-Kendall trend test	2.2

Console Output:

```
> install.packages("rLakeAnalyzer")
Installing package into 'C:/Users/Rachel Pilla/documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/rLakeAnalyzer_1.7.6.zip'
Content type 'application/zip' length 516163 bytes (504 KB)
downloaded 504 KB

package 'rLakeAnalyzer' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\Rachel Pilla\AppData\Local\Temp\RtmpmmKw7\downloaded_packages
> |
```

we need:

tidyverse

Installing a package

- packages only need to be installed ONCE
- but, if you want to use it you'll need to load it:

```
library(package name)
```



Hadley Wickham ✓
@hadleywickham

Replying to @ijlyttle

[@ijlyttle](#) a package is a like a book, a library is like a library; you use library() to check a package out of the library [#rsats](#)

8:34 AM · Dec 8, 2014 · [Echofon](#)

Package information

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for data manipulation using `gather` and `dplyr`.


```

197 # reformatting from wide to long using "gather"
198
199 TempGather <- temp %>%
200   gather(key = "Depth", value = "Temperature", SurfaceWaterTemp, Bot
201
202
203 # grouping data and reformatting data with dplyr
204
205 ?InsectSprays|
206 head(InsectSprays)
207 str(InsectSprays)
208
209 InsectSummary <- InsectSprays %>%
210   group_by(spray) %>%
211   summarize(MeanCount = mean(count),
212             StDevCount = sd(count))
213
214
215
216
217
218
219
220
221
222
      
```
- Console:** Shows the output of the executed code.


```

6      12      A
> str(InsectSprays)
'data.frame':   72 obs. of  2 variables:
 $ count: num   10  7 20 14 14 12 10 23 17 20 ...
 $ spray: Factor w/ 6 levels "A","B","C","D",...: 1 1 1 1 1 1 1 1 1 1 ...
>
> InsectSummary <- InsectSprays %>%
+   group_by(spray) %>%
+   summarize(MeanCount = mean(count),
+             StDevCount = sd(count))
>
      
```
- Environment/History/Connections:** Shows a list of installed R packages.

Name	Description	Version
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.2
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.66.0-1
bindr	Parametrized Active Bindings	0.1.1
bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2
broom	Convert Statistical Analysis Objects into Tidy Data Frames	0.4.4
callr	Call R from R	2.0.4
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
cli	Helpers for Developing Command Line Interfaces	1.0.0
colorspace	Color Space Manipulation	1.3-2
crayon	Colored Terminal Output	1.3.4
curl	A Modern and Flexible Web Client for R	3.2
DBI	R Database Interface	1.0.0
dbplyr	A 'dplyr' Back End for Databases	1.2.1
dichromat	Color Schemes for Dichromats	2.0-0
digest	Create Compact Hash Digests of R Objects	0.6.15
dplyr	A Grammar of Data Manipulation	0.7.5
evaluate	Parsing and Evaluation Tools that Provide More Details than the Default	0.10.1
forcats	Tools for Working with Categorical Variables (Factors)	0.3.0
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	2.2.1
glue	Interpreted String Literals	1.2.0
gridExtra	Miscellaneous Functions for "Grid" Graphics	2.3
gtable	Arrange 'Grob's' in Tables	0.2.0
haven	Import and Export 'SPSS', 'Stata' and 'SAS' Files	1.1.1
highr	Syntax Highlighting for R Source Code	0.7
hms	Pretty Time of Day	0.4.2
htmltools	Tools for HTML	0.3.6

under packages,
click on **dplyr**

Package information

The screenshot shows the RStudio interface with three main panes:

- Source Editor:** Contains R code for data manipulation using `dplyr`.


```

197 # reformatting from wide to long using "gather"
198
199 TempGather <- temp %>%
200   gather(key = "Depth", value = "Temperature", SurfaceWaterTemp, Bot
201
202
203 # grouping data and reformatting data with dplyr
204
205 ?InsectSprays|
206 head(InsectSprays)
207 str(InsectSprays)
208
209 InsectSummary <- InsectSprays %>%
210   group_by(spray) %>%
211   summarize(MeanCount = mean(count),
212             StDevCount = sd(count))
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
      
```
- Console:** Shows the output of the `str(InsectSprays)` command.


```

6 12 A
> str(InsectSprays)
'data.frame': 72 obs. of 2 variables:
 $ count: num 10 7 20 14 12 10 23 17 20 ...
 $ spray: Factor w/ 6 levels "A","B","C","D",...: 1 1 1 1 1 1 1 1 1 ...
>
> InsectSummary <- InsectSprays %>%
+   group_by(spray) %>%
+   summarize(MeanCount = mean(count),
+             StDevCount = sd(count))
>
      
```
- Help Viewer:** Displays the documentation for the `dplyr` package version 0.7.5.

A Grammar of Data Manipulation

Documentation for package 'dplyr' version 0.7.5

 - [DESCRIPTION file](#)
 - [User guides, package vignettes and other documentation](#)

Help Pages

[dplyr-package](#) dplyr: a grammar of data manipulation

-- A --

add_count	Count/tally observations by group
add_tally	Count/tally observations by group
all_equal.tbl_df	Flexible equality comparison for data frames
all_equal	Flexible equality comparison for data frames
all_vars	Apply predicate to all variables

this lists all
available
functions

click on any
function to see
the help file

Coding in R

- write commands in the script (upper left panel)
 - save it, edit it, revisit it later, etc.



Coding in R

- write commands in the script (upper left panel)
 - save it, edit it, revisit it later, etc.
- code NOT automatically run when you hit Enter



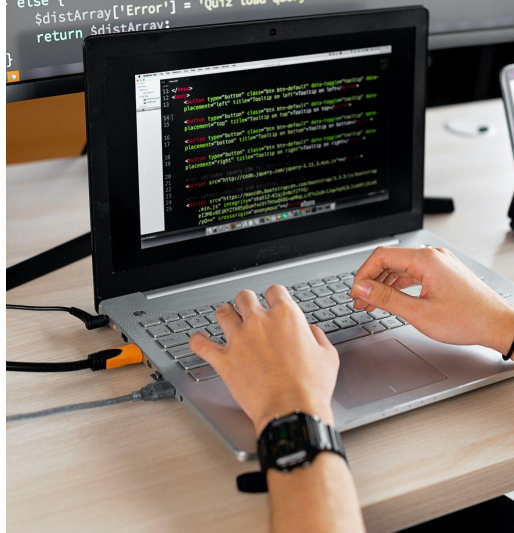
Coding in R

- write commands in the script (upper left panel)
 - save it, edit it, revisit it later, etc.
- code NOT automatically run when you hit Enter
- to run it:
 - “Run” button in upper right corner
 - “CTRL + Enter” (Windows)
 - “Command + Enter” (Mac)



Coding in R

```
if(result != false) {  
  $distArray = array();  
  $row = mysqli_fetch_assoc($result);  
  $correctAnswer = $row['Correct'];  
  $distArray['A'] = $row['Anum'];  
  $distArray['B'] = $row['Bnum'];  
  $distArray['C'] = $row['Cnum'];  
  $distArray['D'] = $row['Dnum'];  
  $distArray['Correct'] = $correctAnswer;  
  $distArray['Answer'] = rtrim($row['CorrectAnswer'], ".");  
  $distArray['Query'] = "SELECT * FROM TechTerms WHERE Date='date'";  
  return $distArray;  
}  
else {  
  $distArray['Error'] = 'Quiz load query failed';  
  return $distArray;  
}
```



- add comments using #
 - additional information
 - will be ignored

Coding in R



- add comments using #
 - additional information
 - will be ignored
- R is case sensitive
 - “Mean” ≠ “mean”

Coding in R



- add comments using #
 - additional information
 - will be ignored
- R is case sensitive
 - “Mean” ≠ “mean”
- R doesn't care about spaces and tabs

Coding in R



- add comments using #
 - additional information
 - will be ignored
- R is case sensitive
 - “Mean” ≠ “mean”
- R doesn't care about spaces and tabs
- need to close all parentheses and quotations

Key components

- **functions:** allow you to manipulate data, apply calculations, run statistical analysis, much more!

Key Components of Code

```
x <- seq(1,10)
```

name of the **function**
to create a **sequence**

Key components

- **functions:** allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments:** defining information for functions, “customize” it

Key Components of Code

```
x <- seq(1, 10)
```

the **arguments** to
define the function, to
create a sequence
from 1 through 10

Key components

- **functions:** allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments:** defining information for functions, “customize” it
- **objects:** pieces of data saved in R, can be called up, reused and manipulated

Key Components of Code

```
x <- seq(1,10)
```

name of the **object**
that saves the results
of the function in R

Key components

- **functions:** allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments:** defining information for functions, “customize” it
- **objects:** pieces of data saved in R, can be called up, reused and manipulated

Key Components of Code

```
x <- seq(1,10)
```

assignment operator
tells R to save the result
of the function as the
named object

Types of data

- **objects (data types):**
 - numeric (2.334, 3.14159)
 - integer (2, 2405, 54)
 - characters ("hello world")
 - logical (TRUE/FALSE)
 - complex (1+4i)
- **data structures:**
 - vector (1-D object with same data type)
 - matrix (2-D object with same data type)
 - data frame (2-D object with different data types per column if needed, very useful!)

Live Coding time!

