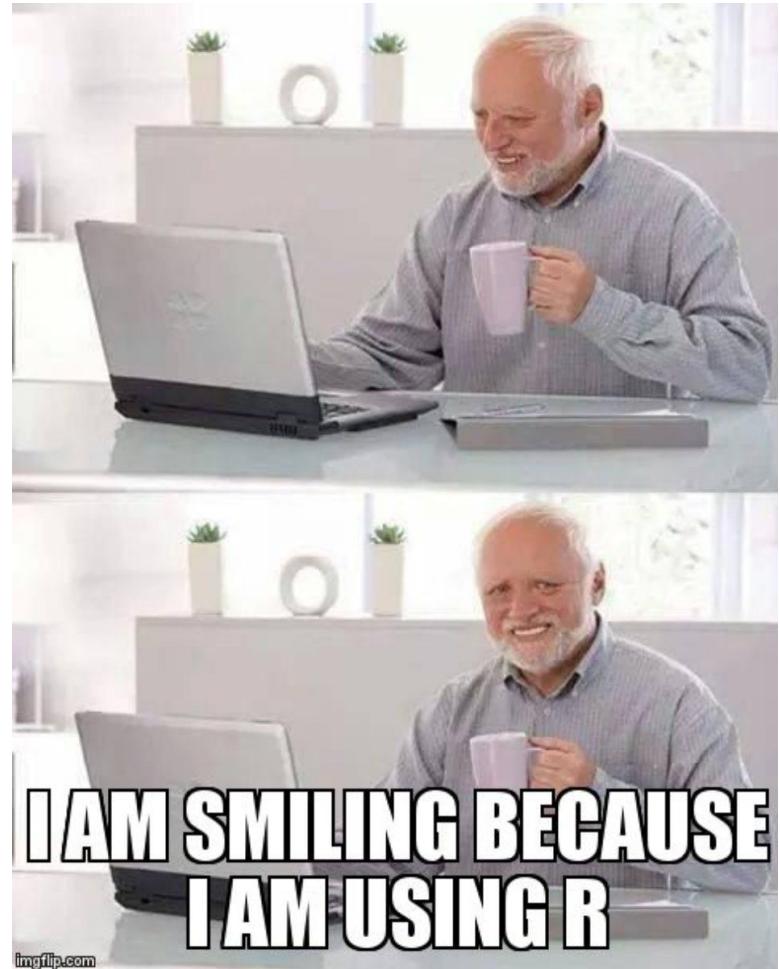


# R ecosystem

Lecture 5  
BIOS 6660, Spring 2019  
Instructor: Pam Russell



**Homework 2: solution posted later today**

Questions on homework 2?

# Homework 3: intermediate R

- Available today, due next Tuesday
- R Markdown (today)
- dplyr and ggplot2 (Thursday)

# The R ecosystem

- User community
- Packages
- Package repositories
- Reports with R Markdown
- ~~Rstudio~~

# R community



<https://www.r-bloggers.com/>



<https://stackoverflow.com/questions/tagged/r>

## Denver R User Group

★★★★★ 193 ratings

Denver, CO

1,349 members · Public group

<https://www.meetup.com/DenverRUG>

# R packages: standing on the shoulders of giants

Collection of functions and/or data developed by members of the community

We can import these into our code and use them



# Thousands of packages



# Where do packages come from?



CRAN:  
Comprehensive R  
Archive Network



# Are all repositories equivalent?



CRAN:  
Comprehensive R  
Archive Network

Requirements promote  
consistent quality and user  
experience



Stricter guidelines than CRAN.  
Interoperability with other  
BioConductor packages highly  
encouraged.



Often used to develop  
packages also on CRAN or  
Bioconductor. No rules for  
GitHub.

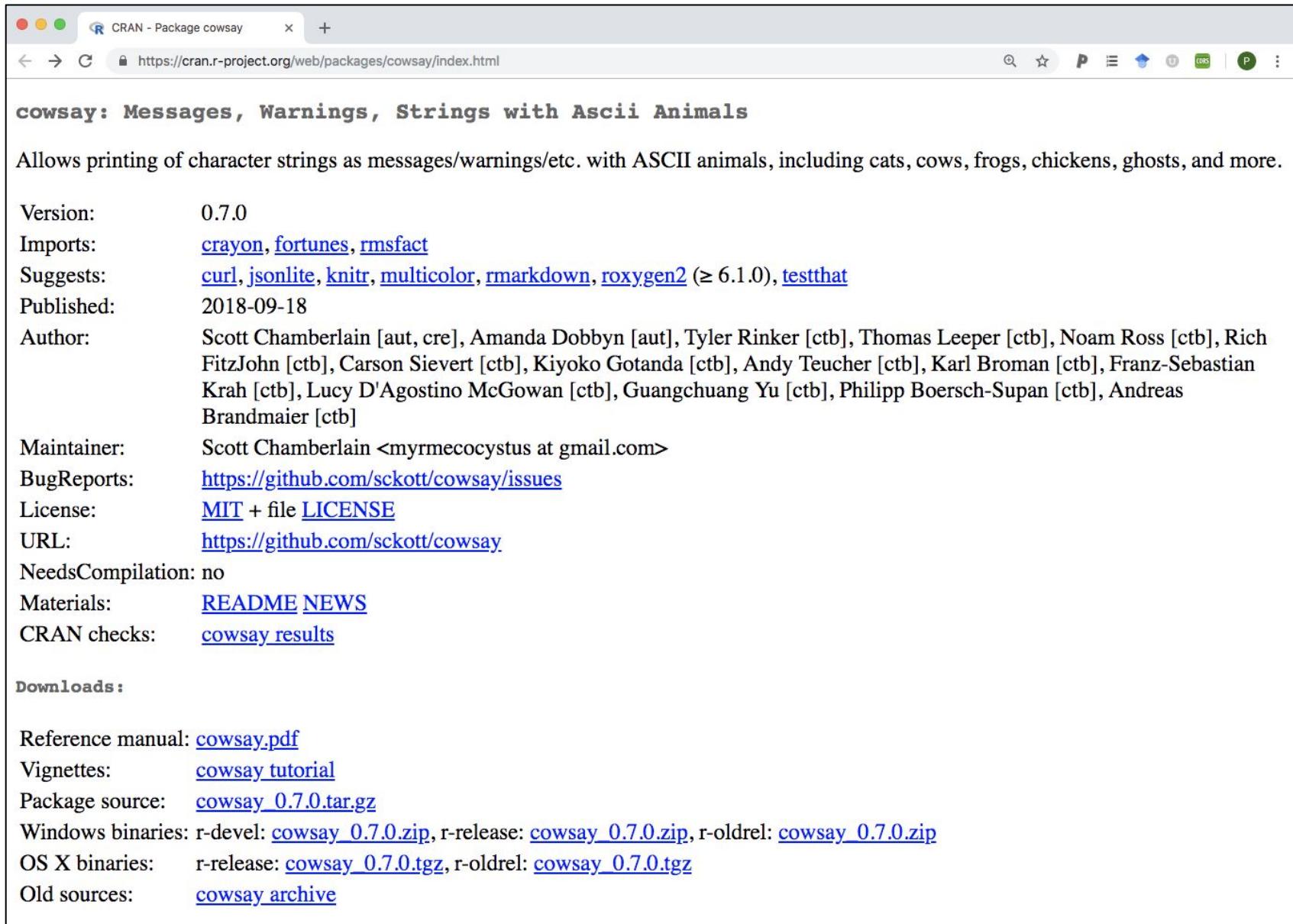
# CRAN

Network of servers that store >13,000 packages

Main repository for subject areas not covered by Bioconductor



# Package pages on CRAN



The screenshot shows a web browser window with the title "CRAN - Package cowsay". The URL in the address bar is <https://cran.r-project.org/web/packages/cowsay/index.html>. The page content is as follows:

**cowsay: Messages, Warnings, Strings with Ascii Animals**

Allows printing of character strings as messages/warnings/etc. with ASCII animals, including cats, cows, frogs, chickens, ghosts, and more.

**Version:** 0.7.0

**Imports:** [crayon](#), [fortunes](#), [rmsfact](#)

**Suggests:** [curl](#), [jsonlite](#), [knitr](#), [multicolor](#), [rmarkdown](#), [roxygen2](#) (≥ 6.1.0), [testthat](#)

**Published:** 2018-09-18

**Author:** Scott Chamberlain [aut, cre], Amanda Dobbyn [aut], Tyler Rinker [ctb], Thomas Leeper [ctb], Noam Ross [ctb], Rich FitzJohn [ctb], Carson Sievert [ctb], Kiyoko Gotanda [ctb], Andy Teucher [ctb], Karl Broman [ctb], Franz-Sebastian Krah [ctb], Lucy D'Agostino McGowan [ctb], Guangchuang Yu [ctb], Philipp Boersch-Supan [ctb], Andreas Brandmaier [ctb]

**Maintainer:** Scott Chamberlain <myrmecocystus at gmail.com>

**BugReports:** <https://github.com/sckott/cowsay/issues>

**License:** [MIT](#) + file [LICENSE](#)

**URL:** <https://github.com/sckott/cowsay>

**NeedsCompilation:** no

**Materials:** [README](#) [NEWS](#)

**CRAN checks:** [cowsay results](#)

**Downloads:**

Reference manual: [cowsay.pdf](#)

Vignettes: [cowsay tutorial](#)

Package source: [cowsay\\_0.7.0.tar.gz](#)

Windows binaries: r-devel: [cowsay\\_0.7.0.zip](#), r-release: [cowsay\\_0.7.0.zip](#), r-oldrel: [cowsay\\_0.7.0.zip](#)

OS X binaries: r-release: [cowsay\\_0.7.0.tgz](#), r-oldrel: [cowsay\\_0.7.0.tgz](#)

Old sources: [cowsay archive](#)

# Some CRAN subject areas

[Bayesian](#)  
[ChemPhys](#)  
[ClinicalTrials](#)  
[Cluster](#)  
[Databases](#)  
[DifferentialEquations](#)  
[Distributions](#)  
[Econometrics](#)  
[Environmetrics](#)  
[ExperimentalDesign](#)  
[ExtremeValue](#)  
[Finance](#)  
[FunctionalData](#)  
[Genetics](#)  
[Graphics](#)  
[HighPerformanceComputing](#)  
[Hydrology](#)  
[MachineLearning](#)  
[MedicalImaging](#)  
[MetaAnalysis](#)  
[MissingData](#)

Bayesian Inference  
Chemometrics and Computational Physics  
Clinical Trial Design, Monitoring, and Analysis  
Cluster Analysis & Finite Mixture Models  
Databases with R  
Differential Equations  
Probability Distributions  
Econometrics  
Analysis of Ecological and Environmental Data  
Design of Experiments (DoE) & Analysis  
Extreme Value Analysis  
Empirical Finance  
Functional Data Analysis  
Statistical Genetics  
Graphic Displays & Dynamic Graphics & Interfaces  
High-Performance and Parallel Computing  
Hydrological Data and Modeling  
Machine Learning & Statistical Learning  
Medical Image Analysis  
Meta-Analysis  
Missing Data

[ModelDeployment](#)  
[Multivariate](#)  
[NaturalLanguageProcessing](#)  
[NumericalMathematics](#)  
[OfficialStatistics](#)  
[Optimization](#)  
[Pharmacokinetics](#)  
[Phylogenetics](#)  
[Psychometrics](#)  
[ReproducibleResearch](#)  
[Robust](#)  
[SocialSciences](#)  
[Spatial](#)  
[SpatioTemporal](#)  
[Survival](#)  
[TimeSeries](#)  
[WebTechnologies](#)  
[gR](#)

Model Deployment with R  
Multivariate Statistics  
Natural Language Processing  
Numerical Mathematics  
Official Statistics & Survey Methodology  
Optimization and Mathematical Programming  
Analysis of Pharmacokinetic Data  
Phylogenetics, Especially Comparative Methods  
Psychometric Models and Methods  
Reproducible Research  
Robust Statistical Methods  
Statistics for the Social Sciences  
Analysis of Spatial Data  
Handling and Analyzing Spatio-Temporal Data  
Survival Analysis  
Time Series Analysis  
Web Technologies and Services  
Graphical Models in R

# Bioconductor

Analysis tools for high-throughput genomic data

>1,600 packages



# Package pages on Bioconductor

Bioconductor - DESeq2

<https://www.bioconductor.org/packages/release/bioc/html/DESeq2.html>

## DESeq2

platforms all rank 26 / 1561 posts 297 / 1 / 2 / 44 in bioc 5.5 years  
build ok updated before release

DOI: [10.18129/B9.bioc.DESeq2](https://doi.org/10.18129/B9.bioc.DESeq2) [f](#) [t](#)

Differential gene expression analysis based on the negative binomial distribution

Bioconductor version: Release (3.7)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love, Simon Anders, Wolfgang Huber

Maintainer: Michael Love <[michaelisaiahlove@gmail.com](mailto:michaelisaiahlove@gmail.com)>

Citation (from within R, enter `citation("DESeq2")`):

Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, **15**, 550. doi: [10.1186/s13059-014-0550-8](https://doi.org/10.1186/s13059-014-0550-8).

### Installation

To install this package, start R and enter:

```
## try http:// if https:// URLs are not supported
source("https://bioconductor.org/biocLite.R")
biocLite("DESeq2")
```

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("DESeq2")
```

[HTML](#) [R Script](#) Analyzing RNA-seq data with DESeq2  
[PDF](#) [Reference Manual](#)  
[NEWS](#)

**Documentation »**

**Bioconductor**

- Package [vignettes](#) and manuals.
- [Workflows](#) for learning and use.
- [Course and conference](#) material.
- [Videos](#).
- Community [resources](#) and [tutorials](#).

R / [CRAN](#) packages and [documentation](#)

**Support »**

Please read the [posting guide](#). Post questions about Bioconductor to one of the following locations:

- [Support site](#) - for questions about Bioconductor packages
- [Bioc-devel](#) mailing list - for package developers

# Some Bioconductor subject areas

## Bioconductor version 3.8 (Release)

Autocomplete biocViews search:

### ▼ Software (1649)

- ▶ AssayDomain (661)
- ▼ BiologicalQuestion (668)
  - AlternativeSplicing (33)
  - Coverage (99)
  - DifferentialExpression (284)
  - DifferentialMethylation (36)
  - DifferentialPeakCalling (12)
  - DifferentialSplicing (26)
  - DNA3DStructure (4)
  - DriverMutation (1)
  - FunctionalPrediction (14)
  - GenePrediction (13)
  - GeneRegulation (79)
  - GeneSetEnrichment (106)
  - GeneSignaling (5)
- ...

## Bioconductor version 3.8 (Release)

Autocomplete biocViews search:

### ▼ Software (1649)

- ▶ AssayDomain (661)
- ▶ BiologicalQuestion (668)
- ▶ Infrastructure (360)
- ▼ ResearchField (728)
  - BiomedicalInformatics (42)
  - CellBiology (46)
  - Cheminformatics (11)
  - ComparativeGenomics (5)
  - Epigenetics (42)
  - FunctionalGenomics (37)
  - Genetics (178)
  - ImmunoOncology (450)
  - Lipidomics (7)
  - MathematicalBiology (5)
  - Metabolomics (51)
  - ...

# Bioconductor package: genomic file formats

## Rsamtools

platforms all

rank 18 / 1649

posts 10 / 0.7 / 2 / 2

in Bioc 9 years

build warnings

updated before release

DOI: [10.18129/B9.bioc.Rsamtools](https://doi.org/10.18129/B9.bioc.Rsamtools)



### Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import

Bioconductor version: Release (3.8)

This package provides an interface to the 'samtools', 'bcftools', and 'tabix' utilities (see 'LICENCE') for manipulating SAM (Sequence Alignment / Map), FASTA, binary variant call (BCF) and compressed indexed tab-delimited (tabix) files.

# Bioconductor package: database access

The screenshot shows the Ensembl BioMart interface. On the left, there's a sidebar with buttons for 'New', 'Count', and 'Results'. The main area has tabs for 'Dataset', 'Filters', and 'Attributes'. Under 'Dataset', it says 'Human genes (GRCh38.p12)'. Under 'Filters', it says '[None selected]'. Under 'Attributes', it lists 'Gene stable ID' and 'Transcript stable ID'. To the right, a dropdown menu titled '- CHOOSE DATASET -' shows several options: 'Chicken genes (GRCg6a)', 'Human genes (GRCh38.p12)' (which is selected and highlighted in blue), 'Mouse genes (GRCm38.p6)', 'Rat genes (Rnor\_6.0)', and 'Zebrafish genes (GRCz11)'. Below this is a dashed line, followed by a list of other datasets: 'Agassiz's desert tortoise genes (ASM289641v1)', 'Algerian mouse genes (SPRET\_EiJ\_v1)', 'Alpaca genes (vicPac1)', and 'Amazon molly genes (Poecilia\_formosa-5.1.2)'. At the bottom of the interface, there's a 'biomaRt' section with various status indicators and social media links.

Dataset

Human genes (GRCh38.p12)

Filters

[None selected]

Attributes

Gene stable ID  
Transcript stable ID

- CHOOSE DATASET -

Chicken genes (GRCg6a)  
Human genes (GRCh38.p12)  
Mouse genes (GRCm38.p6)  
Rat genes (Rnor\_6.0)  
Zebrafish genes (GRCz11)

Agassiz's desert tortoise genes (ASM289641v1)  
Algerian mouse genes (SPRET\_EiJ\_v1)  
Alpaca genes (vicPac1)  
Amazon molly genes (Poecilia\_formosa-5.1.2)

biomaRt

platforms all rank 19 / 1649 posts 28 / 1 / 2 / 7 in Bioc > 13.5 years

build ok updated before release

DOI: [10.18129/B9.bioc.biomaRt](https://doi.org/10.18129/B9.bioc.biomaRt) [f](#) [t](#)

Interface to BioMart databases (i.e. Ensembl)

<https://www.ensembl.org/biomart>

# Bioconductor packages: differential gene expression

## edgeR

platforms all rank 23 / 1649 posts 88 / 1 / 3 / 20 in Bioc 10.5 years

build ok updated < 1 month

DOI: [10.18129/B9.bioc.edgeR](https://doi.org/10.18129/B9.bioc.edgeR)  

Empirical Analysis of Digital Gene Expression Data in R

## DESeq2

platforms all rank 26 / 1649 posts 326 / 1 / 2 / 37 in Bioc 6 years

build ok updated < 1 month

DOI: [10.18129/B9.bioc.DESeq2](https://doi.org/10.18129/B9.bioc.DESeq2)  

Differential gene expression analysis based on the negative binomial distribution

# Bioconductor package: ChIP-seq

## ChIPseeker

platforms all

rank 112 / 1649

posts 8 / 0.4 / 0.8 / 2

in Bioc 5 years

build warnings

updated before release

DOI: [10.18129/B9.bioc.ChIPseeker](https://doi.org/10.18129/B9.bioc.ChIPseeker)



## ChIPseeker for ChIP peak Annotation, Comparison, and Visualization

Bioconductor version: Release (3.8)

This package implements functions to retrieve the nearest genes around the peak, annotate genomic region of the peak, statistical methods for estimate the significance of overlap among ChIP peak data sets, and incorporate GEO database for user to compare the own dataset with those deposited in database. The comparison can be used to infer cooperative regulation and thus can be used to generate hypotheses. Several visualization functions are implemented to summarize the coverage of the peak experiment, average profile and heatmap of peaks binding to TSS regions, genomic annotation, distance to TSS, and overlap of peaks or genes.

# Killer feature of Bioconductor: interoperability

## Common Bioconductor Methods and Classes

We strongly recommend reusing existing methods for importing data, and reusing established classes for representing data. Here are some suggestions for importing different file types and commonly used *Bioconductor* classes. For more classes and functionality also try searching in [BioViews](#) for your data type.

### Importing

- GTF, GFF, BED, BigWig, etc., – [rtracklayer::import\(\)](#)
- VCF – [VariantAnnotation::readVcf\(\)](#)
- SAM / BAM – [Rsamtools::scanBam\(\)](#), [GenomicAlignments::readGAlignment\\*\(\)](#)
- FASTA – [Biostrings::readDNAStringSet\(\)](#)
- FASTQ – [ShortRead::readFastq\(\)](#)
- MS data (XML-based and mfg formats) – [MSnbase::readMSData\(\)](#), [MSnbase::readMgfData\(\)](#)

### Common Classes

- Rectangular feature x sample data – [SummarizedExperiment::SummarizedExperiment\(\)](#) (RNAseq count matrix, microarray, ...)
- Genomic coordinates – [GenomicRanges::GRanges\(\)](#) (1-based, closed interval)
- DNA / RNA / AA sequences – [Biostrings::\\*StringSet\(\)](#)
- Gene sets – [GSEABase::GeneSet\(\)](#) [GSEABase::GeneSetCollection\(\)](#)
- Multi-omics data – [MultiAssayExperiment::MultiAssayExperiment\(\)](#)
- Single cell data – [SingleCellExperiment::SingleCellExperiment\(\)](#)
- Mass spec data – [MSnbase::MSnExp\(\)](#)

# GitHub

Often used for development versions of packages also on CRAN or Bioconductor

Post issues; get bleeding edge version

Generally, beware of packages published only on GitHub



# Should I trust public code on GitHub?

Issues indicate active user community

Pull requests indicate active community contribution

Measures of community engagement

The screenshot shows the GitHub repository page for `broadinstitute/gatk`. The top navigation bar includes links for Code, Issues (922), Pull requests (52), Projects (3), Wiki, and Insights. To the right are buttons for Watch (133), Star (557), Fork (218), and a clone URL. Below the navigation is a summary bar with metrics: 3,735 commits, 464 branches, 34 releases, 80 contributors, and a BSD-3-Clause license. A dropdown menu shows the current branch is master, and there is a 'New pull request' button. The main area displays a list of recent commits from various contributors, with details like commit hash, author, message, date, and file changes. A prominent green 'Clone or download' button is at the bottom right.

Commit	Author	Message	Date
64e2d3a	Ibergelson and droazen	A port of FastaAlternateReferenceMaker and FastaReferenceMaker from G...	25 minutes ago
4796		Add slightly modified version of GATK3 github issue template (#4796)	8 months ago
		Improve proposed tumor in normal docs to account for new Mutect2 opti...	26 days ago

Extensive commit history

Releases indicate active development and professionalism

Large developer team

Open source license

Recently modified

# Should I trust public code on GitHub?

Badges indicate professionalism; build is passing

Test coverage not great, but they care enough to track it



Screenshot of a GitHub README.md page. At the top left is a file icon followed by the text "README.md". Below the file icon are two purple arrows pointing from the text "Badges indicate professionalism; build is passing" and "Test coverage not great, but they care enough to track it" towards the badge area. The badge area contains four items: "build passing", "codecov 13%", "maven-central v4.0.12.0", and "license BSD 3-Clause".

Please see the [GATK website](#), where you can download a precompiled executable, read documentation, ask questions, and receive technical support.

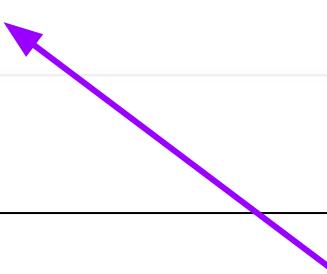
## GATK 4

This repository contains the next generation of the Genome Analysis Toolkit (GATK). The contents of this repository are 100% open source and released under the BSD 3-Clause license (see [LICENSE.TXT](#)).

GATK4 aims to bring together well-established tools from the [GATK](#) and [Picard](#) codebases under a streamlined framework, and to enable selected tools to be run in a massively parallel way on local clusters or in the cloud using [Apache Spark](#). It also contains many newly developed tools not present in earlier releases of the toolkit.

## Table of Contents

- [Requirements](#)
- [Quick Start Guide](#)



Good documentation

# Installing packages

CRAN	<code>install.packages("packagename")</code>
Bioconductor	<code>source("https://bioconductor.org/biocLite.R") biocLite("packagename")</code>
GitHub	<code>install.packages("devtools") devtools::install_github("username/packagename")</code>

# Using package functions

If you will just use package functions a few times, or to eliminate name ambiguity

```
packagename::function1(args)  
packagename::function2(args)
```

If you will use the package a lot, attach the package.

```
library(packagename)  
function1(args)  
function2(args)
```

# Name conflicts

You might load two packages that define functions with the same name

Or, a package might define a function also in base R

Or, you might define a function in your script that masks a package function

Mysterious errors when you are accidentally calling the wrong version of the function

# R looks for function definitions on the search path

R first looks for functions in global environment  
(functions you have defined)

Then, search path is opposite order of  
`library()` statements

In other words, function names replace  
previously loaded function names

# Resolving conflicts

To see potential conflicts on your search path:  
`conflicts()`

To avoid conflicts, use `::`  
e.g. `packagename::function1(args)`

# Package documentation: view help files in Rstudio

> `help(package = cowsay)`

The screenshot shows the RStudio interface with the 'Help' tab selected. The main pane displays the documentation for the 'cowsay' package, version 0.7.0. The title is 'Messages, Warnings, Strings with Ascii Animals'. Below the title is the R logo. The text reads: 'Documentation for package 'cowsay' version 0.7.0'. A bulleted list follows: • DESCRIPTION file. • User guides, package vignettes and other documentation. Under 'Help Pages', there is a table:

Link	Description
<a href="#">cowsay-package</a>	Sling messages and warnings with flair
<a href="#">animals</a>	Animals
<a href="#">cowsay</a>	Sling messages and warnings with flair
<a href="#">endless_horse</a>	Endless horse
<a href="#">say</a>	Sling messages and warnings with flair

The screenshot shows the R Documentation page for the 'say' function. The title is 'say {cowsay}'. The description is 'Sling messages and warnings with flair'. The usage is shown as a code snippet: `say(what = "Hello world!", by = "cat", type = "message", what_color = NULL, by_color = NULL, length = 18, fortune = NULL, ...)`. The arguments section details:

Argument	Description
what	(character) What do you want to say? See details.
by	(character) Type of thing, one of cow, chicken, poop, cat, facecat, bigcat, longcat, shortcat, behindcat, longtailcat, anxiouscat, grumpycat, smallcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, signbunny, stretchycat, fish, trilobite, shark, buffalo, clippy, mushroom, monkey, egret, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use <a href="#">match.arg</a> internally, so you can use unique parts of words that don't conflict with others, like "g" for "ghost" because there's no other animal that starts with "g".
type	(character) One of message (default), warning, or string (returns string). If multiple colors are supplied to what_color or by_color, type cannot be warning. (This is a limitation of the <a href="#">multicolor</a> package ./.)

# Package documentation: PDF reference manual

Linked from CRAN or Bioconductor page

## Package ‘cowsay’

September 18, 2018

**Title** Messages, Warnings, Strings with Ascii Animals

**Description** Allows printing of character strings as messages/warnings/etc. with ASCII animals, including cats, cows, frogs, chickens, ghosts, and more.

**Version** 0.7.0

**License** MIT + file LICENSE

**URL** <https://github.com/sckott/cowsay>

**BugReports** <https://github.com/sckott/cowsay/issues>

**LazyData** yes

**Encoding** UTF-8

**VignetteBuilder** knitr

**Imports** crayon, fortunes, rmsfact

**Suggests** curl, jsonlite, knitr, multicolor, rmarkdown, roxygen2 (>= 6.1.0), testthat

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Author** Scott Chamberlain [aut, cre],

Amanda Dobbyn [aut],

Tyler Rinker [ctb],

Thomas Leeper [ctb],

Noam Ross [ctb],

Rich FitzJohn [ctb],

Carson Sievert [ctb],

Kiyoko Gotanda [ctb],

Andy Teucher [ctb],

Karl Broman [ctb],

Franz-Sebastian Krah [ctb],

Lucy D'Agostino McGowan [ctb],

Guangchuang Yu [ctb],

Philipp Boersch-Supan [ctb],

Andreas Brandmaier [ctb]

## R topics documented:

animals	.....	2
cowsay	.....	3
endless_horse	.....	4
say	.....	4

## Index

animals                  *Animals*

### Description

Named vector of animals

### Usage

animals

### Format

An object of class character of length 41.

### Details

animals is a named character vector of animals, with each element a character string of variable length specifying an ASCII animal. Note that some have unicode characters that won't play well on some operating systems.

### Examples

```
animals['cow']
animals['clippy']
animals[['clippy']]
```

# Package documentation: vignettes

Linked from CRAN or Bioconductor page

Or from R command prompt with `browseVignettes("cowsay")`

## Introduction to cowsay

*Scott Chamberlain, Tyler Rinker, Thomas Leeper, Noam Ross, Rich FitzJohn, Kiyoko Gotanda, Carson Sievert, Andy Teucher, Karl Broman, Franz-Sebastian Krah, Lucy D'Agostino McGowan, Guangchuang Yu, Paolo Sonego, and Philipp Boersch-Supan*

**2018-09-17**

cowsay makes it easy to print messages, warnings, or character strings with various animals and other creatures.

## Installation

Stable version from CRAN

```
install.packages("cowsay")
```

or dev version from GitHub

```
install.packages("devtools")
devtools::install_github("sckott/cowsay")
```

```
library(cowsay)
```

## The animals

The animals, and other ascii creatures, are all in a single named character vector that is exported from the package. Thus, you can access each animal yourself, and do whatever you want with it.

As of this writing, there are 41 animals.

The names of the animals:

```
sort(names(animals))
#> [1] "ant"          "anxiouscat"    "bat"          "bat2"
#> [5] "behindcat"    "bigcat"        "buffalo"      "cat"
#> [9] "chicken"       "clippy"         "cow"          "daemon"
#> [13] "duck"         "duckling"      "egret"        "endlesshorse"
#> [17] "facecat"      "fish"          "frog"         "ghost"
#> [21] "grumpycat"    "hypnotoad"     "longcat"      "longtailcat"
#> [25] "monkey"        "mushroom"      "owl"          "pig"
```

## Say something

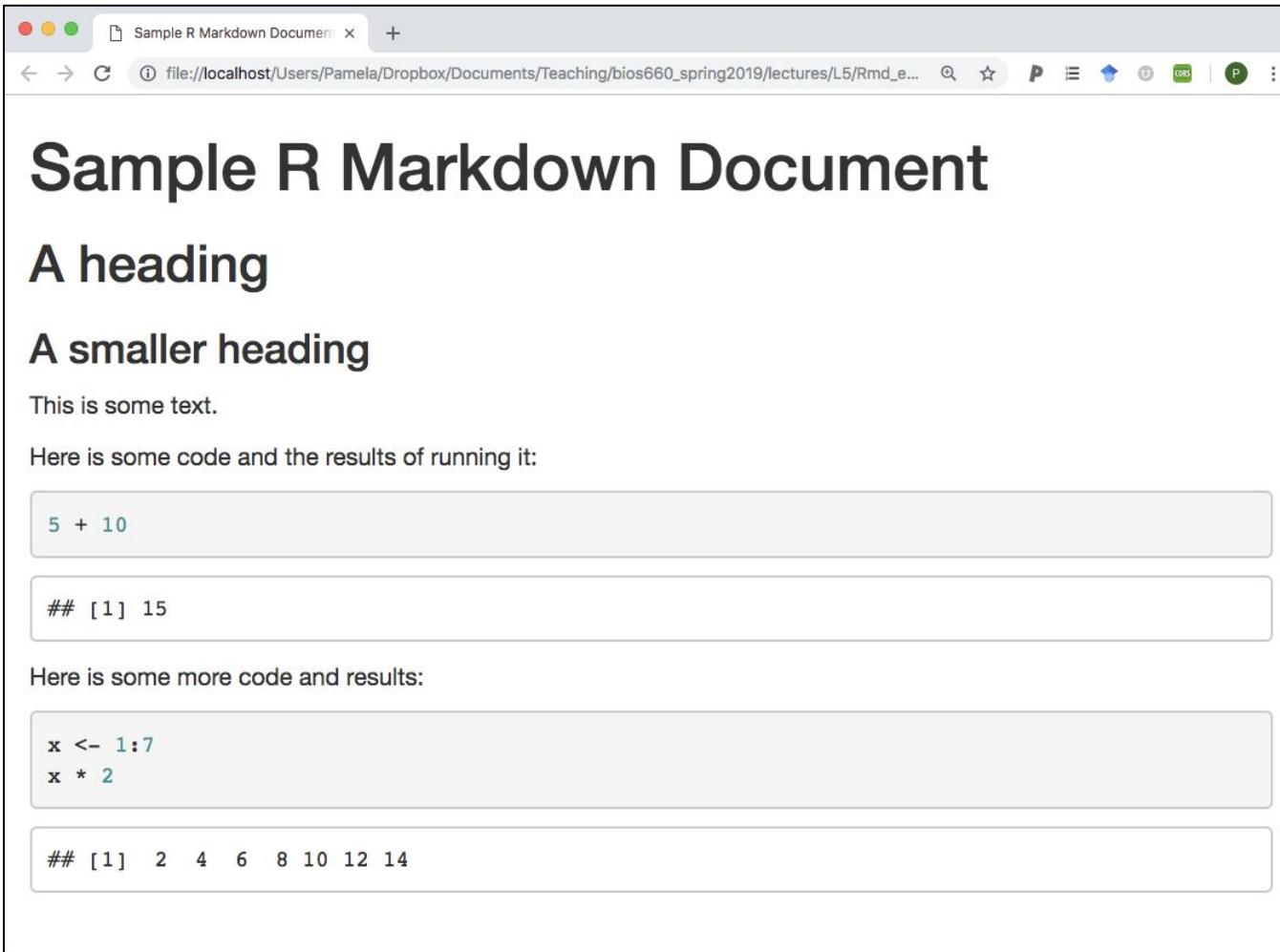
We expose the function `say()` in this package, which you can use to evoke any animal in the package, and make it say whatever you want. Some examples:

```
say("why did the chicken cross the road", "chicken")
#> Colors cannot be applied in this environment :( Try using a terminal or RStudio.
#>
#> -----
#> why did the chicken cross the road
#> -----
#> \
#> \
#> \
#> _/ \
#> ^ / \
#> ^ / \_-
#> | /'-+ .-+
#> \ ' .-'-'-'-'-
#> \ ' .-'-'-'-
#> ' .-'--.-^
#> ^ / -.-^
#> ^ / -.-^
#> -| -|
#> /` /` [nosig]
#>
```

```
say("boo!", "ghost")
#> Colors cannot be applied in this environment :( Try using a terminal or RStudio.
#>
#> -----
#> boo!
#> -----
#> \
#> \
#> .-.
#> (o o)
#> / o \
#> \ / \
#> `~~~' [nosig]
```

# Reports with R Markdown

Combine formatted text, R code and results



The screenshot shows a web browser window titled "Sample R Markdown Document". The URL in the address bar is "file:///localhost/Users/Pamela/Dropbox/Documents/Teaching/bios660\_spring2019/lectures/L5/Rmd\_e...". The page content is as follows:

# Sample R Markdown Document

## A heading

### A smaller heading

This is some text.

Here is some code and the results of running it:

```
5 + 10
```

```
## [1] 15
```

Here is some more code and results:

```
x <- 1:7
x * 2
```

```
## [1]  2  4  6  8 10 12 14
```

# What is markdown?

Simple syntax for defining formatting within plain text

Can be converted to formatted HTML and other formats

# Markdown

The screenshot shows the RStudio interface with a Markdown file open in the left pane and its rendered output in the right pane.

**Left Pane (Code Editor):**

```
1 ---  
2 title: "Markdown Demo"  
3 output: html_document  
4 bibliography: rmarkdown.bib  
5 ---  
6  
7 Markdown provides an easy way to make standard types of formatted  
text, like  
8  
9 - *italics*  
10 - **bold**  
11 - `code`  
12 - [links](rmarkdown.rstudio.com)  
13 - etc.  
14  
15 But did you know that you can also use R Markdown's markdown to  
make  
16  
17 - Latex equations, $E = mc^2$  
18 - And bibliographies [@rmarkdown15].  
19  
20 # References  
21  
22
```

**Right Pane (Viewer):**

## Markdown Demo

Markdown provides an easy way to make standard types of formatted text, like

- *italics*
- **bold**
- `code`
- [links](#)
- etc.

But did you know that you can also use R Markdown's markdown to make

- Latex equations,  $E = mc^2$
- And bibliographies (JJ Allaire 2015).

## References

JJ Allaire, et. al. 2015. *R Markdown*. <http://rmarkdown.rstudio.com>.

# What is R Markdown?

R Markdown documents can include formatted text and chunks of R code

Code and results can be included in generated HTML

# R Markdown

~/Documents/rmarkdown - gh-pages - RStudio

1-example.Rmd x Knit Addins Environment History Build Git Files Plots Packages Help Viewer

```
1---  
2 title: "Viridis Demo"  
3 output: html_document  
4---  
5  
6```{r include = FALSE}  
7 library(viridis)  
8```  
9  
10 The code below demonstrates two color palettes in the  
[viridis](https://github.com/sjmgarnier/viridis) package. Each  
plot displays a contour map of the Maunga Whau volcano in  
Auckland, New Zealand.  
11  
12## Viridis colors  
13  
14```{r}  
15 image(volcano, col = viridis(200))  
16```  
17  
18## Magma colors  
19  
20```{r}  
21 image(volcano, col = viridis(200, option = "A"))  
22```  
23
```

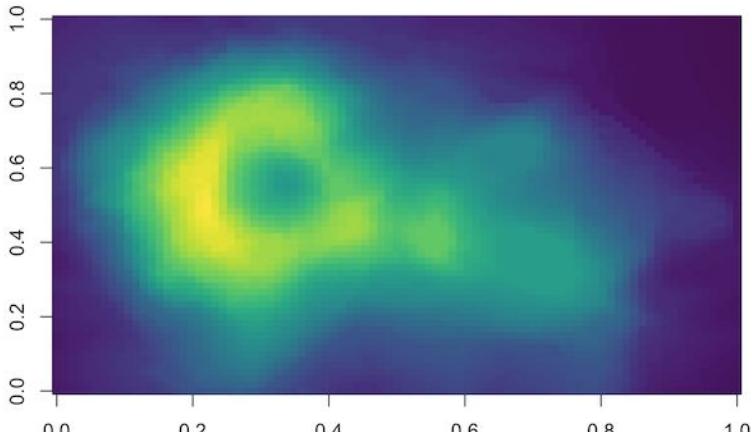
1:1 Viridis Demo R Markdown Console

## Viridis Demo

The code below demonstrates two color palettes in the `viridis` package. Each plot displays a contour map of the Maunga Whau volcano in Auckland, New Zealand.

### Viridis colors

```
image(volcano, col = viridis(200))
```



### Magma colors



# Anatomy of an R Markdown document

File parameters

Code chunk

Formatted text

Section title

Code chunk

The screenshot shows the RStudio interface with an R Markdown file open. The left pane displays the R Markdown code, and the right pane shows the rendered output.

**File parameters:** Lines 1-4 of the code, which define the title and output type.

**Code chunk:** Lines 6-8, which include a library call within a code chunk.

**Formatted text:** Line 10, which contains a block of text describing the purpose of the demo.

**Section title:** Line 12, which starts a section titled "## Viridis colors".

**Code chunk:** Lines 15-16, which include an image command within a code chunk.

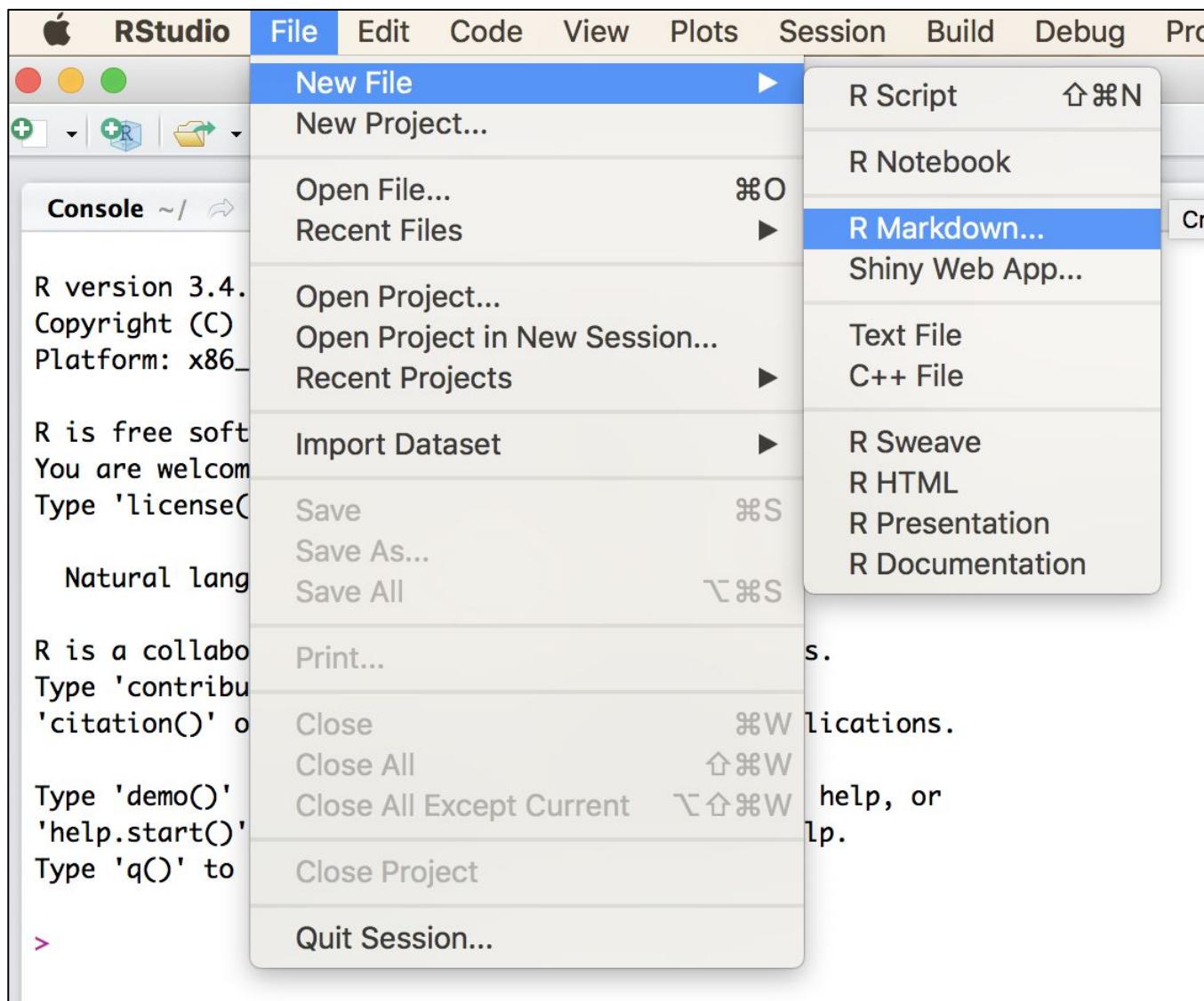
**R Markdown Output:** The right pane shows the rendered output. It includes a section titled "Viridis Demo" with a description of the code's purpose. Below that is a section titled "Viridis colors" containing the R code `image(volcano, col = viridis(200))`. At the bottom, there is a heatmap titled "Magma colors" showing a contour map of the Maunga Whau volcano in Auckland, New Zealand, using the Viridis color palette.

# Knitr

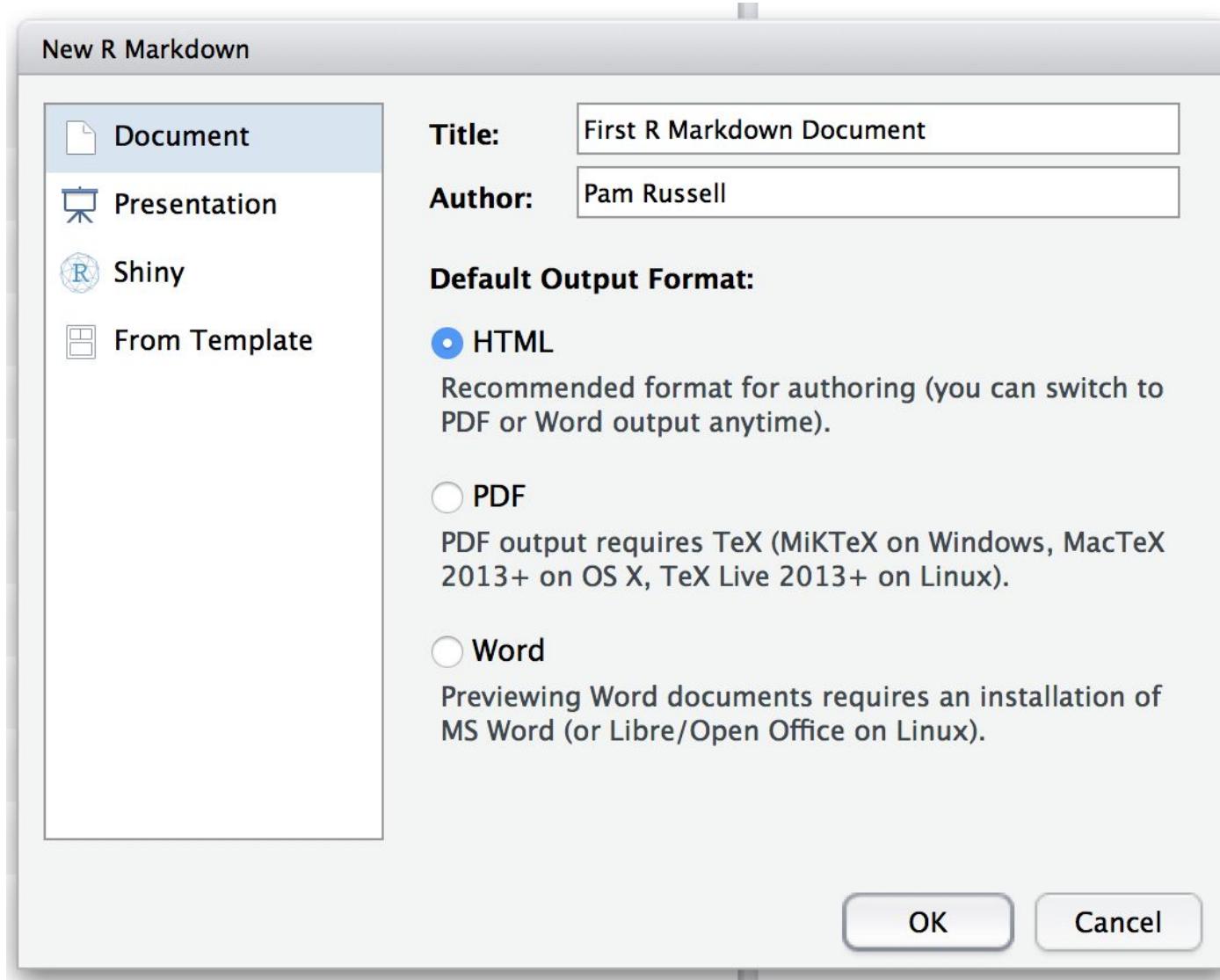
The engine that generates formatted reports from R Markdown (and other documentation languages)

Integrated into Rstudio

# Creating an R Markdown document



# Creating an R Markdown document



# Creating an R Markdown document

The screenshot shows the RStudio interface with the following details:

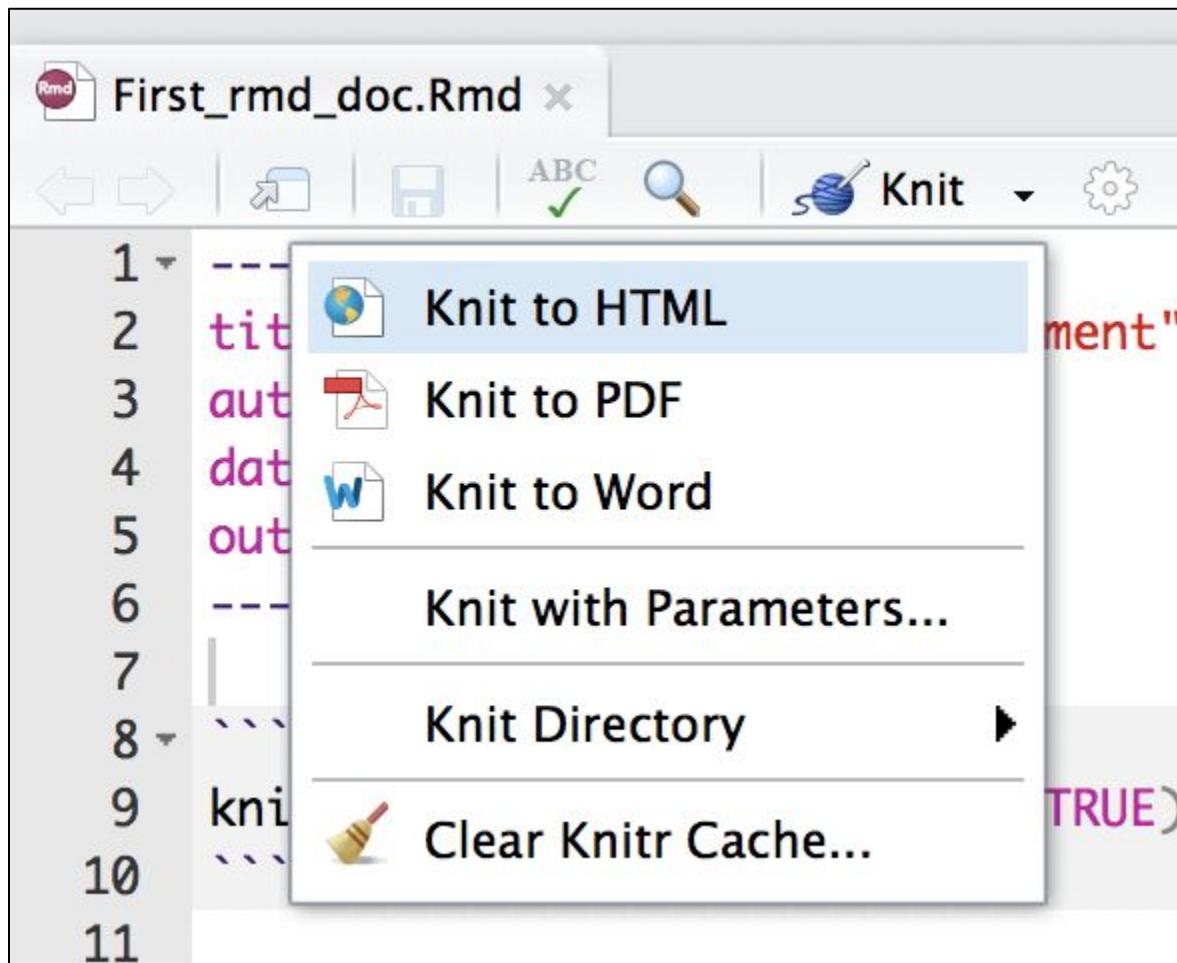
- Title Bar:** RStudio
- File Bar:** Go to file/function, Addins (None)
- Document Area:** The file "First\_rmd\_doc.Rmd" is open. The code content is as follows:

```
1 ---  
2 title: "First R Markdown Document"  
3 author: "Pam Russell"  
4 date: "10/16/2018"  
5 output: html_document  
6 ---  
7 |  
8 ````{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ````{r cars}  
19 summary(cars)  
20 ````  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ````{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ````  
29  
7:1 (Top Level) FALSE
```

- Environment Tab:** Environment, History, Connections
- Files Tab:** Files, Plots, Packages, Help, Viewer
- Console Tab:** Console, R Markdown (selected). The console output is:

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

# “Knitting” to HTML



# “Knitting” to HTML

RStudio  
Project: (None)

First\_rmd.Rmd

```
1 ---  
2 title: "First R Markdown Document"  
3 author: "Pam Russell"  
4 date: "10/16/2018"  
5 output: html_document  
6 ---  
7 |  
8 ``{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ...  
11 |  
12 ## R Markdown  
13 |  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring  
HTML, PDF, and MS Word documents. For more details on using R Markdown see  
http://rmarkdown.rstudio.com.  
15 |  
16 When you click the Knit button a document will be generated that includes both  
content as well as the output of any embedded R code chunks within the document. You  
can embed an R code chunk like this:  
17 |  
18 ``{r cars}  
19 summary(cars)  
20 ...  
21 |  
22 ## Including Plots  
23 |  
24 You can also embed plots, for example:  
25 |  
26 ``{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ...  
29 (Top Level) ▾
```

Console R Markdown

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

Environment History Connections

Files Plots Packages Help Viewer

First R Markdown Document

Pam Russell  
10/16/2018

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
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
```

## Including Plots

You can also embed plots, for example:



# Other output formats

- HTML document
- PDF
- Microsoft Word
- Slide presentation
- And more...

# R Markdown reference guide

<https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>



## R Markdown Reference Guide

Learn more about R Markdown at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com)  
Learn more about Interactive Docs at [shiny.rstudio.com/articles](http://shiny.rstudio.com/articles)

**Contents:**

- 1. Markdown Syntax**
- 2. Knitr chunk options
- 3. Pandoc options

Syntax	Becomes
Plain text	Plain text
End a line with two spaces to start a new paragraph.	End a line with two spaces to start a new paragraph.
*italics* and _italics_	<i>italics</i> and <i>italics</i>
**bold** and __bold__	<b>bold</b> and <b>bold</b>
superscript <sup>2</sup>	superscript <sup>2</sup>
~~strikethrough~~	strikethrough
[link](www.rstudio.com)	link
# Header 1	<h1>Header 1</h1>
## Header 2	<h2>Header 2</h2>

# **Knitr options**

Slides:

L5\_knitr\_options.html