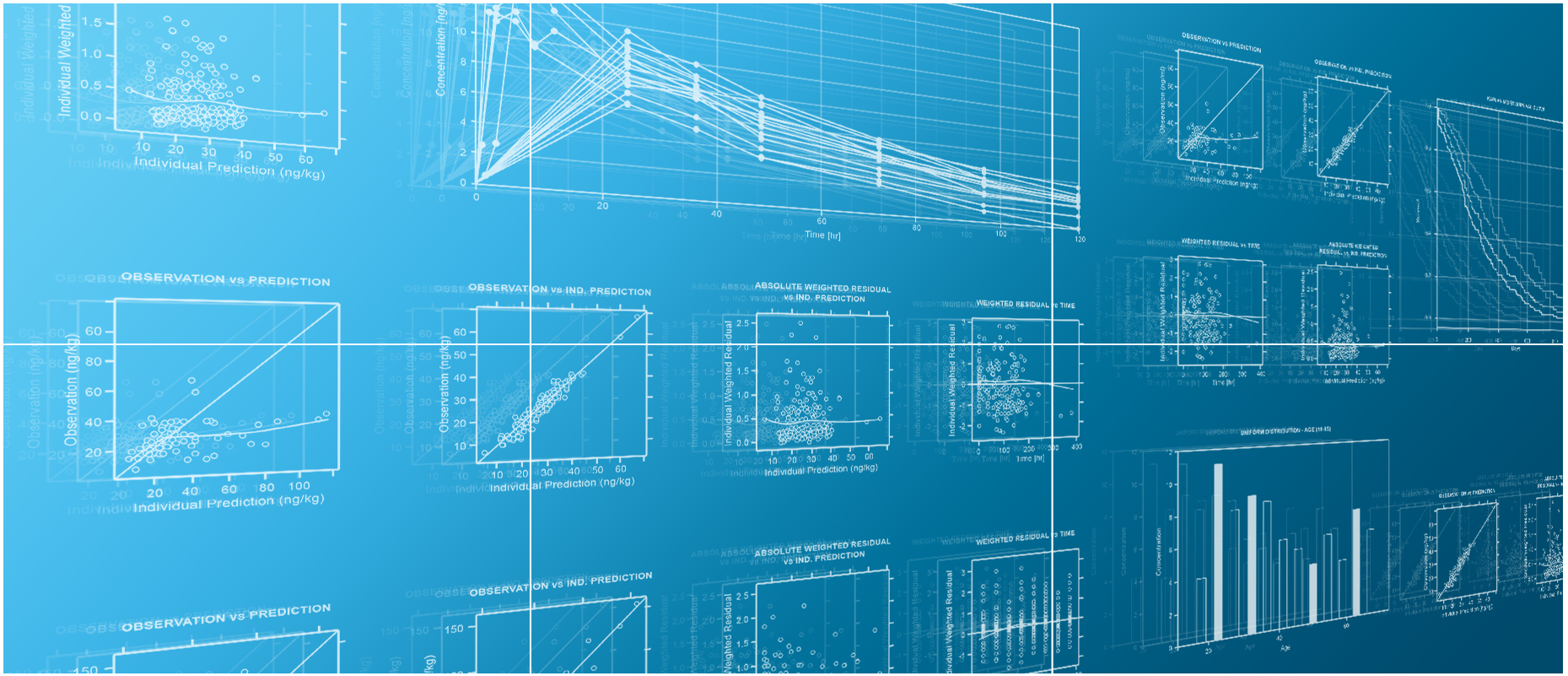
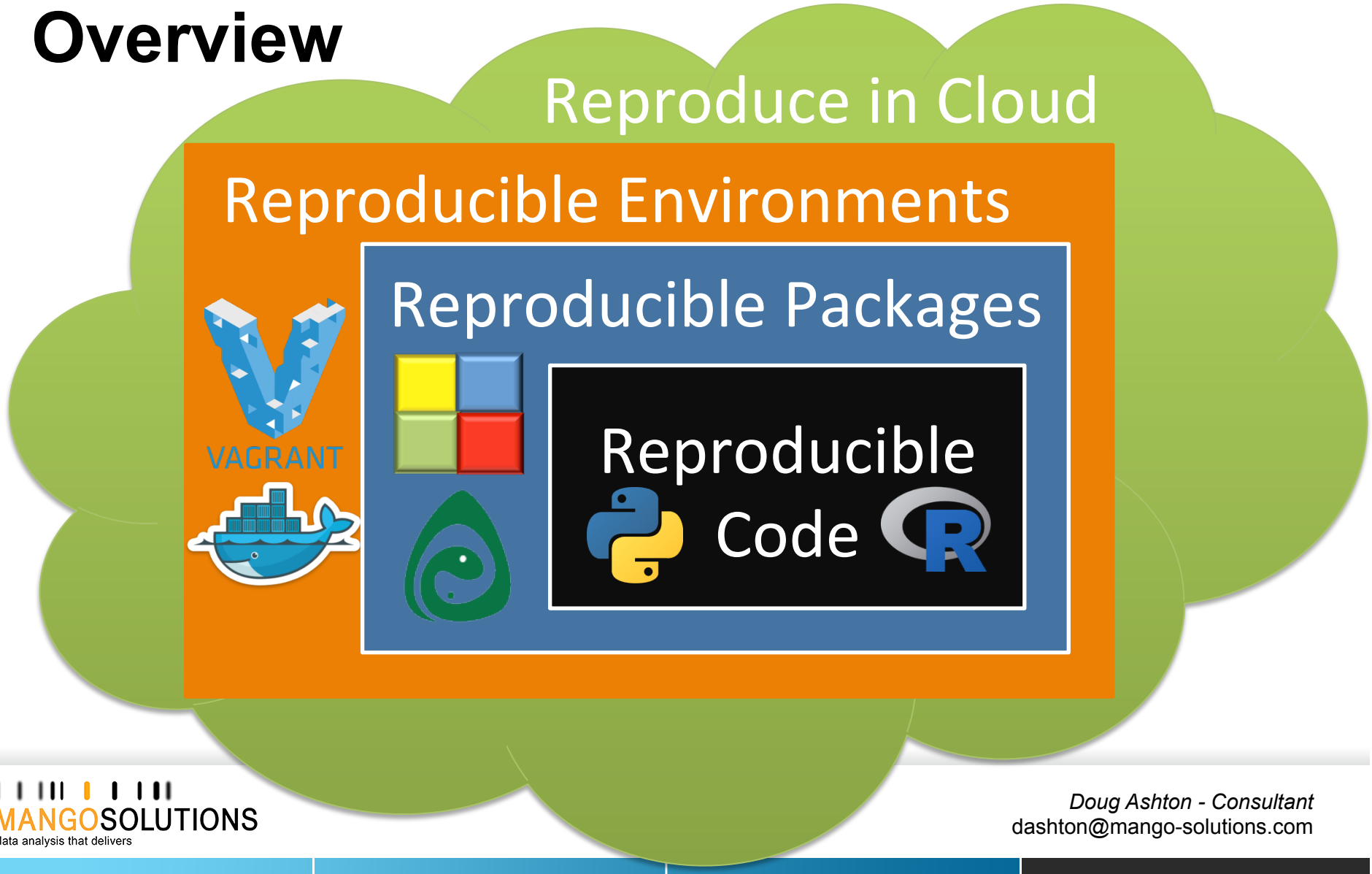
MANGOSOLUTIONS
data analysis that delivers

Reproducible Environments with Docker

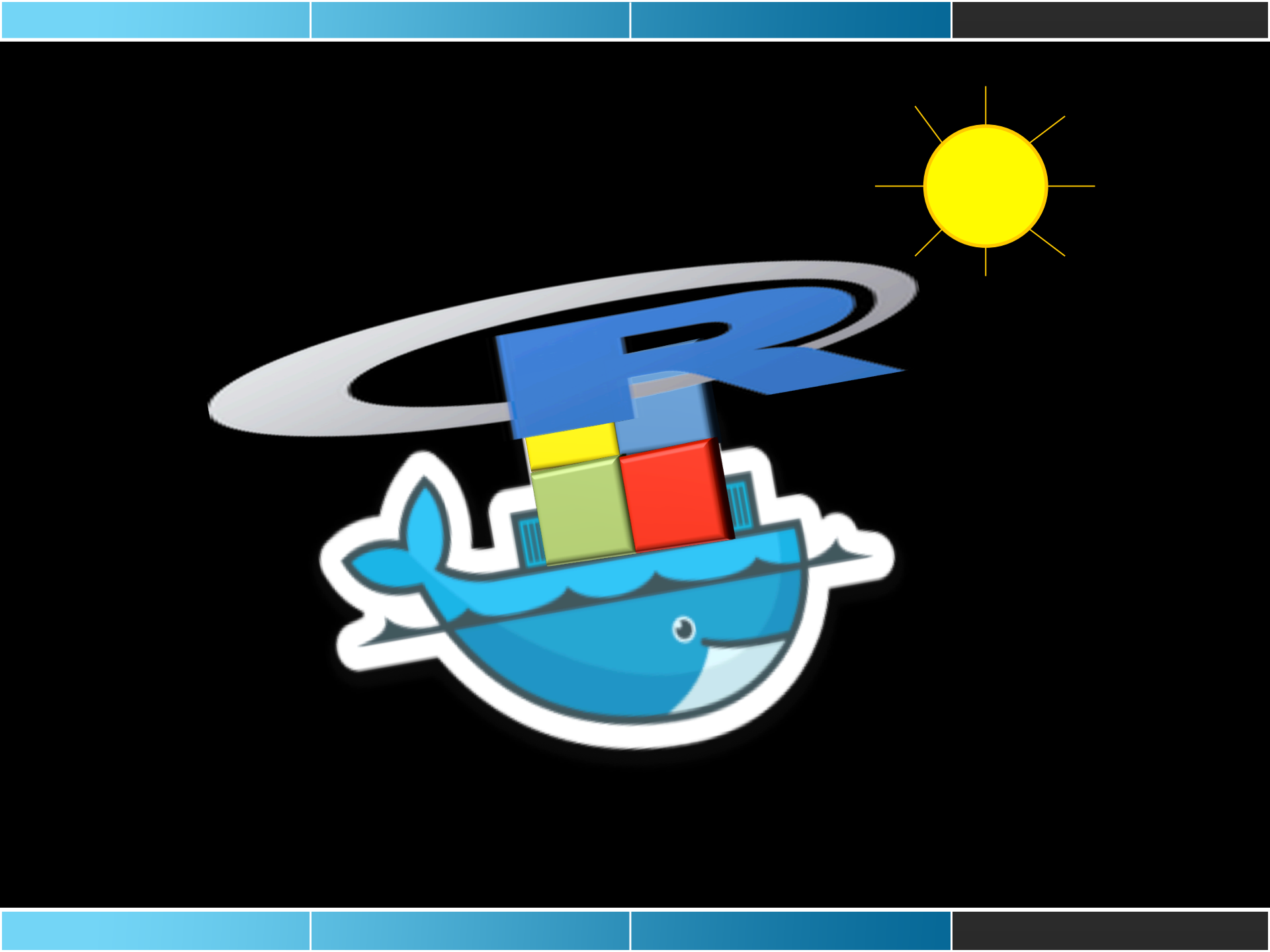


Doug Ashton

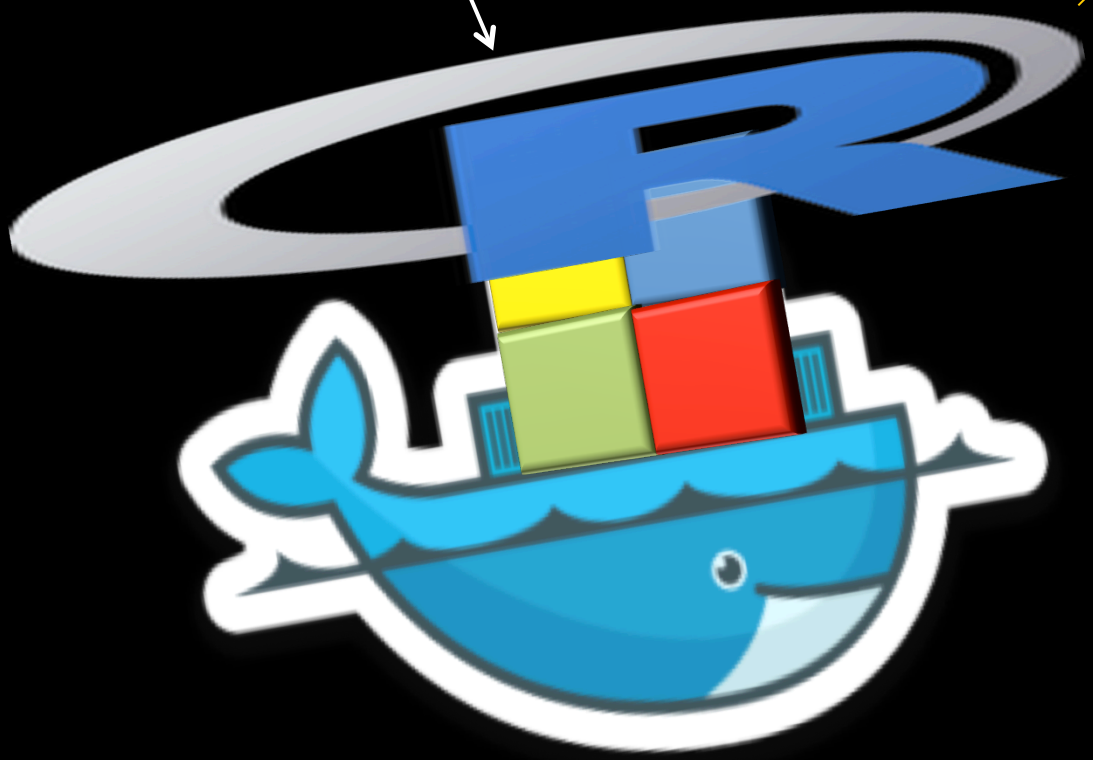
Overview



Doug Ashton - Consultant
dashton@mango-solutions.com



Reproducible Code



Can you just re-run those numbers for me?

Doug Ashton - Consultant
dashton@mango-solutions.com

Doug Ashton - Consultant
dashton@mango-solutions.com

Two Warnings

- Retracted cancer research – http://bit.ly/repres01 (video)
- Reinhart, Rogoff, and the Excel Error That Changed History - http://bit.ly/repres02

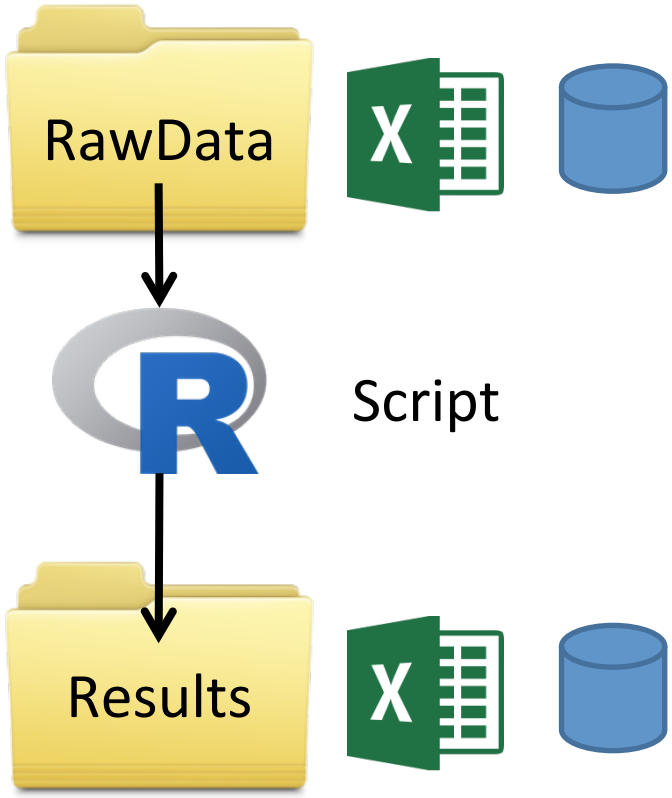
Doug Ashton - Consultant
dashton@mango-solutions.com

the solution

code

Doug Ashton - Consultant
dashton@mango-solutions.com

A Reproducible Flow



Doug Ashton - Consultant
dashton@mango-solutions.com

Demo

- https://github.com/dougmet/repEnvDemo

Doug Ashton - Consultant
dashton@mango-solutions.com

Dependencies

Input data

Manipulation

Plots

Calculation

Output data

```
library(zoo)
library(xts)
library(copula) # packrat headache

# Data Wrangle ------------------------------------------------------------

# Retrieve the processed daily data from csv
allClose <- read.zoo("data/allClose.csv",
                     format="%Y-%m-%d",header=TRUE)

# Get log daily returns
allLog <- diff(log(allClose))

# Sum to log weekly returns
allLogWk <- apply.weekly(allLog, colSums)


# Plot of the time series -------------------------------------------------
pdf("results/ts.pdf")
plot(allLogWk, main="Weekly Log Returns")
dev.off()


# Make a plot of empirical copulas ----------------------------------------
pdf("results/pairsCop.pdf")
pairs(pobs(allLogWk), main="Empirical Copulas")
dev.off()


# Correlation matrix ------------------------------------------------------
corMat <- cor(allLogWk, use="pairwise.complete.obs")

# Write out the correlation matrix
write.csv(corMat, "results/correlation.csv")
```


Doug Ashton - Consultant
dashton@mango-solutions.com

Version Control

- Edit with confidence
- ~~script-v1a.R~~ use tags
- Backup to remote
- Work together

Doug Ashton - Consultant
dashton@mango-solutions.com

Reproducible Reports

- Text and
- `code`
- All together
- `run demo report.Rmd in repo`

Doug Ashton - Consultant
dashton@mango-solutions.com

Market Correlation

Doug Ashton

06 October, 2016

Reproduce this document

```
seed <- 909
gitCommit <- system2("git", c("rev-parse", "HEAD"), stdout = TRUE)
gitBranch <- system2("git", c("rev-parse", "--abbrev-ref", "HEAD"), stdout = TRUE)
set.seed(seed)
repData <- data.frame(Git.Commit = gitCommit, Git.Branch = gitBranch,
                      Random.Seed = seed, R.Version = R.version$version.string,
                      stringsAsFactors = FALSE)
knitr::kable(repData, format = "markdown")
```

Git.Commit	Git.Branch	Random.Seed	R.Version
37b1435fad26e116adb459858486b3b3e147f635	odsc	909	R version 3.2.3 (2015-12-10)

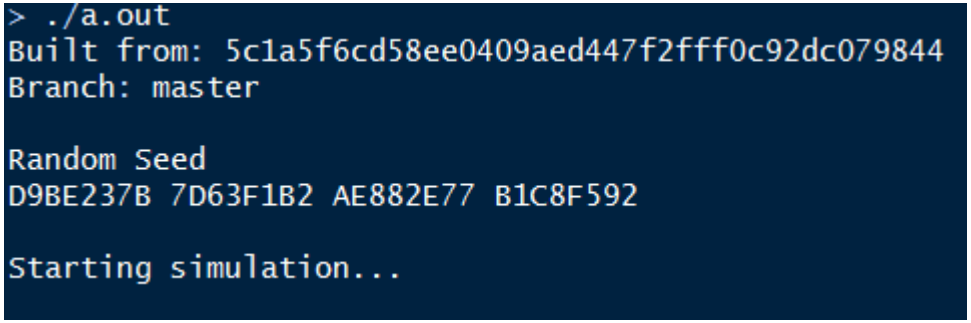
It is nice to layout all dependencies at the top.

```
library(zoo)
library(xts)
```


Doug Ashton - Consultant
dashton@mango-solutions.com

Literate Programming Tools

- Knitr/Sweave
- rmarkdown
- Jupyter (Ipython) Notebook
- RCloud
- git



Doug Ashton - Consultant
dashton@mango-solutions.com

Data Pipelines Interlude (my next talk)

Data Pipelines

```
raw -> clean -> prepared
```

- “Make” for data
  - Drake, Airflow (Python)
  - Remake (R)
  - Many more
- Awesome-pipline list: http://bit.ly/repres03

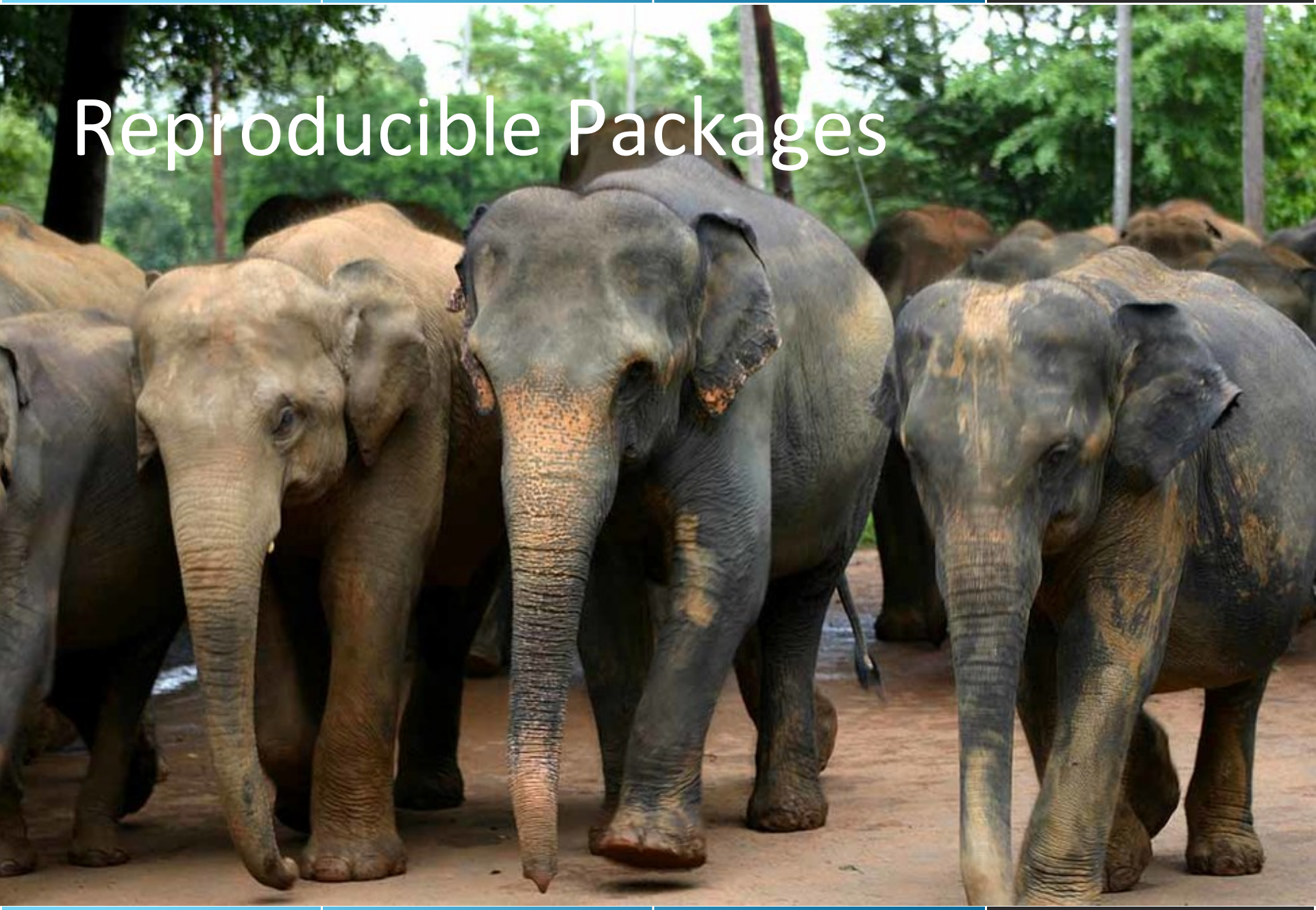
Doug Ashton - Consultant
dashton@mango-solutions.com

Reproducible Code Summary

```
Raw data + 6c30934a0ea2c0473d37b6d
= reproducible results
```


Doug Ashton - Consultant
dashton@mango-solutions.com

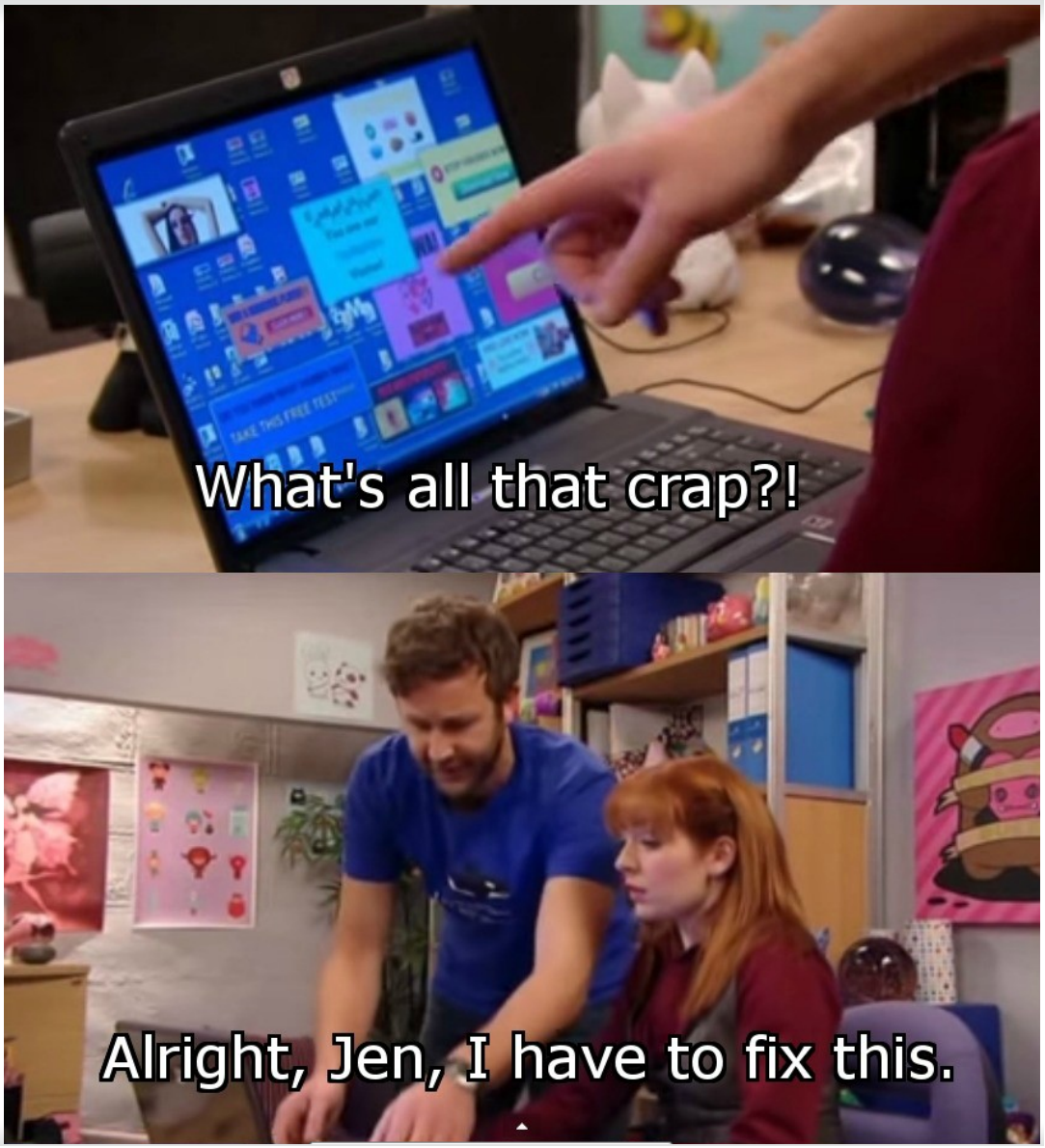
Reproducible Packages



the new problem

Doug Ashton - Consultant
dashton@mango-solutions.com

Your Laptop



Doug Ashton - Consultant
dashton@mango-solutions.com

Doug Ashton - Consultant
dashton@mango-solutions.com

Their server

Doug Ashton - Consultant
dashton@mango-solutions.com

Package Versions

- R: Packrat/checkpoint/switchr + conda
- R: MRAN mirror snapshots
  - E.g. https://mran.revolutionanalytics.com/snapshot/2015-03-01
- Python: Conda / pip

Doug Ashton - Consultant
dashton@mango-solutions.com

Package Management Packages

Packages	Pros	Cons
Packrat	Built into Rstudio	Bulky in repo (eg git) Needs foresight
Checkpoint	Easy Rescues old scripts	Lots of libraries Only one date
Switchr	Package manifests Repo friendly	Not easy

Doug Ashton - Consultant
dashton@mango-solutions.com

Packrat

```
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.

> packrat::init()
Initializing packrat project in directory:
- "~/Documents/Talks/RepEnvironments/packrat"

Adding these packages to packrat:

                _
    ADGofTest   0.3
    Matrix      1.2-2
    TTR         0.23-0
    colorspace  1.2-6
    copula      0.999-14
    gsl         1.9-10
    lattice     0.20-33
    mvtnorm     1.0-3
    packrat     0.4.6-1
    pspline     1.0-17
    quantmod    0.4-5
    stabledist  0.7-0
    xts         0.9-7
    zoo         1.7-12

Fetching sources for ADGofTest (0.3) ... OK (CRAN current)
Fetching sources for Matrix (1.2-2) ... OK (CRAN current)
Fetching sources for TTR (0.23-0) ... OK (CRAN current)
Fetching sources for colorspace (1.2-6) ... OK (CRAN current)
Fetching sources for copula (0.999-14) ...
```

Packrat

Pros

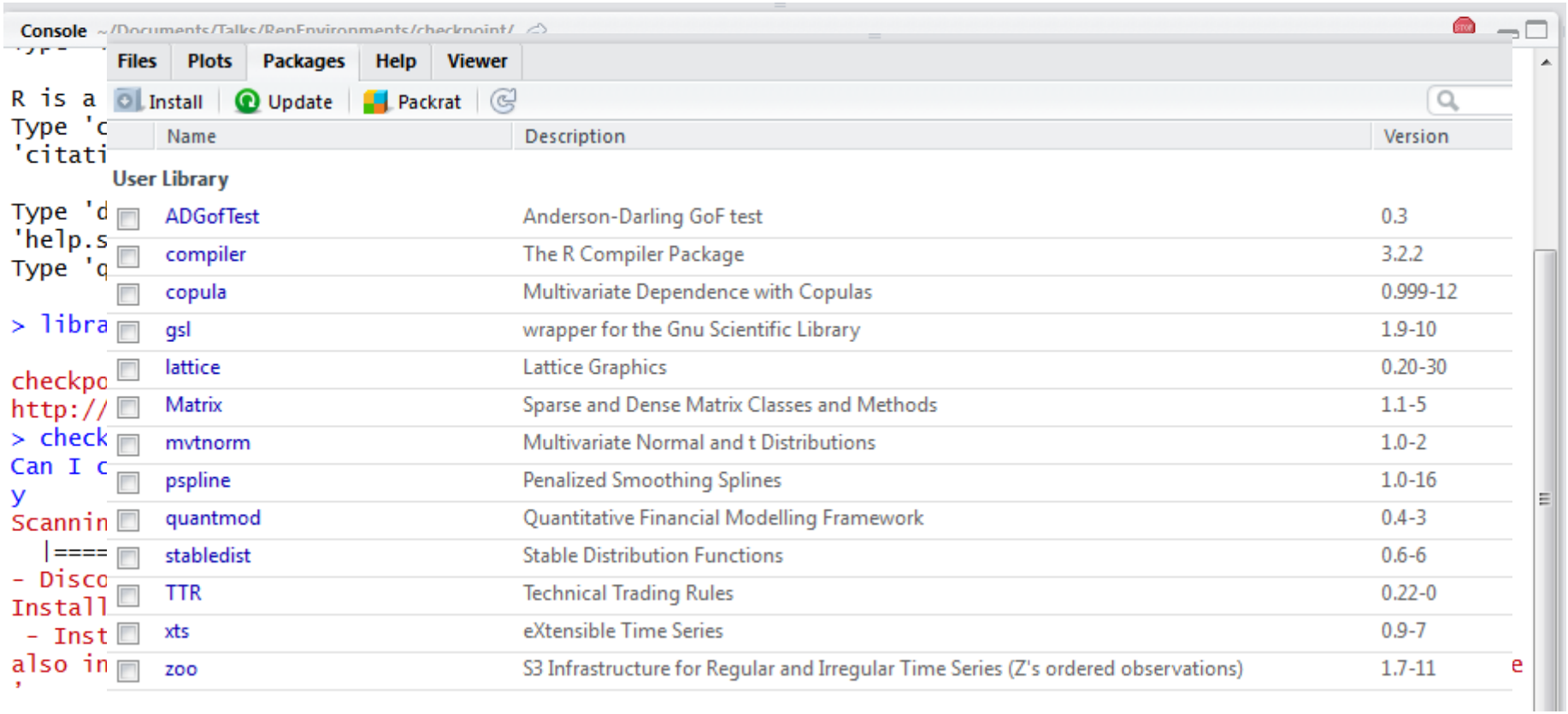
- Baked into RStudio
- Works how you expect
  `install.packages(...)`
  `install_github(...)`
- Not much else to do

Cons

- Saves entire source into repo (potentially bulky)
  - There is a workaround
- Plan in advance

Doug Ashton - Consultant
dashton@mango-solutions.com

Checkpoint



Doug Ashton - Consultant
dashton@mango-solutions.com

Switchr

- No demo

Doug Ashton - Consultant
dashton@mango-solutions.com

SwitchR

Pros

- Save versioned package inventory
- Very repository friendly

Cons

- Not as easy to use
- Might struggle with custom sources

Doug Ashton - Consultant
dashton@mango-solutions.com

Python Package Management

- conda
- Listing packages
  - `conda list`
- Save environment
  - `conda env export > environment.yml`
- Install environments
  - `conda env create -f environment.yml`

- pip / virtualenv
- Listing packages
  - `pip list`
- Save/freeze packages
  - `pip freeze > requirements.txt`
- Install
  - `pip install -r requirements.txt`

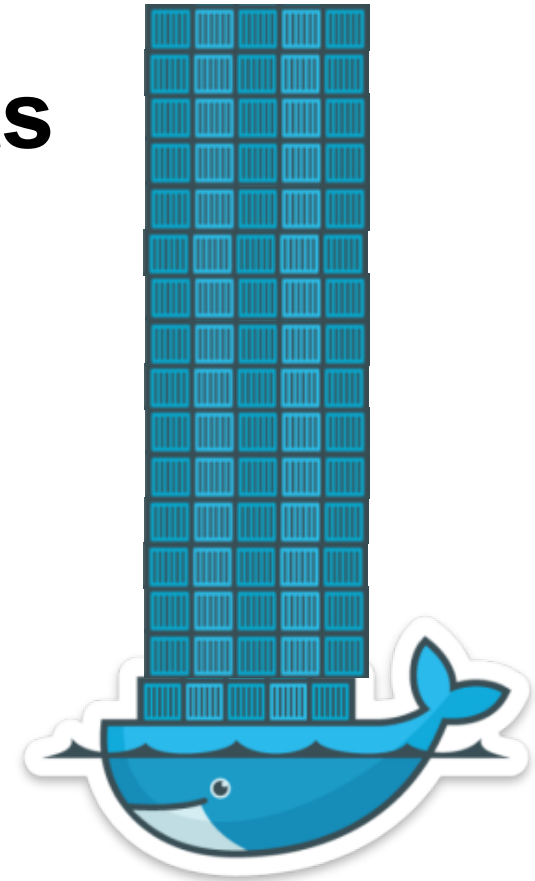
Doug Ashton - Consultant
dashton@mango-solutions.com

Reproducible Environments

Reproducible Environments

1. System Dependencies

2. Managed Environments



3. Scalability

Doug Ashton - Consultant
dashton@mango-solutions.com

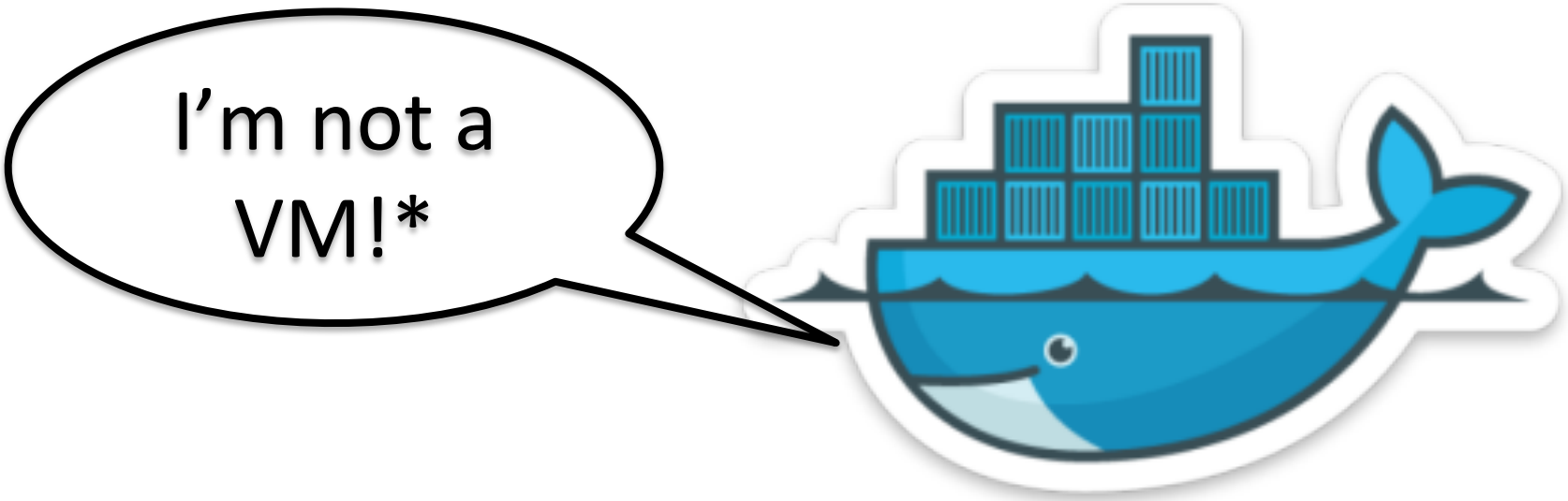
Vagrant

- www.vagrantup.com
- Primarily for development
- Each project gets a virtual machine
- https://github.com/dougmet/vagrantR

Doug Ashton - Consultant
dashton@mango-solutions.com

Docker

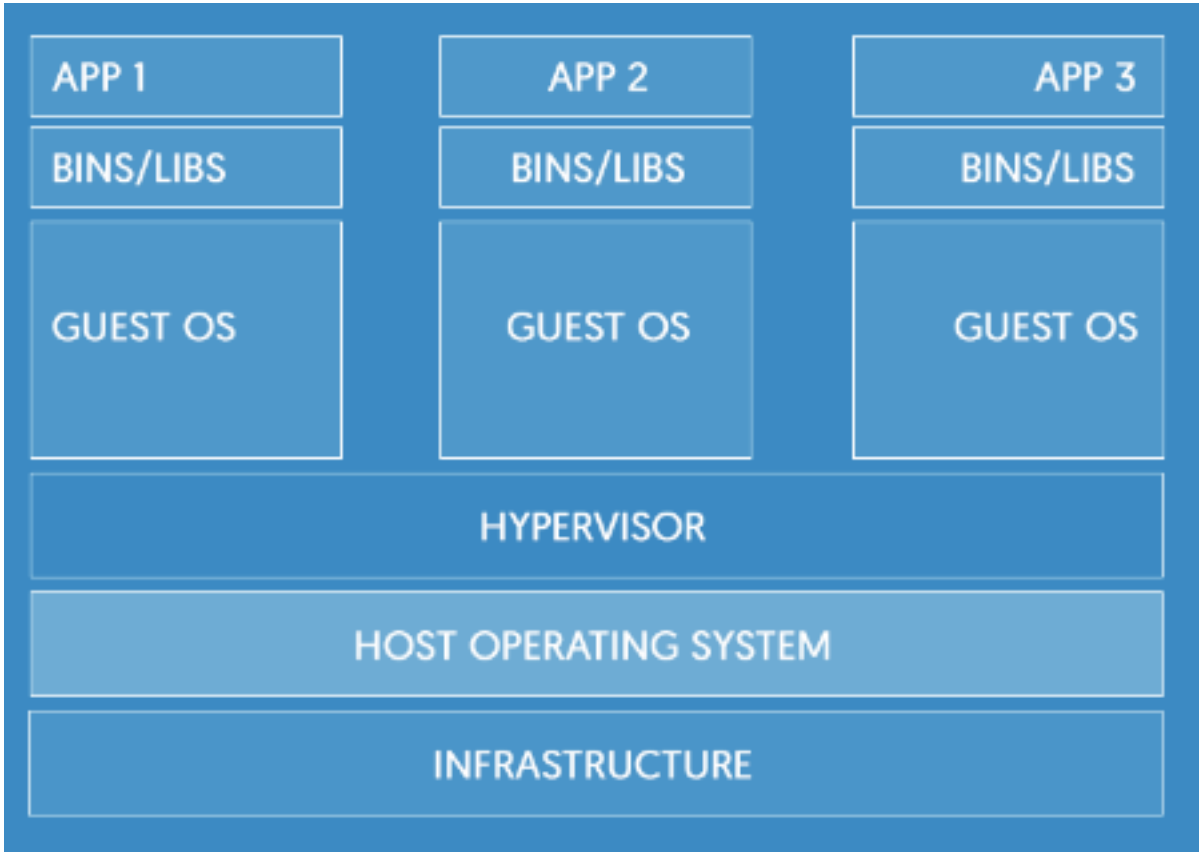
- www.docker.com
- Primarily for deployment



* OK, except Windows and Mac

Doug Ashton - Consultant
dashton@mango-solutions.com

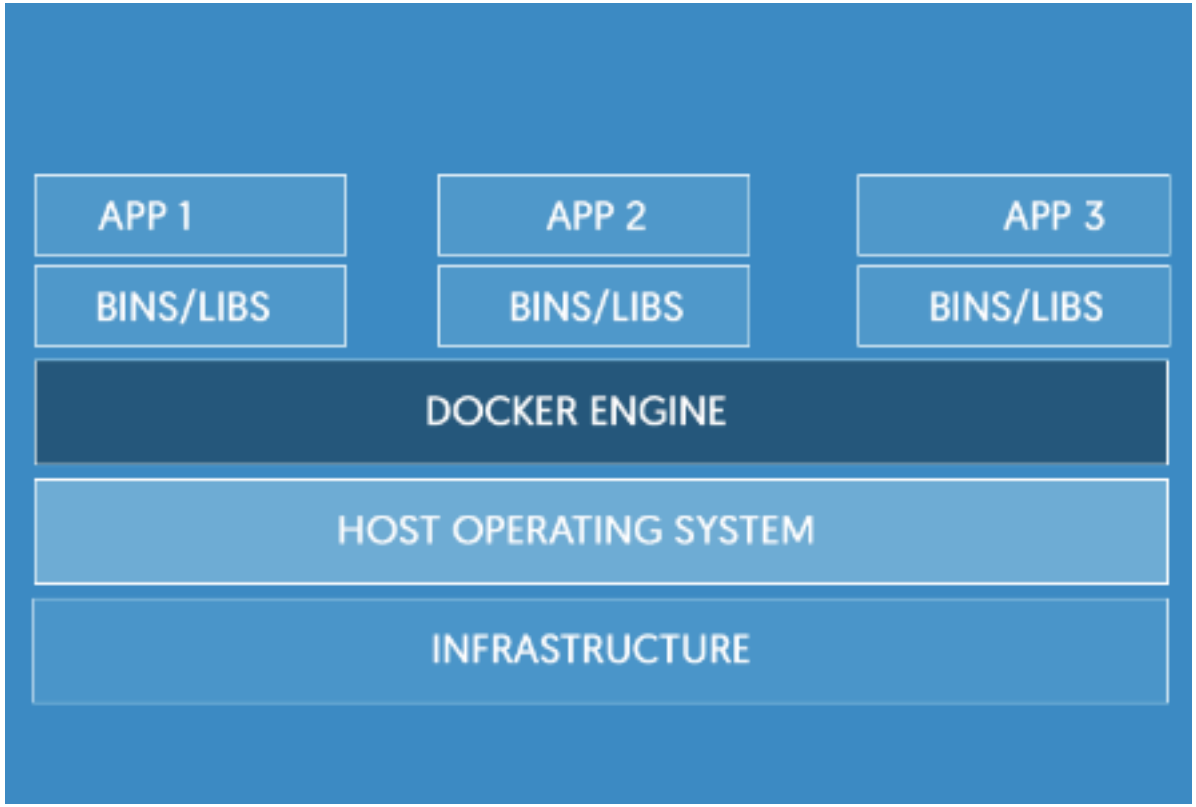
How's it different to a VM?



Source:
Docker Website

Doug Ashton - Consultant
dashton@mango-solutions.com

How's it different to a VM?



Source:
Docker Website

Doug Ashton - Consultant
dashton@mango-solutions.com

How's it different to a VM?

- Shares resources (same kernel)
- Starts nearly instantaneously
- Destroy just as quickly
- Launch as many as you want

Doug Ashton - Consultant
dashton@mango-solutions.com

Old Docker For Windows / Mac

- Linux on Virtual Box
- docker-machine application

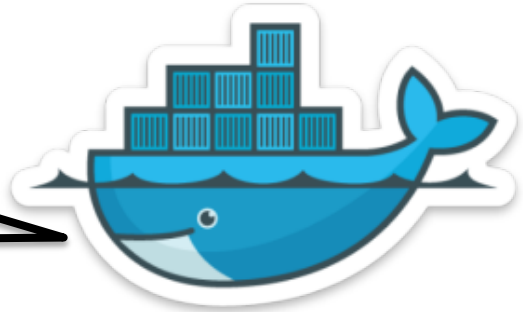
Doug Ashton - Consultant
dashton@mango-solutions.com

New Docker For Windows / Mac

- Mac
- Linux on **xhyve**
- Can use alongside old Docker

- Windows
- Linux on **Hyper-V**
- **Cannot** use alongside old Docker

There's also Windows on Windows Containers http://bit.ly/repres05



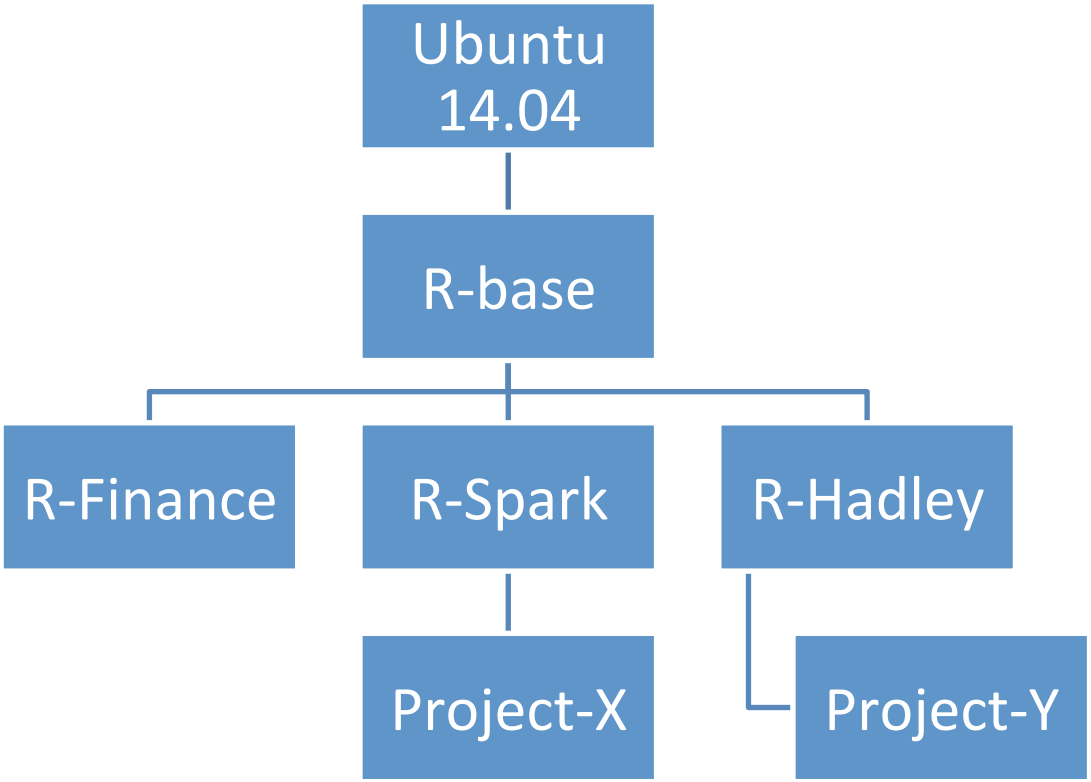
Doug Ashton - Consultant
dashton@mango-solutions.com

Key Components

- Images
  - Big (>200Mb). A whole system. `docker images`
- Registry
  - Repository of images `docker pull <image>`
- Docker File
  - Recipe to build an image `docker build <dockerfile>`
- Containers
  - An instance of an image. `docker run <image>`

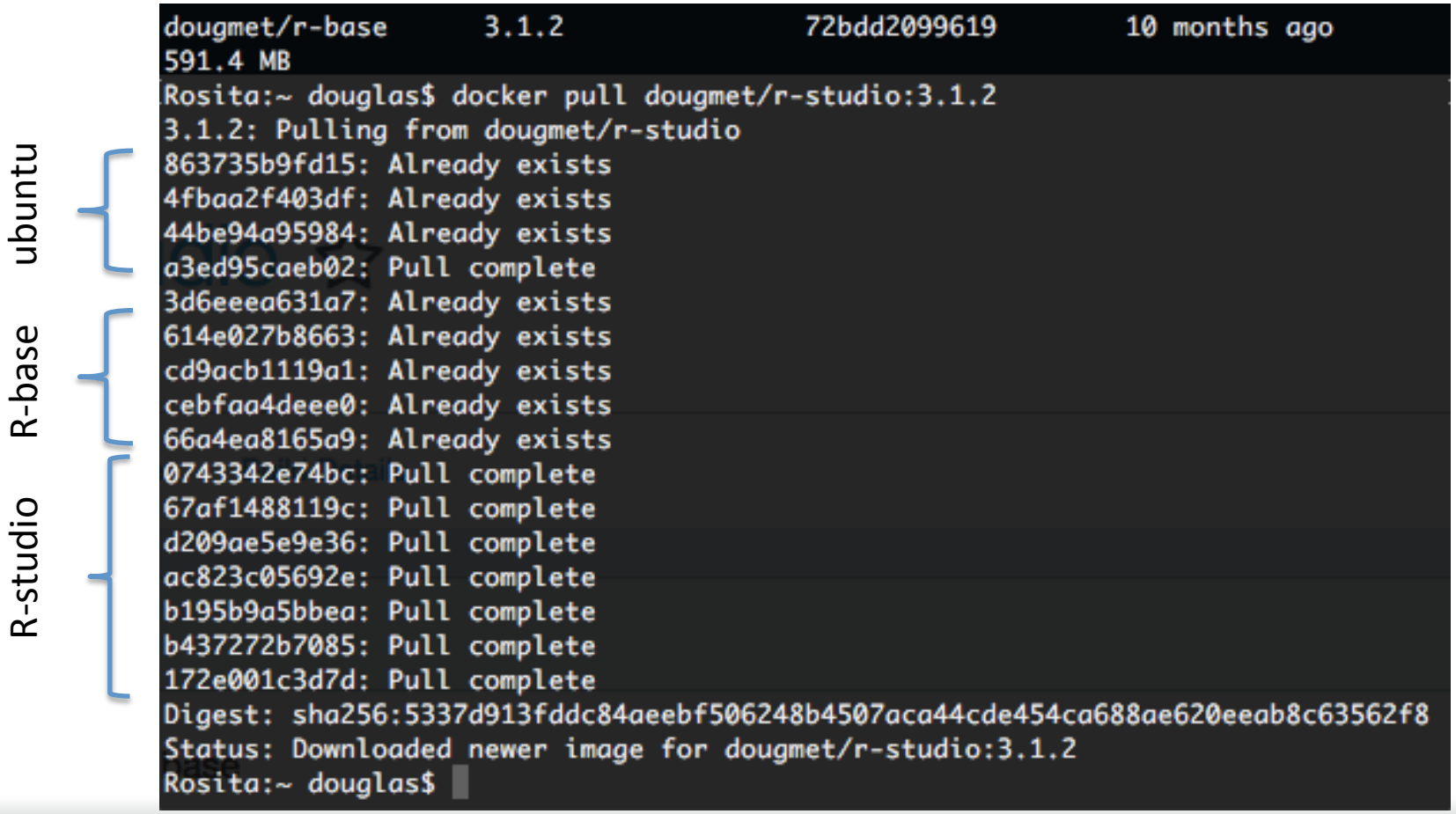
Doug Ashton - Consultant
dashton@mango-solutions.com

Docker Images are Hierachical



Doug Ashton - Consultant
dashton@mango-solutions.com

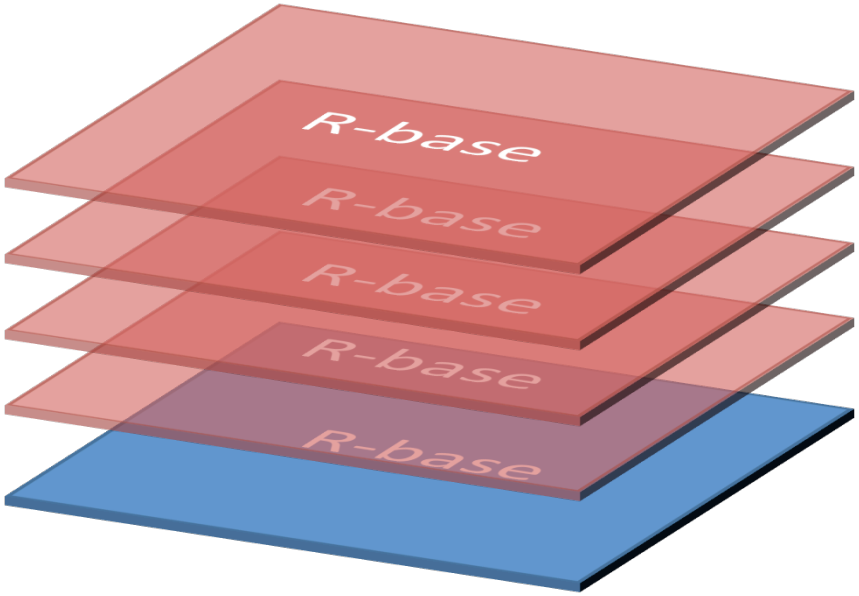
Docker Images are Hierachical



Doug Ashton - Consultant
dashton@mango-solutions.com

Docker Containers

Multiple containers from one image, launch rapidly



Doug Ashton - Consultant
dashton@mango-solutions.com

Sandbox

- Demo rocker/hadleyverse http://bit.ly/repres06
- jupyter/all-spark-notebook http://bit.ly/repres07

- `docker run -d -p 8787:8787 -v $PWD:/home/rstudio/host dougmet/r-studio:3.1.2`
  - `-d` keep running in background
  - `-p` map local port to container port
  - `-v` map a local volume to container

Doug Ashton - Consultant
dashton@mango-solutions.com

Rocker Warning

cboettig commented 27 days ago — Owner

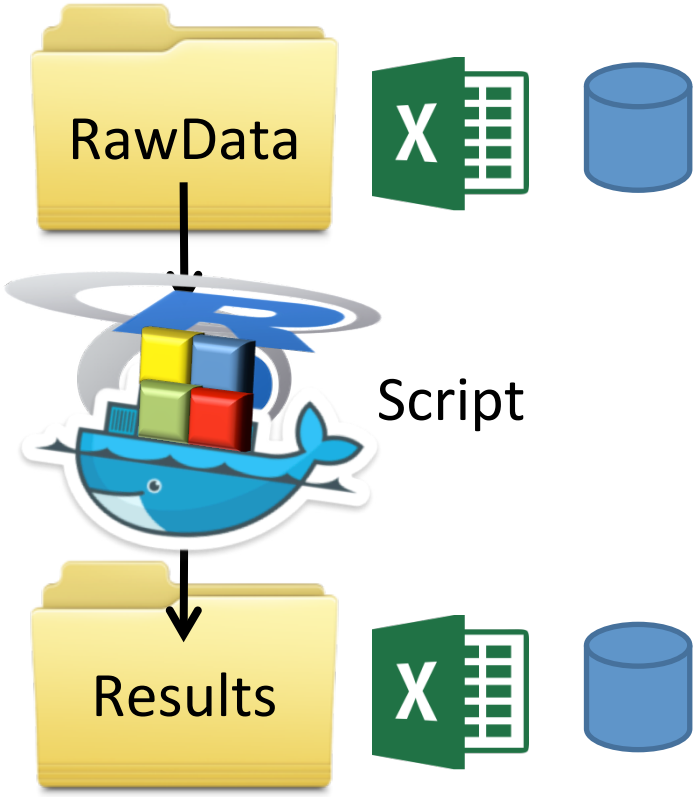
@dougmet Yeah, sorry about that. The docker container pulls whatever it gets from the debian:testing repos, so once those updated, 3.1.2 updated as well.

Future tags shouldn't do this (I think), because the version number is now explicit in the `apt-get` line.

Nevertheless, because our image builds R from the Debian binaries rather than from source (but see rocker/r-devel for the latter), the r-base images will probably never be well suited for running older R versions. That is, you couldn't come back and run the Dockerfile for 3.2.2 a few years later and get R 3.2.2, because apt-get simply wont be able to find that version on the debian:testing repos in a few years, which will have moved on.

Doug Ashton - Consultant
dashton@mango-solutions.com

A Reproducible Flow



Doug Ashton - Consultant
dashton@mango-solutions.com

Docker Demo

- https://github.com/dougmet/dockerR

```
FROM dougmet/r-base:3.1.2

RUN apt-get -y install libgsl0ldbl=1.16*1 libgsl0-dev=1.16*

# Install R package manifest
COPY loadPackages.R /tmp/
COPY packages.csv /tmp/
RUN Rscript /tmp/loadPackages.R

CMD ["R"]
```


Doug Ashton - Consultant
dashton@mango-solutions.com

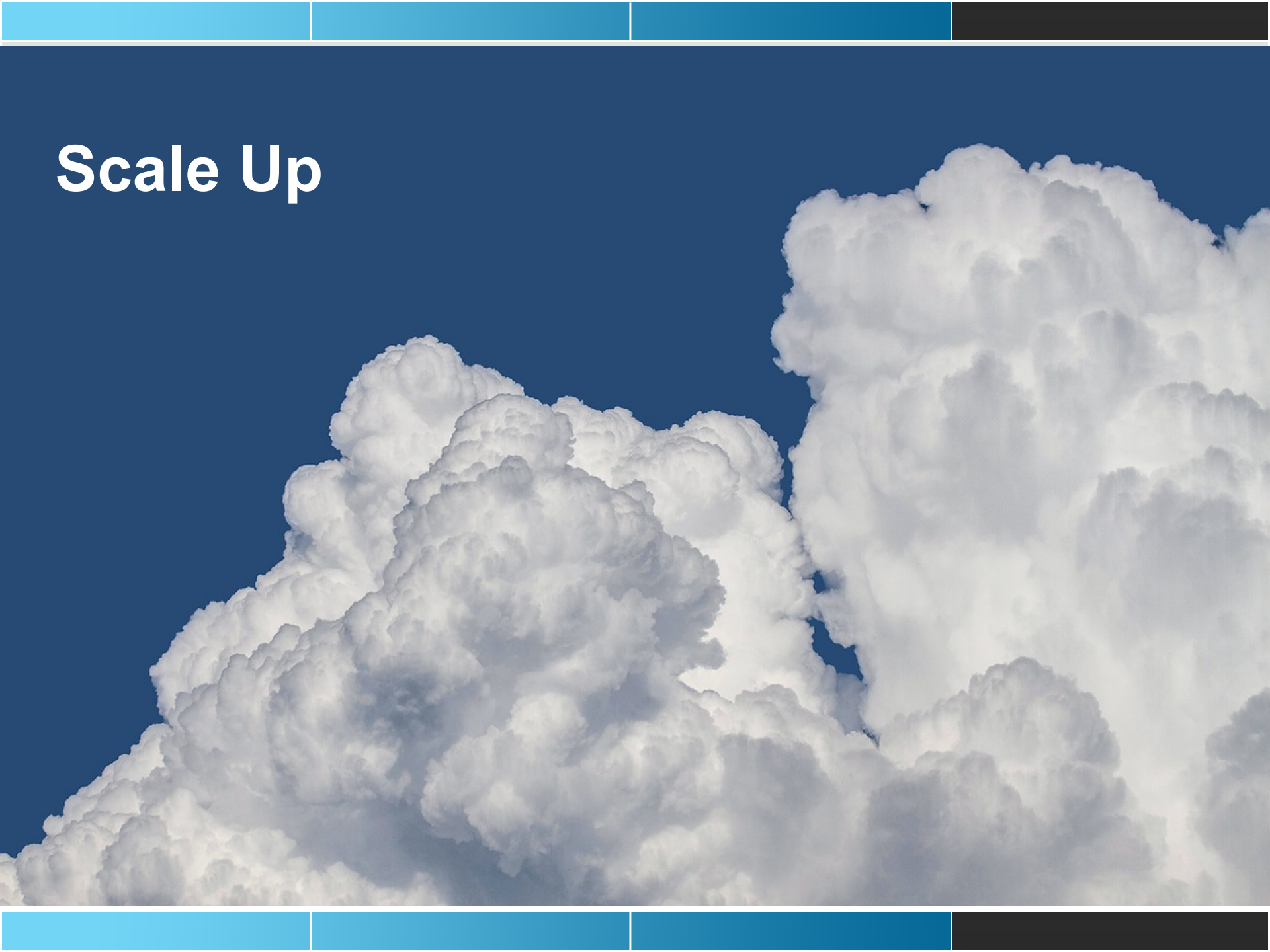
A Reproducible Flow

- `docker run --rm -v $PWD:/home/docker/host repenv Rscript /home/docker/host/script.R`

- `--rm` : Delete container when done
- `-v` : Mount local volume to container
- `repenv` : The image to run
- `Rscript` : The command to run

Doug Ashton - Consultant
dashton@mango-solutions.com

Scale Up



Moving to the cloud

- Digital Ocean Demo
  - Use the Docker “one click app”
  - Clone repo. Build image. Done.
- You could use anything (AWS, Azure, etc)

Doug Ashton - Consultant
dashton@mango-solutions.com

Docker Swarm

- “turns a pool of Docker hosts into a single, virtual Docker host”
- Launch lots of containers
- Out of my league!

Doug Ashton - Consultant
dashton@mango-solutions.com

Summary

- Use code
  - Version Control
  - Keep text with code (notebooks / Rmd)
- Save package versions in repo
  - R: Packrat / pkgsnap / MRAN
  - Python: Conda
- Save the environment in repo
  - Docker / Vagrant File

Doug Ashton - Consultant
dashton@mango-solutions.com

Get in touch

- GitHub
  - MangoTheCat
  - dougmet
- Twitter
  - @dougashton
- Email
  - dashton@mango-solutions.com

Doug Ashton - Consultant
dashton@mango-solutions.com