Doing Data Science Right in Excel-Pervasive Utilities

Eina Ooka

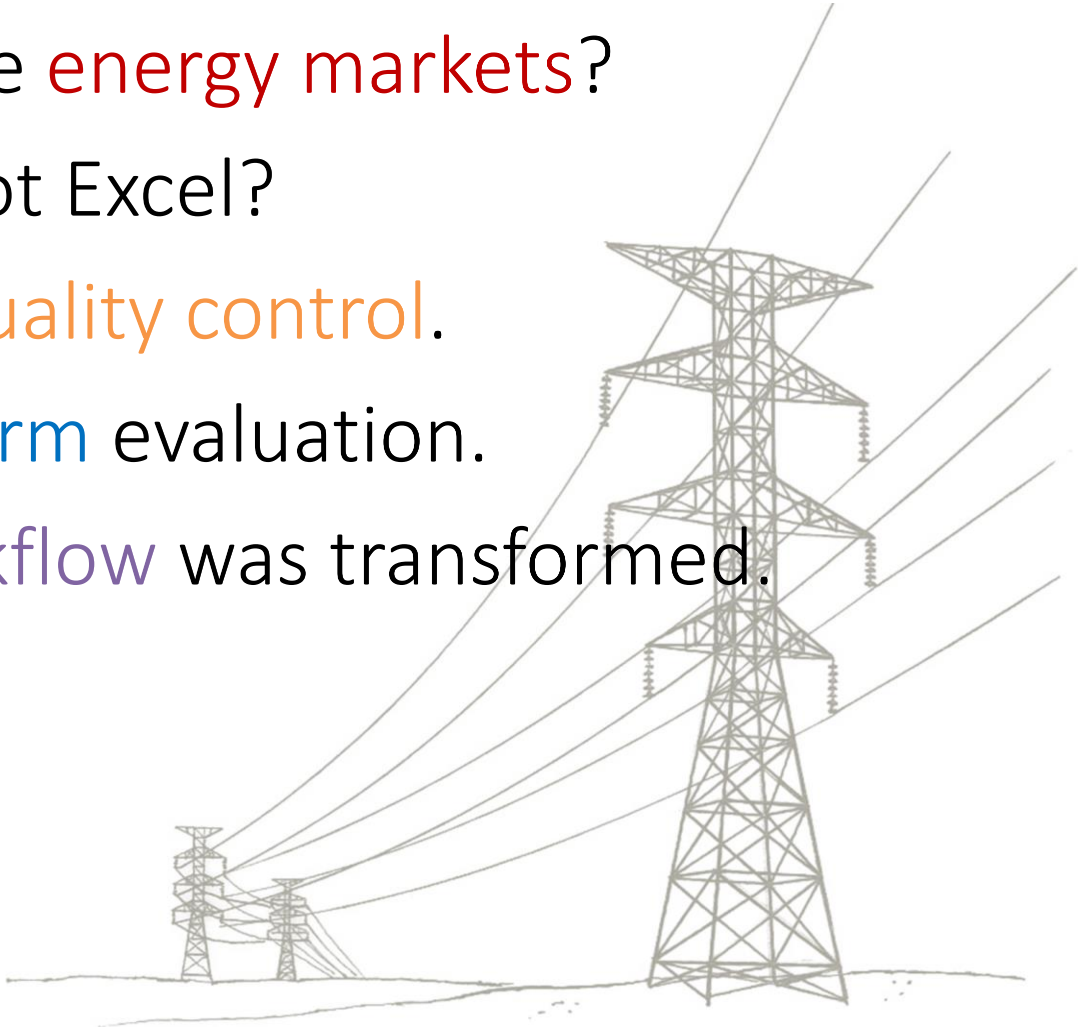
November, 2018

TEA
THE EnergyAuthority

EARL CONFERENCE 2018

Agenda

- ✓ What are wholesale energy markets?
- ✓ Why Excel? Why not Excel?
- ✓ Aims in analytics quality control.
- ✓ Data science platform evaluation.
- ✓ How analytics workflow was transformed.

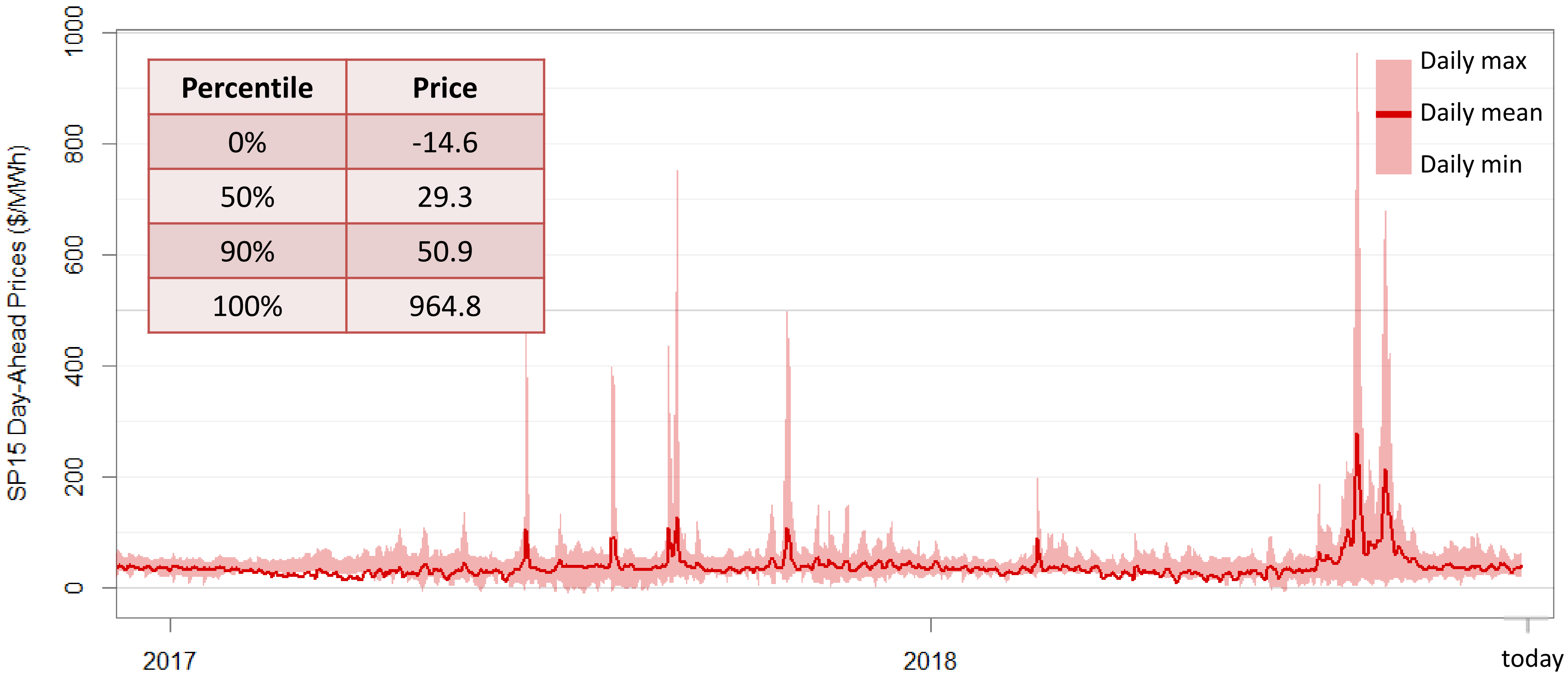


Wholesale Energy Markets

Power Utility Industry

- The Energy Authority serves public utilities nationwide for trading and analytics.
- Analytics team provides various forecasting and analysis services.
- A few dozen analysts in the team.

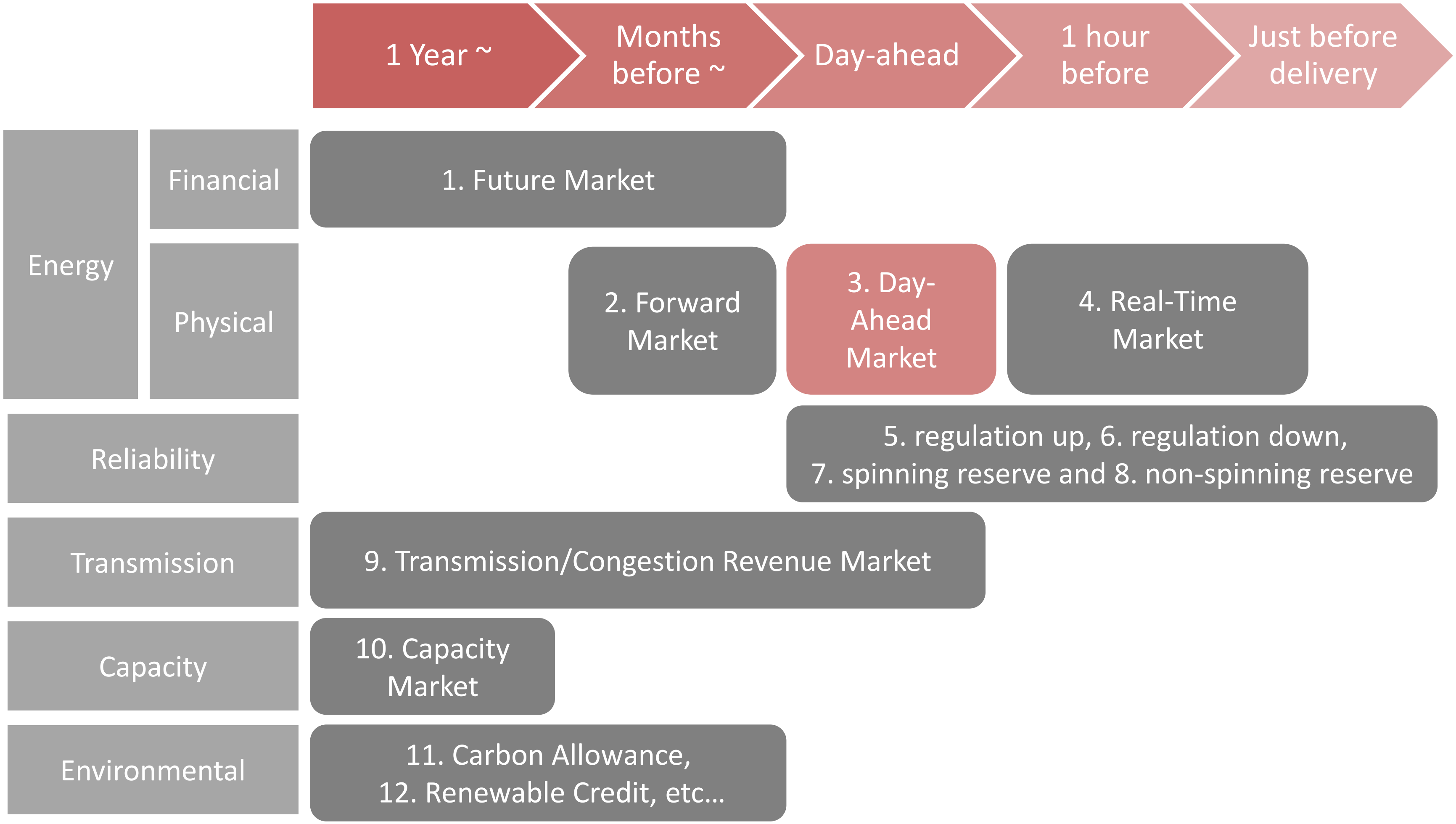
Wholesale Energy Price



Percentile	Price
0%	-14.6
50%	29.3
90%	50.9
100%	964.8

- Energy can’t be stored → Volatility.

Wholesale Energy Markets



1 Year ~ → Months before ~ → Day-ahead → 1 hour before → Just before delivery

Energy – Financial: 1. Future Market

Energy – Physical: 2. Forward Market; 3. Day-Ahead Market; 4. Real-Time Market

Reliability: 5. regulation up, 6. regulation down, 7. spinning reserve and 8. non-spinning reserve

Transmission: 9. Transmission/Congestion Revenue Market

Capacity: 10. Capacity Market

Environmental: 11. Carbon Allowance, 12. Renewable Credit, etc…

Utility Analytics

Wholesale

- Market Analysis
- Forecasting
  - Power prices
  - Load
  - Generation
- Optimization
  - Resource dispatch
  - Auction strategies
- Risk Management
  - Stochastic portfolio modeling
  - Hedging strategies
- Long Term Resource Planning
  - Investment and divestitures

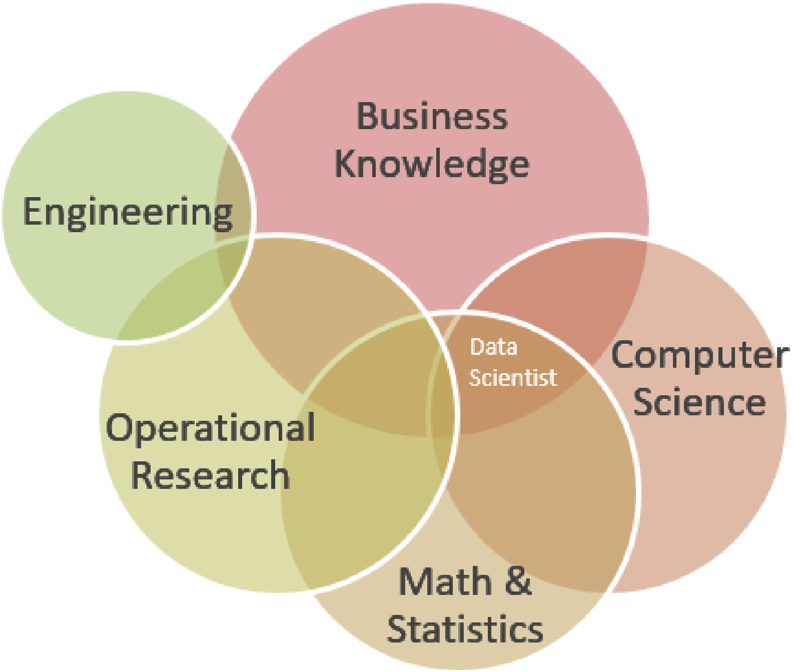
Retail

- Smart meter data analysis
- Distribution system analysis & optimization
- Retail rate analysis
- Customer behavior analysis

And many more…

Utility Analytics

- Team is made up of analysts with different backgrounds.
- Analysts are forced to wear many hats.
- Excel is the industry standard.
- No clear guidelines for analytics quality control.



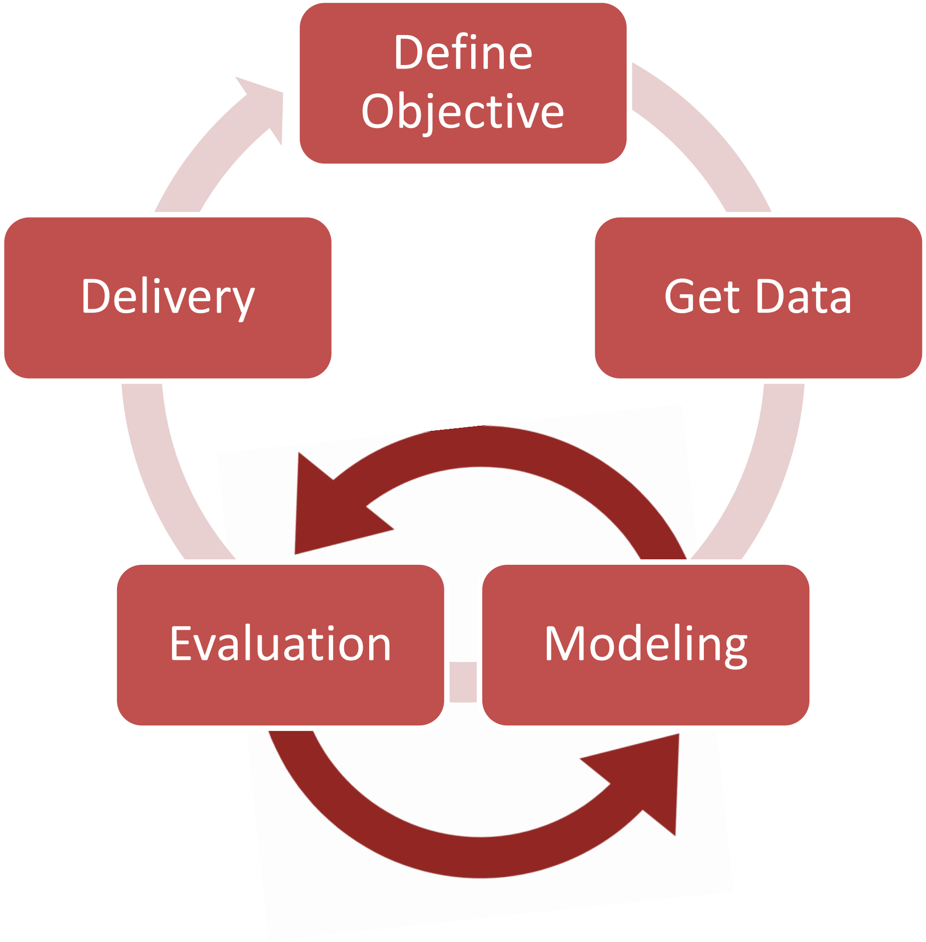
Why Excel? Why Not Excel?

Data Science (Analytics) Lifecycle

Frequency
- One-time study
- recurrent

Format
- csv
- Excel
- Power point
- Documents
- Reports

Interactive (I/O)
- User Interface
- API

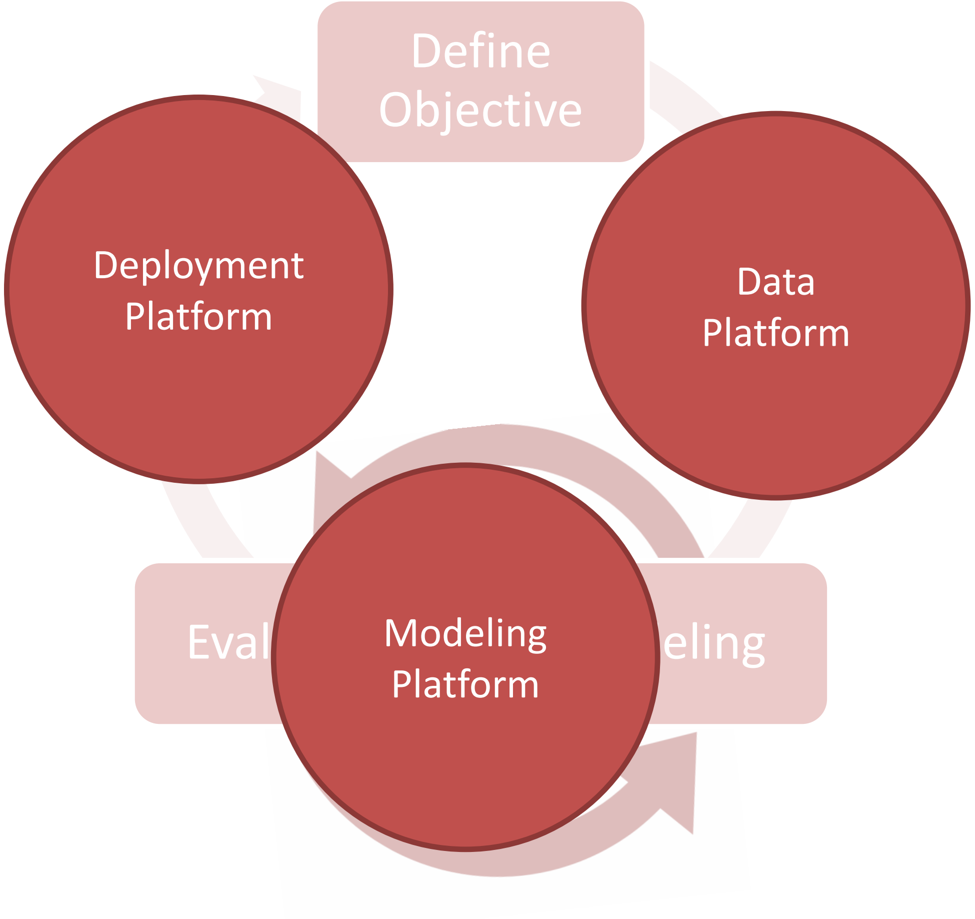


Data Science Platforms

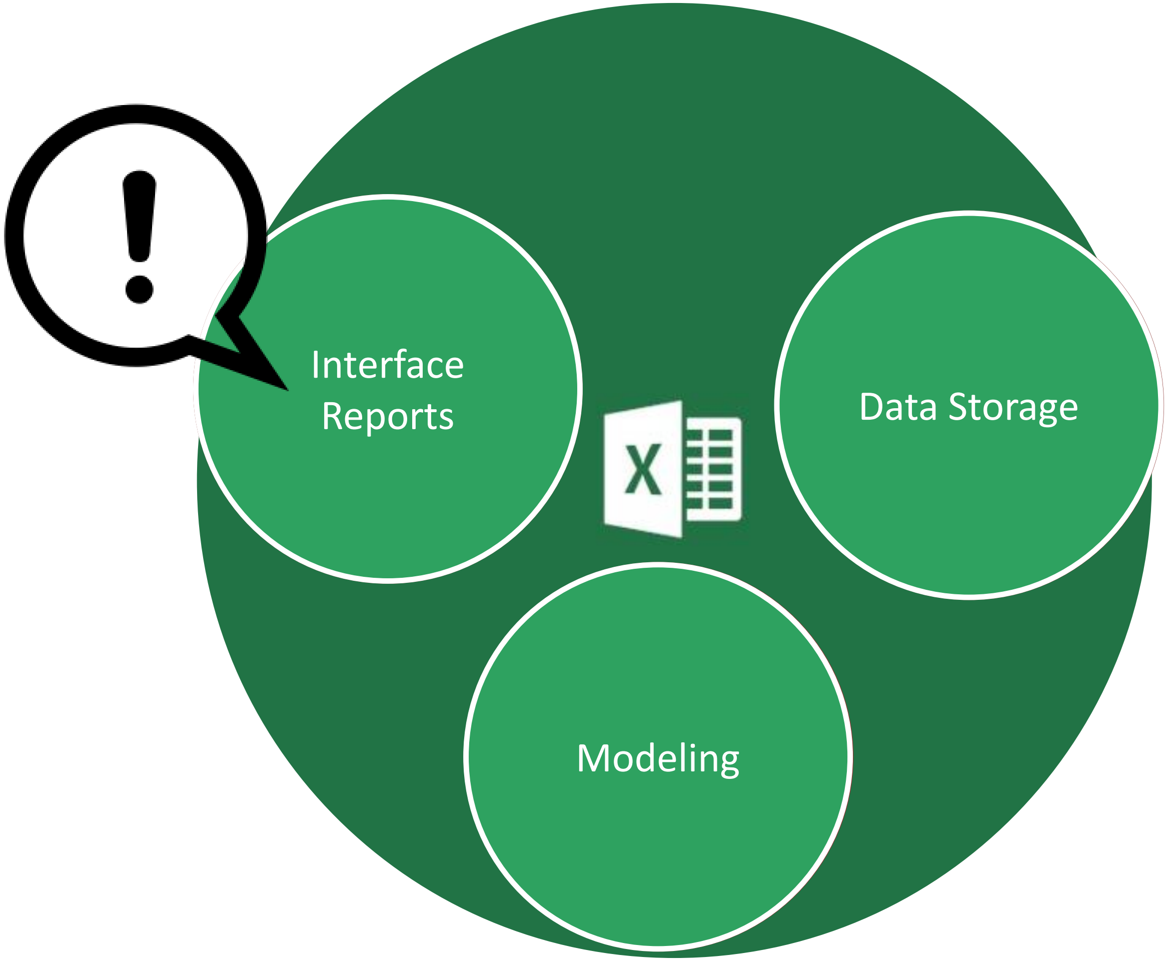
Frequency
- One-time study
- recurrent

Format
- csv
- Excel
- Power point
- Documents
- Reports

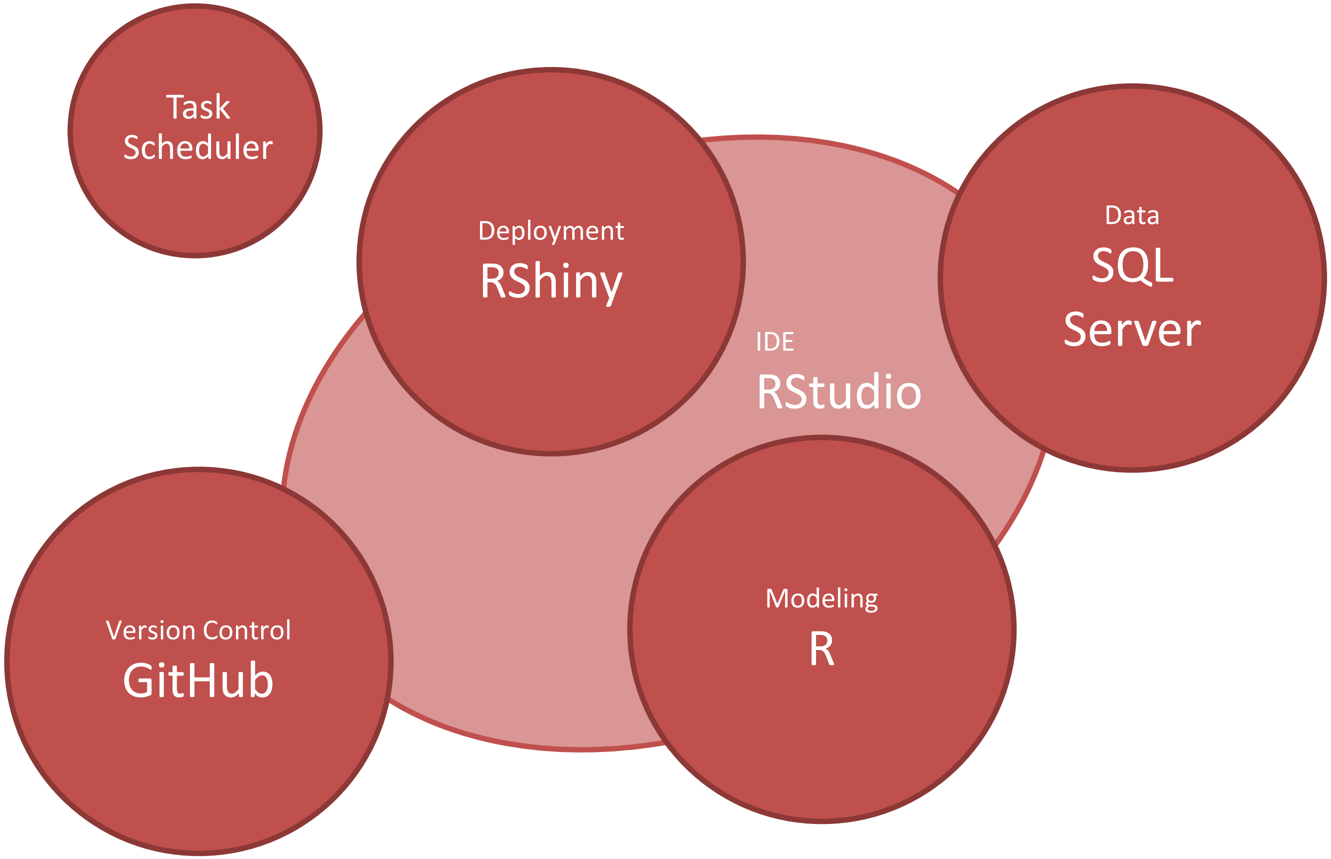
Interactive (I/O)
- User Interface
- API



Excel - Jack of All Trades



Data Science Platforms



Task Scheduler

Deployment
RShiny

IDE
RStudio

Data
SQL Server

Version Control
GitHub

Modeling
R

Analytics Quality Control

Aspirations for Model Design

1. Accurate
   - Which accuracy measures were used?
   - Does it perform better than a naïve model? Which naïve model was used in such a comparison?
2. Robust
   - How are outliers managed?
   - How do outliers affect the results?
3. Parsimonious
   - Can we use a simpler model without sacrificing accuracy?
   - Are the model results better than simpler ones' by a statistically significant margin?

Aspirations for Model Design

4. Explainable
   - Are there variable importance measures for the model?
   - Can we explain the model behavior, locally or globally?
5. Consistency
   - Is the behavior consistent over time?
   - Did backcasting include historical outlier events and how did it behave then?
6. Reproducible
   - How is source code stored and managed?
   - Are necessary input data saved?
   - Are environment information (packages, R, …) saved?

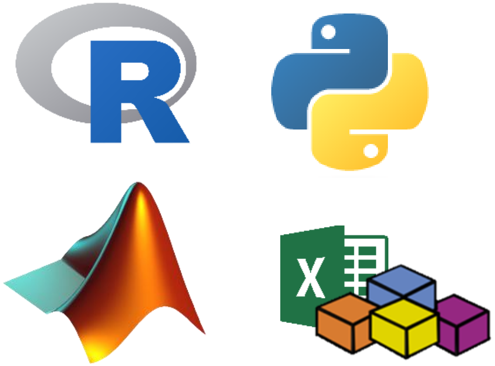
Aspirations for Model Design

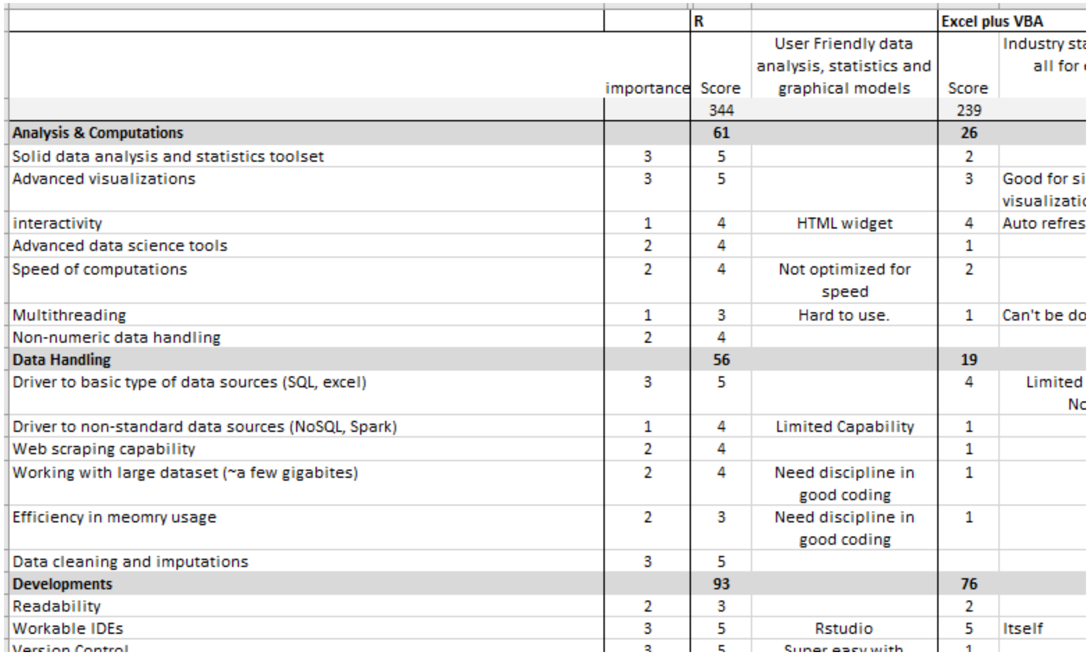
7. Adaptable
   - Did we follow good coding practices, code structures and modularization?
   - Can we use the model or its parts on other projects? Can we generalize them for packaging?
8. Scalable
   - Can we scale up the project to includes many other nodes, customers, etc…?
   - What are limits? When and how would it fail?
9. Accessible
   - Can other people understand and run the model?
   - Is there documentation?

Data Science Platform Evaluations

Modeling Platform

- We evaluated several scripting languages.
- Quantification by scoring was helpful but was not the determining factor.
- Our choice was R.



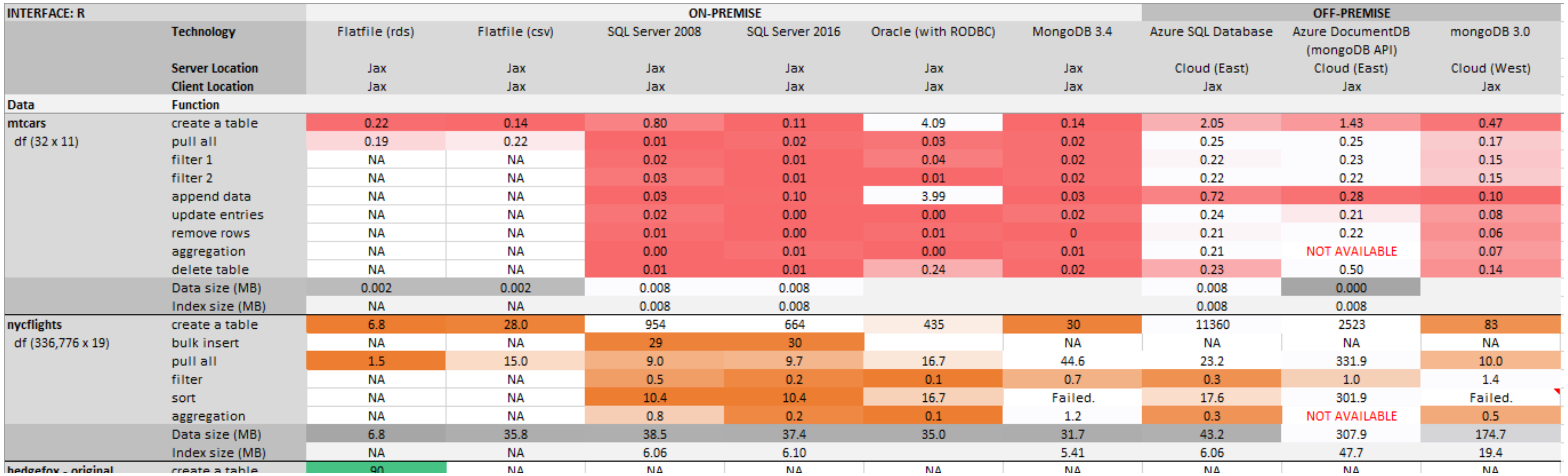


		R		Excel plus VBA	
	importance	Score	User Friendly data analysis, statistics and graphical models	Score	Industry sta all for
		344		239	
Analysis & Computations		**61**		**26**	
Solid data analysis and statistics toolset	3	5		2	
Advanced visualizations	3	5		3	Good for si visualizati
interactivity	1	4	HTML widget	4	Auto refres
Advanced data science tools	2	4		1	
Speed of computations	2	4	Not optimized for speed	2	
Multithreading	1	3	Hard to use.	1	Can't be do
Non-numeric data handling	2	4			
Data Handling		**56**		**19**	
Driver to basic type of data sources (SQL, excel)	3	5		4	Limited Nc
Driver to non-standard data sources (NoSQL, Spark)	1	4	Limited Capability	1	
Web scraping capability	2	4		1	
Working with large dataset (~a few gigabites)	2	4	Need discipline in good coding	1	
Efficiency in meomry usage	2	3	Need discipline in good coding	1	
Data cleaning and imputations	3	5			
Developments		**93**		**76**	
Readability	2	3		2	
Workable IDEs	3	5	Rstudio	5	Itself
Version Control	3	5	Super easy with	1	

Data Platform

Dedicated to Analytics

- We evaluated on-premise and on-cloud databases including types of SQL & NoSQL.
- How does it perform w.r.t. R?
  - DBI compliant? Flexibility of NoSQL?
- Our choice was SQL Server.



INTERFACE: R		ON-PREMISE						OFF-PREMISE		
	Technology	Flatfile (rds)	Flatfile (csv)	SQL Server 2008	SQL Server 2016	Oracle (with RODBC)	MongoDB 3.4	Azure SQL Database	Azure DocumentDB (mongoDB API)	mongoDB 3.0
	Server Location	Jax	Jax	Jax	Jax	Jax	Jax	Cloud (East)	Cloud (East)	Cloud (West)
	Client Location	Jax	Jax	Jax	Jax	Jax	Jax	Jax	Jax	Jax
Data	Function									
mtcars	create a table	0.22	0.14	0.80	0.11	4.09	0.14	2.05	1.43	0.47
df (32 x 11)	pull all	0.19	0.22	0.01	0.02	0.03	0.02	0.25	0.25	0.17
	filter 1	NA	NA	0.02	0.01	0.04	0.02	0.22	0.23	0.15
	filter 2	NA	NA	0.03	0.01	0.01	0.02	0.22	0.22	0.15
	append data	NA	NA	0.03	0.10	3.99	0.03	0.72	0.28	0.10
	update entries	NA	NA	0.02	0.00	0.00	0.02	0.24	0.21	0.08
	remove rows	NA	NA	0.01	0.00	0.01	0	0.21	0.22	0.06
	aggregation	NA	NA	0.00	0.01	0.00	0.01	0.21	NOT AVAILABLE	0.07
	delete table	NA	NA	0.01	0.01	0.24	0.02	0.23	0.50	0.14
	Data size (MB)	0.002	0.002	0.008	0.008			0.008	0.000	
	Index size (MB)	NA	NA	0.008	0.008			0.008	0.008	
nycflights	create a table	6.8	28.0	954	664	435	30	11360	2523	83
df (336,776 x 19)	bulk insert	NA	NA	29	30		NA	NA	NA	NA
	pull all	1.5	15.0	9.0	9.7	16.7	44.6	23.2	331.9	10.0
	filter	NA	NA	0.5	0.2	0.1	0.7	0.3	1.0	1.4
	sort	NA	NA	10.4	10.4	16.7	Failed.	17.6	301.9	Failed.
	aggregation	NA	NA	0.8	0.2	0.1	1.2	0.3	NOT AVAILABLE	0.5
	Data size (MB)	6.8	35.8	38.5	37.4	35.0	31.7	43.2	307.9	174.7
	Index size (MB)	NA	NA	6.06	6.10		5.41	6.06	47.7	19.4
hedgefox - original	create a table	90	NA	NA	NA	NA	NA	NA	NA	NA

Delivery Format

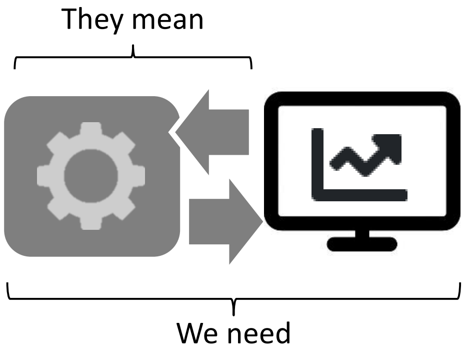
- If we simply deliver flat files or reports, no need for deployment platform.
  - R-markdown has made R reporting extremely easy and visually appealing.
  - There are numerous r packages that enable saving results in Excel.

- Deployment platform is necessary when a deliverable is a model with inputs/outputs.

Data Science Deployment

- When non-utility data scientists say “deployment,” they usually mean:
  1. The model is turned over to software developers to translate into a production language.
     → We don’t have software developers to spare…
  2. Deploy models on their own as APIs.
     → We don’t have web developers to spare…

- How do we create a user interface for our models without going back to Excel or relying on IT developers?



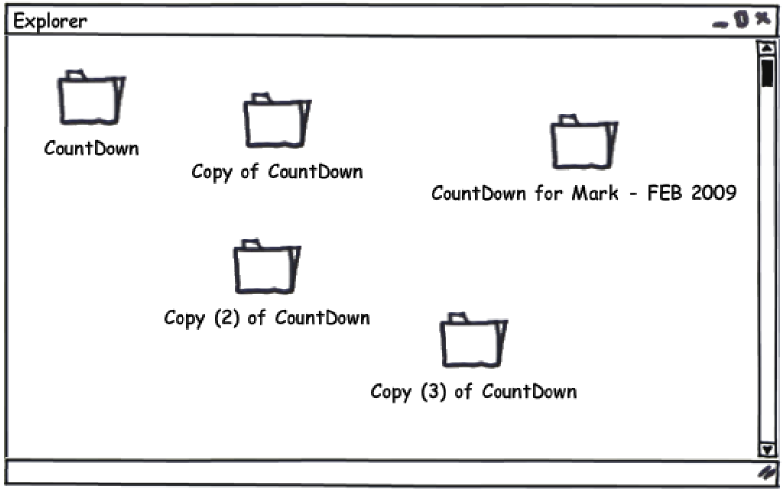
Deployment Platform

- We evaluated
  - Various “Data Science Platforms.”
    - There were fewer products to evaluate in 2015.
  - Various BI tools
  - R Shiny Server

- We chose R Shiny Server (Pro).

Source Control

- Git for
  - Source Control
- GitHub for
  - Managing Repositories
  - Project Management
  - Collaboration
  - Issue tracker

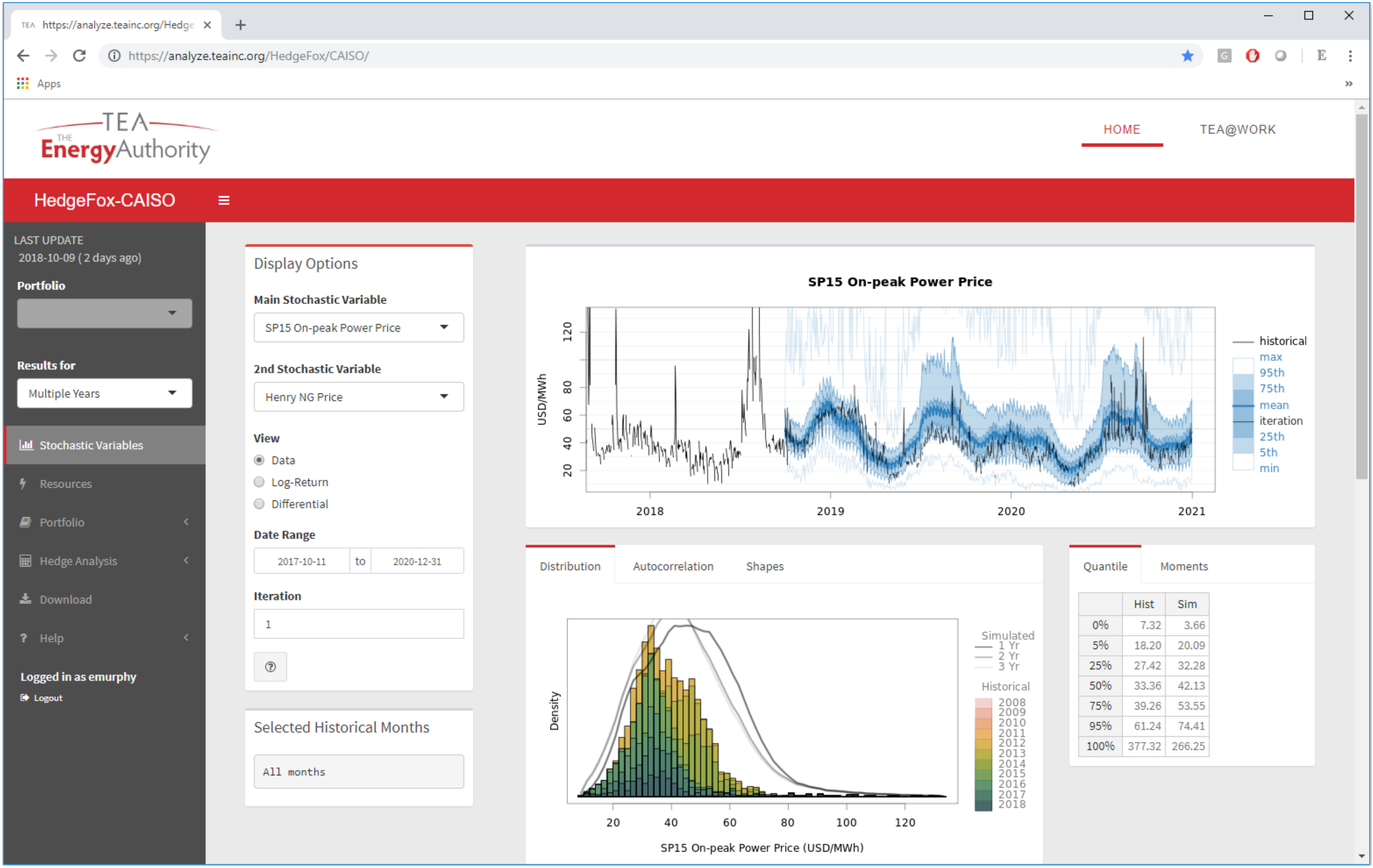


- Use of Git, together with Shiny was a game changer for us.
  - Eliminated accumulation of similarly named files or commented-out old scripts.
  - Easy differentiation between multiple (Dev, PROD, …) environments.

Other Supporting Tools

- RStudio IDE
- Task Scheduler with Batch Tasks
- Packages
  - CRAN
  - Custom

Shiny
R Studio
ggplot2
dplyr
rmarkdown
knitr
lubridate
stringr
devtools
tidyr
%>%
Ceci n'est pas un pipe.
packrat
ggvis
readr



LAST UPDATE
2018-10-09 (2 days ago)

Portfolio

Results for: Multiple Years

Logged in as emurphy

Display Options

Main Stochastic Variable: SP15 On-peak Power Price

2nd Stochastic Variable: Henry NG Price

View
- ● Data
- ○ Log-Return
- ○ Differential

Date Range: 2017-10-11 to 2020-12-31

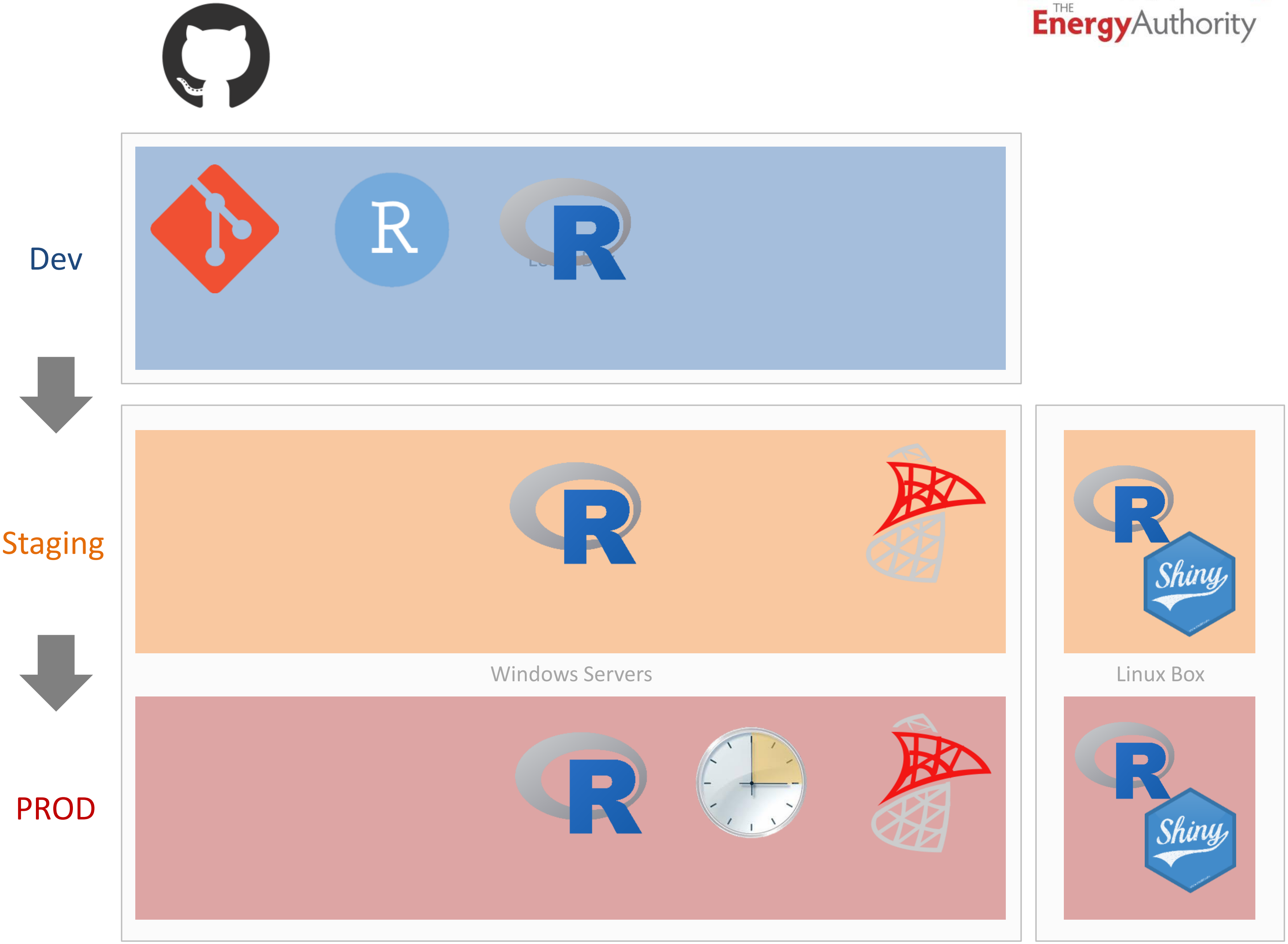
Iteration: 1

Selected Historical Months

All months

	Hist	Sim
0%	7.32	3.66
5%	18.20	20.09
25%	27.42	32.28
50%	33.36	42.13
75%	39.26	53.55
95%	61.24	74.41
100%	377.32	266.25

New Analytics Workflow



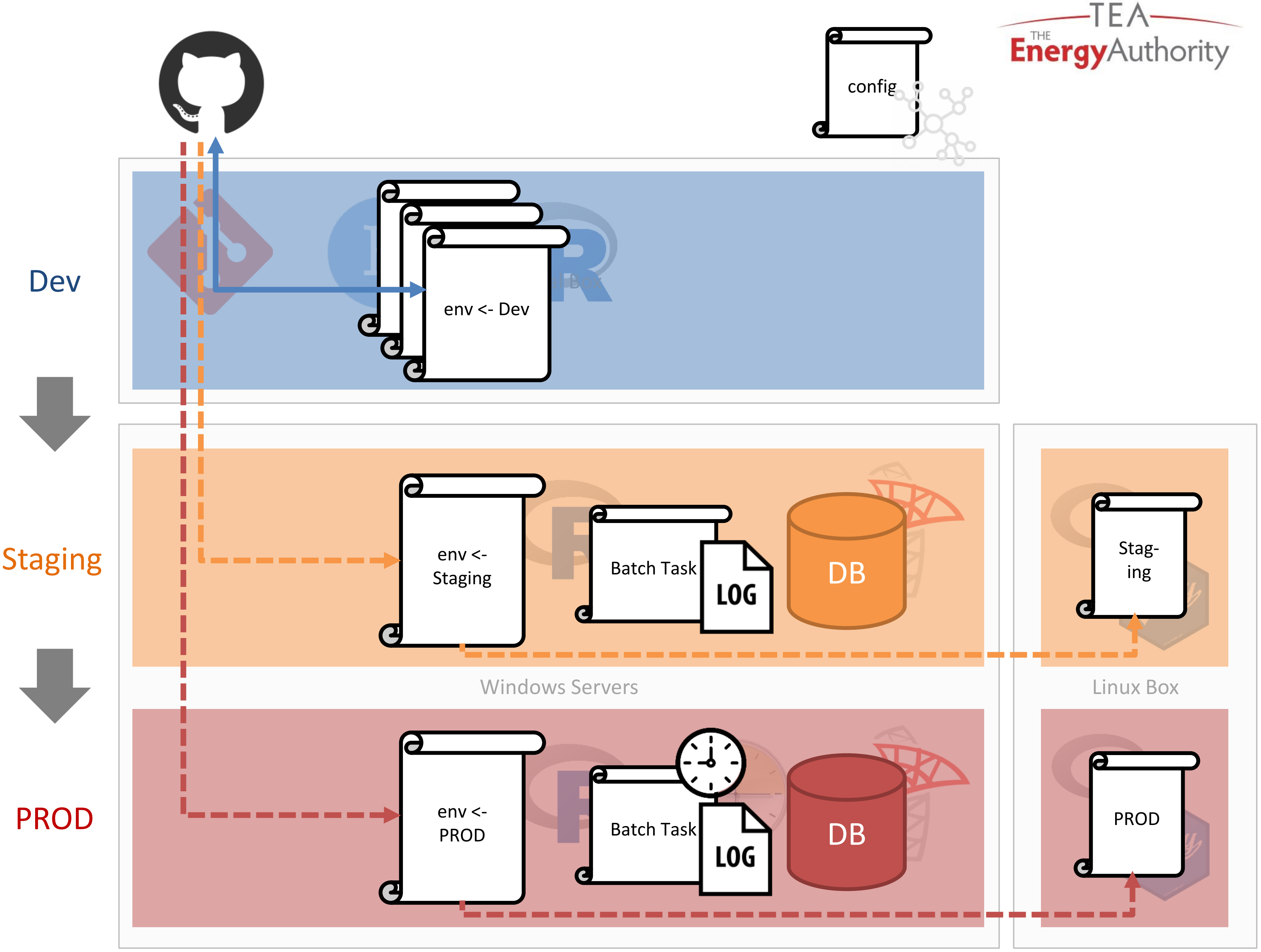
Dev

Staging

PROD

Windows Servers

Linux Box


config
Dev
env <- Dev
Staging
env <- Staging
Batch Task
LOG
DB
Stag-ing
Windows Servers
Linux Box
PROD
env <- PROD
Batch Task
LOG
DB
PROD

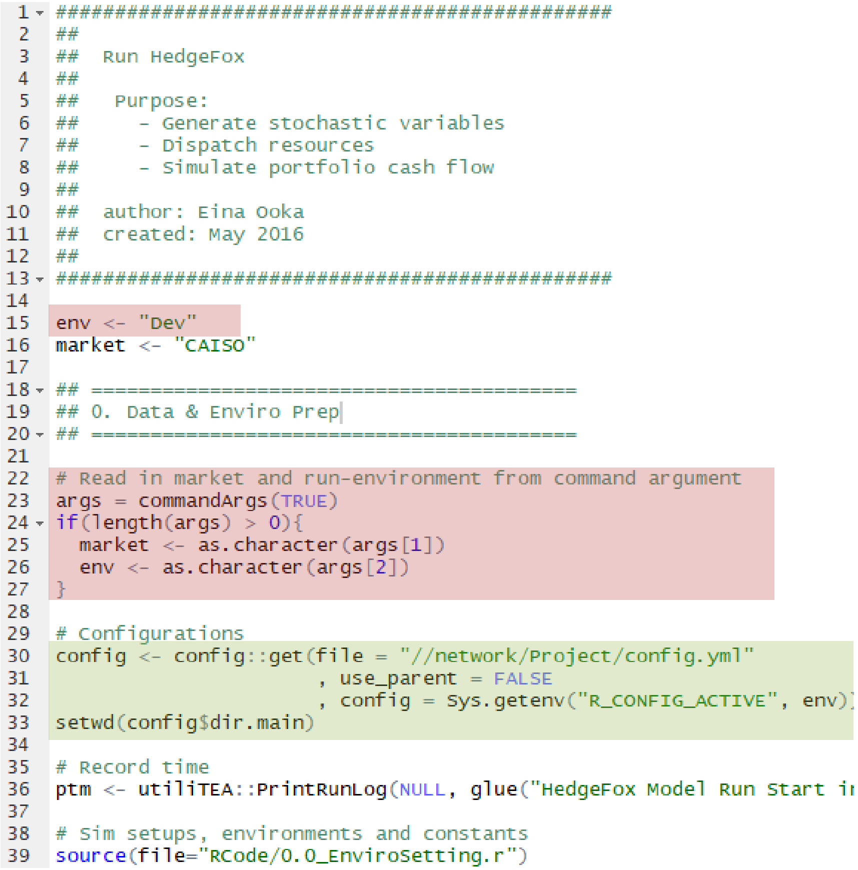
Environment Settings

Two Ways:

1. Says it in the script.
2. Read from a batch file as an argument.

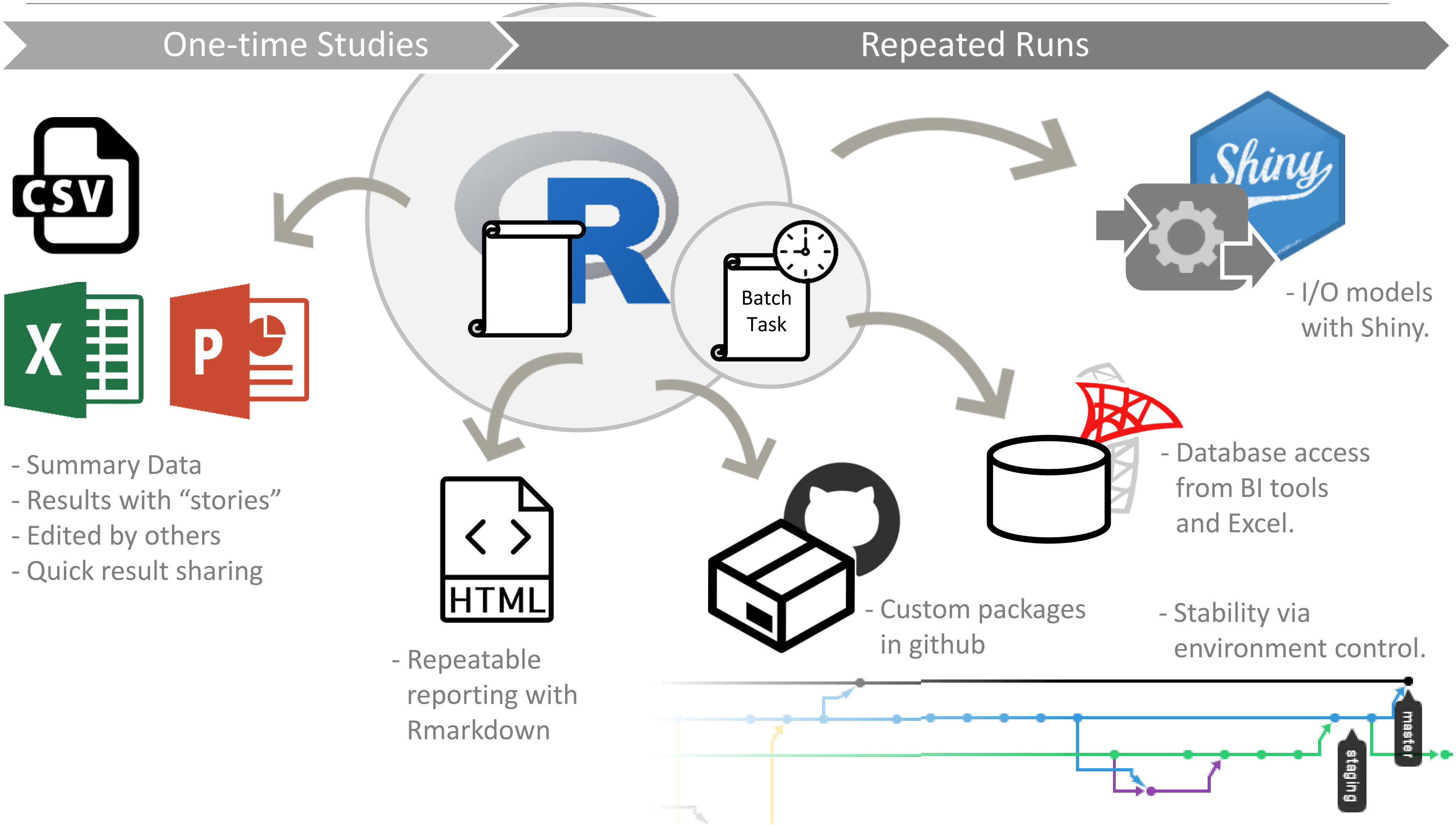
The rest is set by a config file

1. Directories & folders
2. Database
3. Username/Password



```r
###############################################
##
##  Run HedgeFox
##
##   Purpose:
##     - Generate stochastic variables
##     - Dispatch resources
##     - Simulate portfolio cash flow
##
##  author: Eina Ooka
##  created: May 2016
##
###############################################

env <- "Dev"
market <- "CAISO"

## =========================================
## 0. Data & Enviro Prep
## =========================================

# Read in market and run-environment from command argument
args = commandArgs(TRUE)
if(length(args) > 0){
  market <- as.character(args[1])
  env <- as.character(args[2])
}

# Configurations
config <- config::get(file = "//network/Project/config.yml"
                      , use_parent = FALSE
                      , config = Sys.getenv("R_CONFIG_ACTIVE", env)
setwd(config$dir.main)

# Record time
ptm <- utiliTEA::PrintRunLog(NULL, glue("HedgeFox Model Run Start i

# Sim setups, environments and constants
source(file="RCode/0.0_EnviroSetting.r")
```

Summary of Delivery Methods



One-time Studies

Repeated Runs

- Summary Data
- Results with “stories”
- Edited by others
- Quick result sharing

- Repeatable reporting with Rmarkdown

- Custom packages in github

- I/O models with Shiny.

- Database access from BI tools and Excel.

- Stability via environment control.

Conclusions

- Excel is still a useful tool. Not all projects need to or can move away from it.
- We need multiple platforms that all work seamlessly together.
- Choosing the right set of platforms can help broaden our capability and achieve quality control.
- Watch for new products and packages.

Thank you!

Contact:
eooka@teainc.org