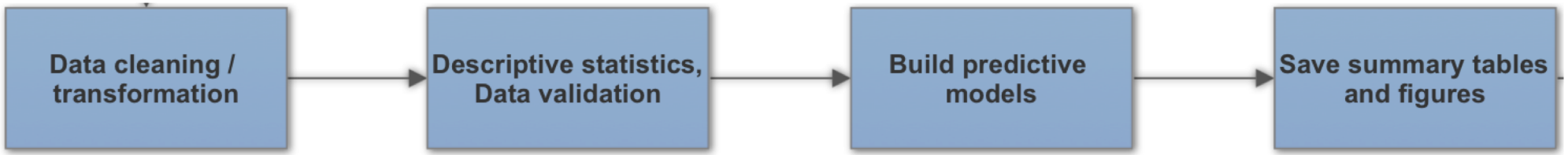
EARL US Roadshow | November, 2018

R-bots for Data Science Workflow Automation

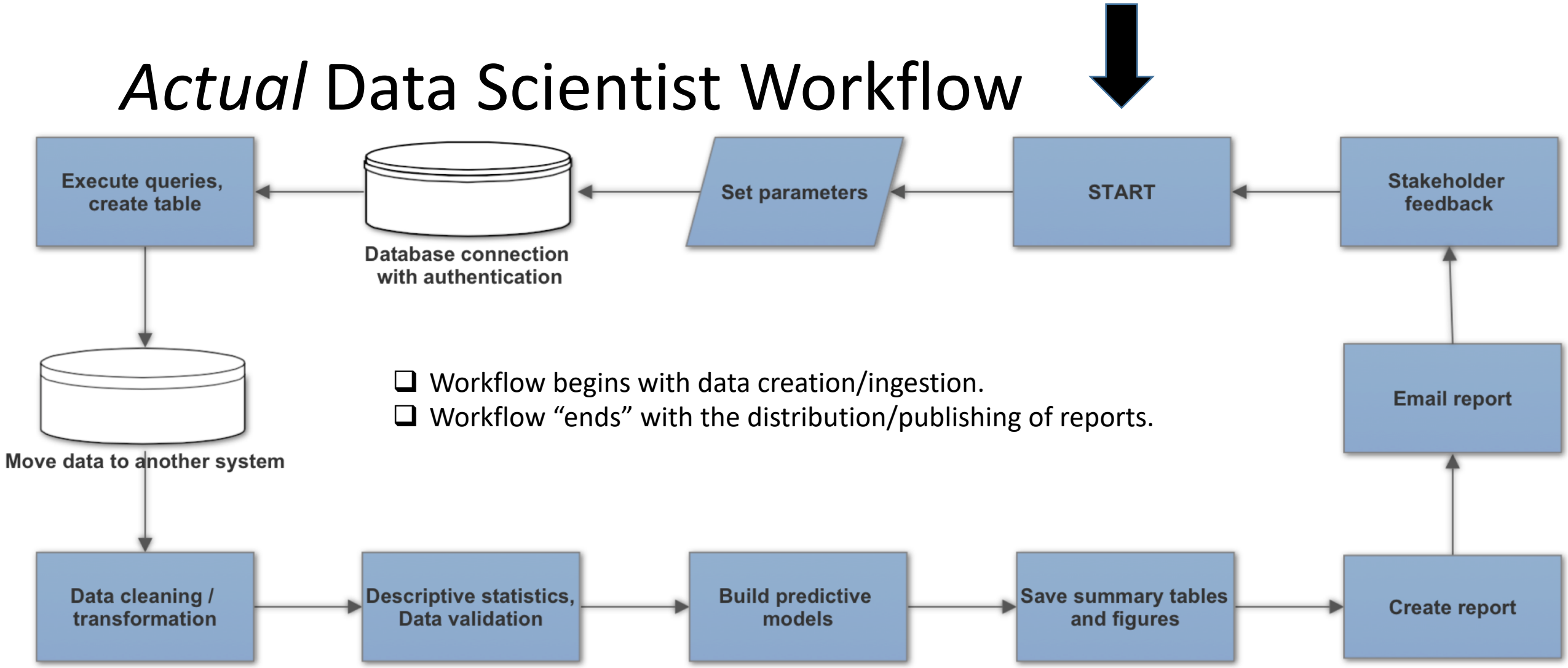
Sydeaka Watson, Ph.D.

Senior Data Scientist, AT&T Chief Data Office

Data Scientist Workflow



Actual Data Scientist Workflow



- Workflow begins with data creation/ingestion.
- Workflow "ends" with the distribution/publishing of reports.

Automated Data Science Workflows

- In an automated data science workflow, we link together a series of scripts that *could* be run without user intervention.
- An ***R*-Bot** is an R script that executes one or more tasks in a data science workflow.
- Examples:
  - Data ingestion
  - Create report with dynamically generated content
  - Send emails
  - Version control

Use Cases

- Well-defined analyses that will be performed over and over again
- Analyses in development stages, where feedback must be incorporated over and over again
- “Hands-free” automation
  - Task scheduler manages the runs
- “Manual” automation
  - Entire process kicks off as soon as the data scientist presses the button

Benefits

- Reproducible research
- Version control over entire process
- Less time spent on mundane, repetitive tasks
- Easier to get results reports generated and distributed to stakeholders

Workflow automation framework

- Adopt modularized approach, with flexibility to swap out components
- Use common framework from which all scripts are executed (e.g., R, Bash)
- Set parameters (manually or dynamically)
- Re-use code (functions)
- Write code that creates/executes dynamically generated code
- Relinquish low level decisions to computer
  - Use conditional logic
  - Error handling
  - Automated validation of results

Tools/Utilities

Task	R Packages	Bash functions
Connect to SQL database, with authentication username/password; execute queries	`RJDBC`, `ODBC`	--
Connect to Hadoop, with Kerberos authentication	`krb5`	`kinit` using keytab credentials file
Create/populate Excel spreadsheet	`XLConnect`	--
Generate reports (documents, html website)	`rmarkdown`, `knitr`	--
Email notifications, reports	`mailR`, `sendmailR` ** These don't always work behind firewalls with restricted network settings.	`mailx`
Schedule task to run at regular intervals	`cronR`	`crontab`

Enterprise considerations

- In enterprise setting, you may have limitations on which types of packages you can install on the server, so you may be limited to mostly base R or system commands.

- In the demo, I give an example of a workflow created under this scenario.

DEMO

Example is available in my public repository:

https://github.com/sydeaka/workflow-automation

DEMO: Instructions

- System requirements
  - Mac / Linux operating system
  - MYSQL (optional)
    - If you have MYSQL installed, set use_mysql=TRUE and create a credentials file in your system's home directory. The file should contain two lines:

```
username=YourUserName
password=YourPassword
```

    - If not, set use_mysql=FALSE to run without MYSQL
- To run
  - navigate to the top-level directory in the repository in your terminal
  - modify working directory at top of "mysql_example.sh" file
  - modify from/to email addresses in email.R, and create "gmail.txt" file in system home directory (file just contains your password as raw text, no quotes)
  - execute the following:

```
./mysql_example.sh
```

DEMO: Github directory structure

Folder	Subfolder	Contents
config		Session parameters, pointer to credential file location
data	bash	Bash scripts
	dictionary	Lending Club data dictionary
	downloads	Downloaded zip files from Lending Club website + unzipped CSVs
	join_data	State-level datasets to be joined to Lending Club dataset
	modeling_data	Cleaned/transformed Lending Club data subset prepared for modeling
	sql	SQL scripts created/executed in the process
model_results		h2o model files + R workspace with objects saved for use in report
plots		Plots to be rendered in the report
reports		Rmarkdown files + rendered reports (HTML/Markdown formats)
utils		R scripts

DEMO: Skeleton of `mysql_example.sh`

1. Read in MYSQL credentials and set session parameters
2. Download dataset for selected year/quarter if it does not exist
3. Generate code for ingestion of entire dataset into MYSQL database + join with existing dataset
4. Execute code created in #3
5. Generate code to create new MYSQL table with selected attributes for modeling
6. Execute code created in #5
7. Pull MYSQL modeling table into an R session, apply transformations
8. Build predictive models and save results
9. Create report of descriptive stats + modeling results
10. Email report
11. Check workflow code into Github repository

Other considerations

- Software, package, platform dependencies (Ex. Spark 2.0 vs 2.2; h2o 3.14 vs 3.18); solution: environments
- Storage space and memory requirements
- Scale: if dataset gets larger over time, the solution might not work as well (scale up). May need to adjust framework on the front end or do it later. Monitor strain on resource demands, run time, and other metrics
- Tech dev requirements if you want to move process to production
- Documentation of the workflow
- Redundancy: make sure more than one person understands the process well and can help maintain it

THANKS!