Robust Data Pipelines with drake and Docker
The end goal for most data analytics projects is to build a data product, whether it is a weekly dashboard or deploying a ML model.

From getting and cleaning the data to generating plots or fitting models, each project can be split up into individual tasks.

Afterwards, we want to make sure that the environment we wrote our code in can be replicated.
Automation.
Reproducibility.
Workflow management tool that enables you to build managed workflows. It is a first serious attempt to create a tool like Airflow or Luigi, but in R.
Your analysis is a sequence of transformations.
write a plan.

A plan can be simple, or very complex. It all depends on how you define the tasks’ dependencies.
Not only targets are kept track of.

By default we can visualise all imports, functions and transformations in our DAG, not only completed tasks. As a nice bonus we can also monitor progress for longer jobs.
Checks dependencies and cache before creating plan. This means that on subsequent runs, only the changed tasks will rerun, leaving the rest intact.
A container is kinda like a VM but... **different.**
benefits of containers

- Easily reproduce your infrastructure
- Runs independent of host OS
- Consistency in production
step 1: Dockerfile

FROM rocker/r-base

## install XML dependency
RUN apt-get update && apt-get -y install libxml2-dev

## install R packages from CRAN
RUN install2.r -e dplyr drake

## Put script inside container
COPY ./run_workflow.R /

## Make scripts executable
RUN chmod +x /run_workflow.R

## Run workflow
CMD ['Rscript', 'run_workflow.R']
step 2: build image

This is the stage where we actually build our mini computer, ready for deployment.

```
Tamass-MacBook-Air:eRum tamas$ docker build -t iris .
Sending build context to Docker daemon 17.35MB
Step 1/6 : FROM rocker/r-base
  ---> cc7da2b50c3f
Step 2/6 : RUN apt-get update && apt-get -y install libxml2-dev
  ---> Using cache
  ---> 08820573d297
Step 3/6 : RUN install2.r --e dplyr drake
  ---> Using cache
  ---> bebc0073935c
Step 4/6 : COPY ./run_workflow.R /
  ---> Using cache
  ---> 99fc09fe6e26
Step 5/6 : RUN chmod +x /run_workflow.R
  ---> Using cache
  ---> 232d5b0bc0c4
Step 6/6 : CMD ["Rscript", "run_workflow.R"]
  ---> Running in f6f6deced2aa
Removing intermediate container f6f6deced2aa
  ---> 59cae4a54a8b
Successfully built 59cae4a54a8b
Successfully tagged iris:latest
```
step 3: run container

Instantiates the container, mounts the folder from our host where the data resides and runs our executable as defined in the Dockerfile.

docker run -d \
  -v /path_to/data/::/data/ \
  --name erum iris:latest
Find me

http://tamaszilagyi.com/
@tudosgar
Thank you.