Planbureau voor de Leefomgeving

PEARL in de cloud

Trond Grytli Husby

Netherlands Environmental Assessment Agency (PBL)

trond.husby@pbl.nl

Den Haag 27 June, 2019

Regional population projections of PBL and CBS

- Every three years the PBL and CBS make the Regional (and National) Population Projections
- Projections of population, households and demographic events in Dutch municipalities until 2050
- Previous edition was in 2016, next in September 2019

Source: https://img.clipartxtras.com/03843abee543e741be870c4ada22c760_free-to-use-public-domain-cauldron-clip-art-witch-cauldron-clipart_500-500.png

Cohort-component model

$$P_{t+1} = P_t + B - X + I - E + A - D$$

P_t: population in t

B: births in the interval $(t, t+1)$

X: deaths in the interval $(t, t+1)$

I: immigration in the interval $(t, t+1)$

E: emigration in the interval $(t, t+1)$

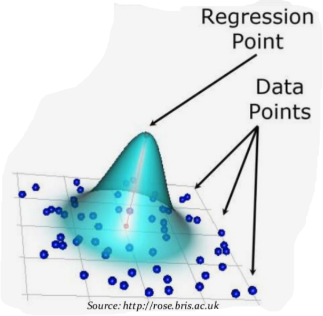
A: arrivals in the interval $(t, t+1)$

D: departures in the interval $(t, t+1)$

- PEARL is a book-keeping system, per definition no uncertainty (unless coding errors)
- But considerable uncertainty in the (exogenous) time-path of growth components
- We can obtain a consistent estimate of the projection uncertainty with Monte Carlo Simulation
- An attempt to carry out a fully stochastic projection stranded on computing power

Azure case 1: estimation of short-distance migration model [1]

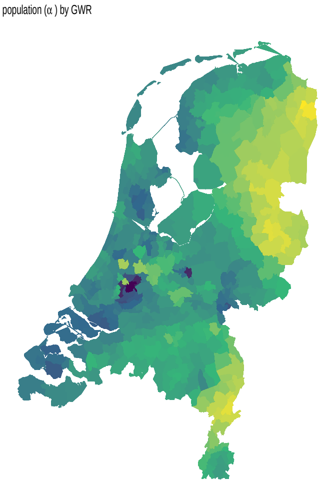
- Short-distance migration in PEARL governed by estimated parameters
- Gravity model, estimated with geographically weighted Poisson regression (maximum likelihood)

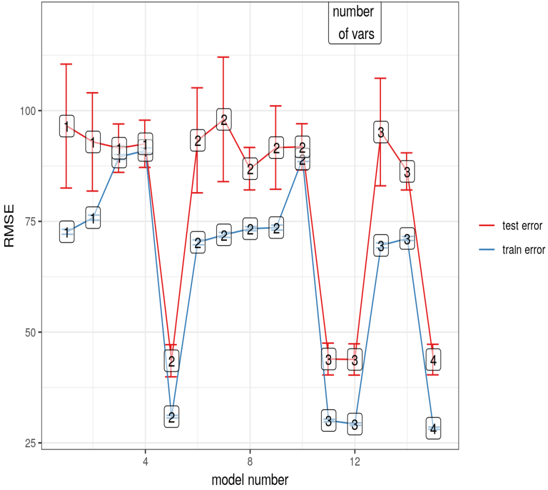


Source: http://rose.bris.ac.uk

[1] https://github.com/trondhusby/GWmodel

Results


population (α) by GWR


number
of vars
RMSE
100
75
50
25
4
8
12
model number
test error
train error
1
2
3
4

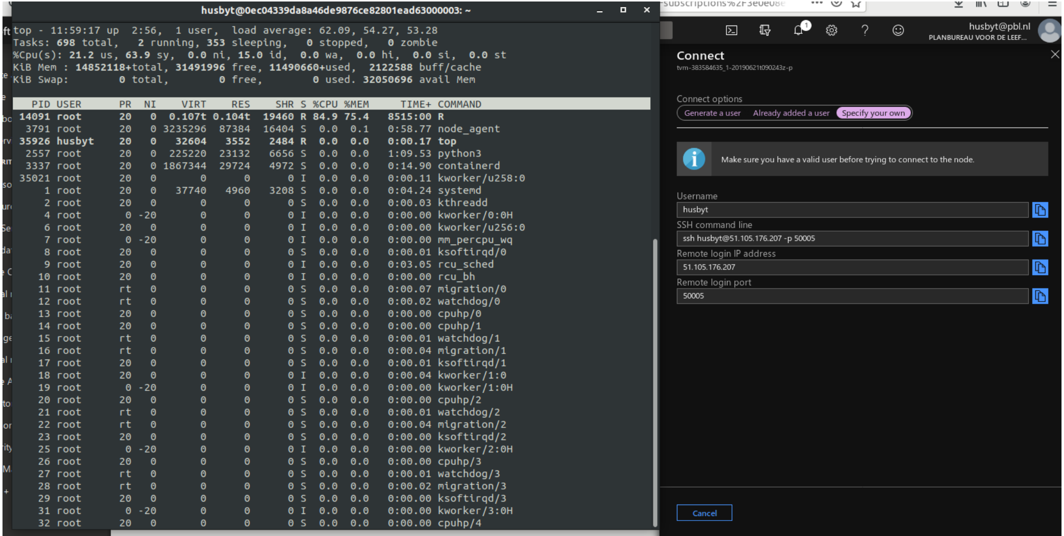
Why Azure?

- GWR: one model per regression point (16000) times number of MLE iterations. The entire run typically $\tilde{4}$ hours
- Kernel type selection: 4 different kernel types, OLS and MLE (8 models)
- Variable selection: 10-fold cross validation of MLE for all permutations of 4 variables (15 models times 10)
- Bandwidth selection: 10-fold cross validation of MLE for four models from previous step for 7 bandwiths (28 models times 10)

Azure and R

- R package *rAzureBatch*: interaction with the Azure Batch service's REST API
- R package *doAzureParallel*: parallel backend for the *foreach* package.
- `out <- foreach(i=1:n, .options) %dopar% {func(i)}`
- 12 - 15 F64s v2 / F72s v2 Linux VMs in parallel
- Copy of the user's existing R environment stored in Azure Storage. Each VM fetches and loads the R environment. The VM runs the R code inside each iteration of the foreach loop under the loaded R environment. Once the code is finished, results are pushed back to Azure Storage, and a merge task is used to aggregate the results.
- Note: high-level parallelisation in R not always efficient. Need to check whether low-level parallelisation via multicore BLAS/LAPACK is in place

Run-time access to node via ssh



husbyt@0ec04339da8a46de9876ce82801ead63000003: ~

```
top - 11:59:17 up  2:56,  1 user,  load average: 62.09, 54.27, 53.28
Tasks: 698 total,   2 running, 353 sleeping,   0 stopped,   0 zombie
%Cpu(s): 21.2 us, 63.9 sy,  0.0 ni, 15.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 14852118+total, 31491996 free, 11490660+used,  2122588 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 32050696 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
14091 root      20   0  0.107t 0.104t  19460 R 84.9 75.4   8515:00 R
 3791 root      20   0 3235296  87384  16404 S  0.0  0.1   0:58.77 node_agent
35926 husbyt    20   0   32604   3552   2484 R  0.0  0.0   0:00.17 top
 2557 root      20   0  225220  23132   6656 S  0.0  0.0   1:09.53 python3
 3337 root      20   0 1867344  29724   4972 S  0.0  0.0   0:14.90 containerd
35021 root      20   0       0      0      0 I  0.0  0.0   0:00.11 kworker/u258:0
    1 root      20   0   37740   4960   3208 S  0.0  0.0   0:04.24 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.03 kthreadd
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 I  0.0  0.0   0:00.00 kworker/u256:0
    7 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.01 ksoftirqd/0
    9 root      20   0       0      0      0 I  0.0  0.0   0:03.05 rcu_sched
   10 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_bh
   11 root      rt   0       0      0      0 S  0.0  0.0   0:00.07 migration/0
   12 root      rt   0       0      0      0 S  0.0  0.0   0:00.02 watchdog/0
   13 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/0
   14 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/1
   15 root      rt   0       0      0      0 S  0.0  0.0   0:00.01 watchdog/1
   16 root      rt   0       0      0      0 S  0.0  0.0   0:00.04 migration/1
   17 root      20   0       0      0      0 S  0.0  0.0   0:00.01 ksoftirqd/1
   18 root      20   0       0      0      0 I  0.0  0.0   0:00.04 kworker/1:0
   19 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/1:0H
   20 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/2
   21 root      rt   0       0      0      0 S  0.0  0.0   0:00.01 watchdog/2
   22 root      rt   0       0      0      0 S  0.0  0.0   0:00.04 migration/2
   23 root      20   0       0      0      0 S  0.0  0.0   0:00.00 ksoftirqd/2
   25 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/2:0H
   26 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/3
   27 root      rt   0       0      0      0 S  0.0  0.0   0:00.01 watchdog/3
   28 root      rt   0       0      0      0 S  0.0  0.0   0:00.02 migration/3
   29 root      20   0       0      0      0 S  0.0  0.0   0:00.00 ksoftirqd/3
   31 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/3:0H
   32 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/4
```

husbyt@pbl.nl
PLANBUREAU VOOR DE LEEF...

Connect

tvm-383584635_1-20190621t090243z-p

Connect options: Generate a user | Already added a user | [✓ Specify your own]

Make sure you have a valid user before trying to connect to the node.

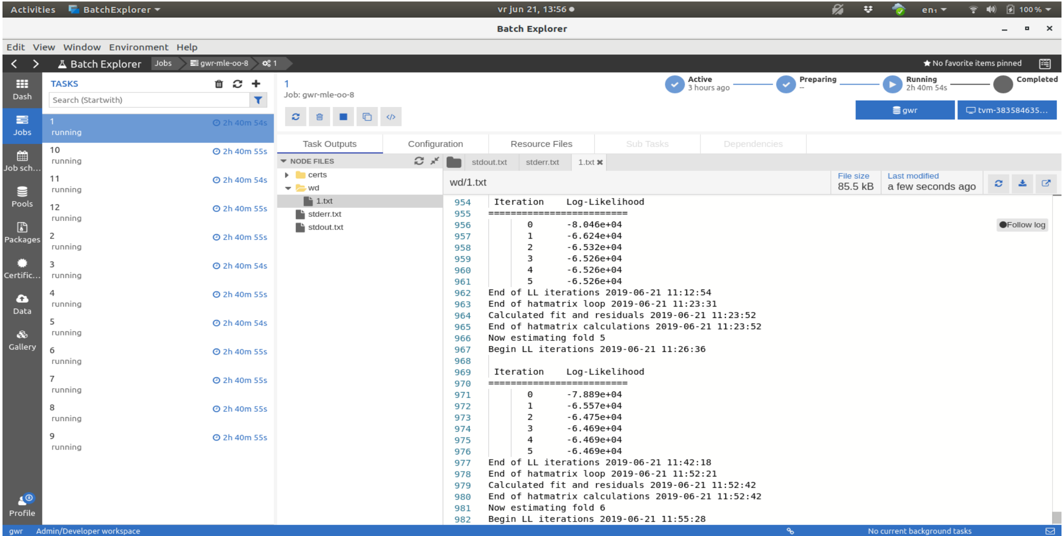
Username: husbyt

SSH command line: ssh husbyt@51.105.176.207 -p 50005

Remote login IP address: 51.105.176.207

Remote login port: 50005

Run-time access to log via Batch Explorer



1	running	2h 40m 54s
10	running	2h 40m 55s
11	running	2h 40m 54s
12	running	2h 40m 55s
2	running	2h 40m 55s
3	running	2h 40m 54s
4	running	2h 40m 55s
5	running	2h 40m 54s
6	running	2h 40m 55s
7	running	2h 40m 55s
8	running	2h 40m 55s
9	running	2h 40m 55s

1

Job: gwr-mle-oo-8

wd/1.txt

File size 85.5 kB

Last modified a few seconds ago

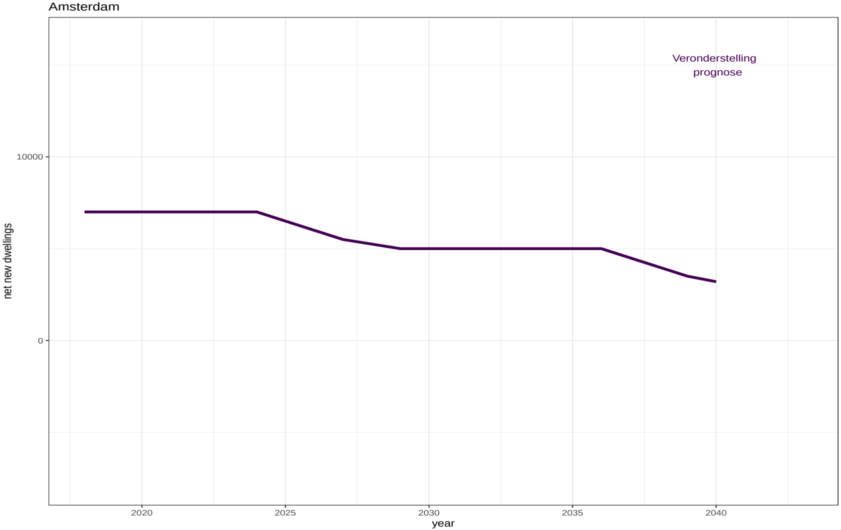
```
 Iteration     Log-Likelihood
=========================
         0      -8.046e+04
         1      -6.624e+04
         2      -6.532e+04
         3      -6.526e+04
         4      -6.526e+04
         5      -6.526e+04
End of LL iterations 2019-06-21 11:12:54
End of hatmatrix loop 2019-06-21 11:23:31
Calculated fit and residuals 2019-06-21 11:23:52
End of hatmatrix calculations 2019-06-21 11:23:52
Now estimating fold 5
Begin LL iterations 2019-06-21 11:26:36

 Iteration     Log-Likelihood
=========================
         0      -7.889e+04
         1      -6.557e+04
         2      -6.475e+04
         3      -6.469e+04
         4      -6.469e+04
         5      -6.469e+04
End of LL iterations 2019-06-21 11:42:18
End of hatmatrix loop 2019-06-21 11:52:21
Calculated fit and residuals 2019-06-21 11:52:42
End of hatmatrix calculations 2019-06-21 11:52:42
Now estimating fold 6
Begin LL iterations 2019-06-21 11:55:28
```

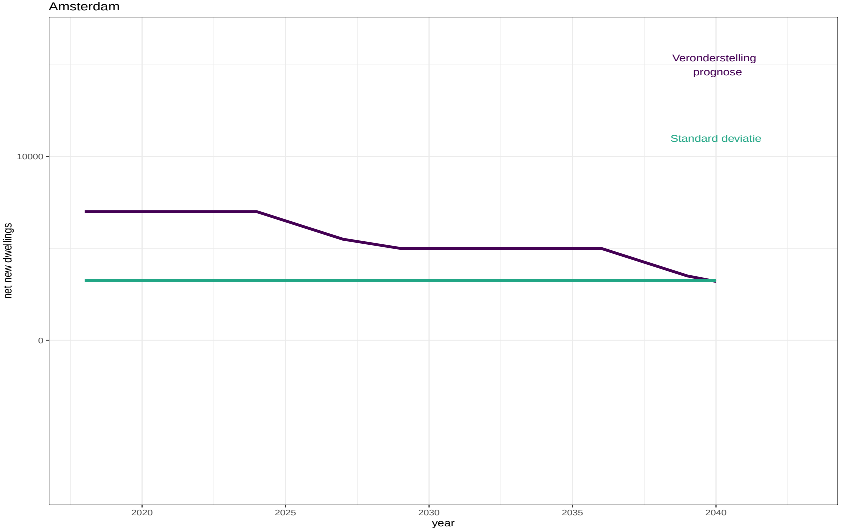
Azure case 2: Monte Carlo simulation with PEARL

- Planned net new dwellings per municipality per year (zachte plannen) have been obtained through a questionnaire among provincial- and municipality-level planners
- 60 municipalities responded, the rest have been imputed
- A major source of uncertainty: do the planners (or we) know building construction until 2050?
- PoC with Azure: run 100 simulations where stochastic construction plans are created with a simple method

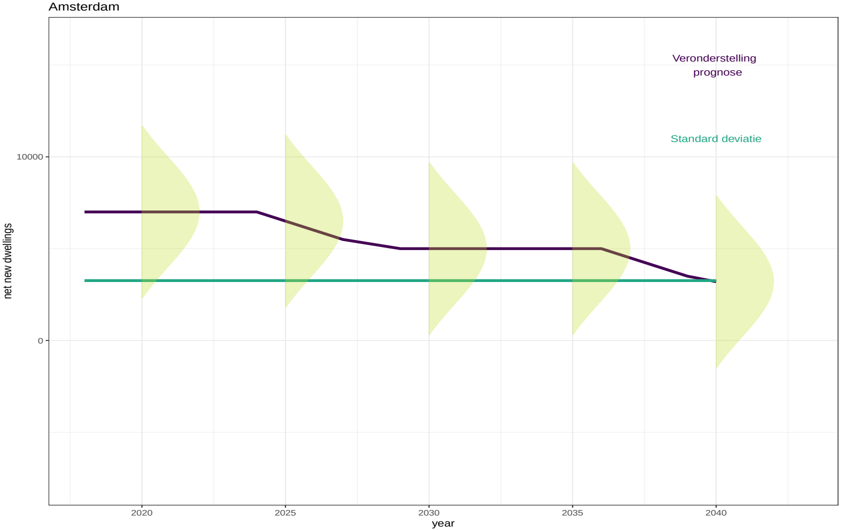
Stochastic construction plans



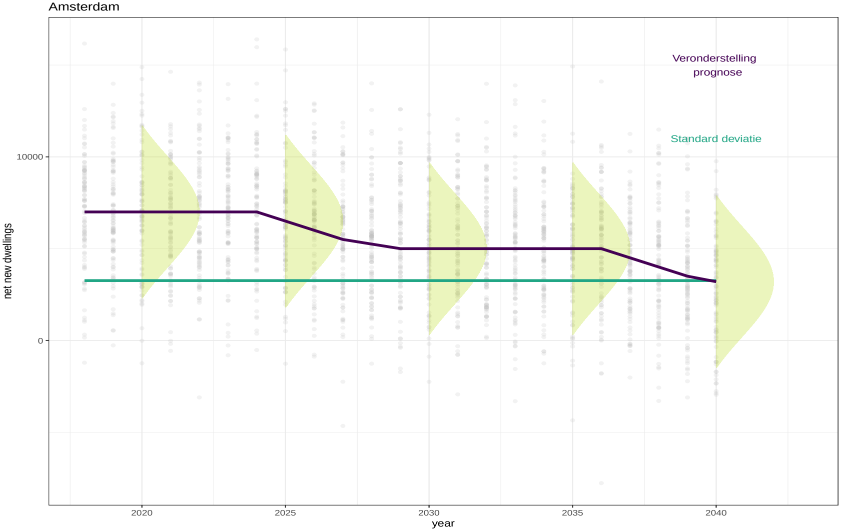
Stochastic construction plans



Stochastic construction plans



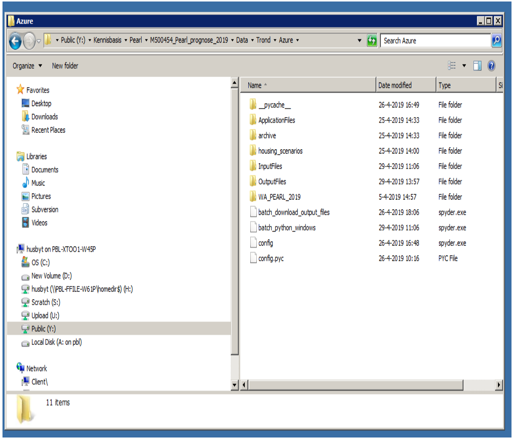
Stochastic construction plans



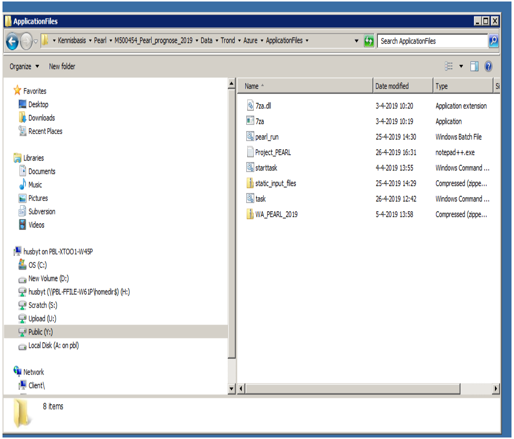
PEARL in de cloud

- PEARL: vb.net application; settings with .ini; and ASCII input files. Run time 4̃0 min
- Connection with Azure using Ruben's python script from the PBL Onderzoeksomgeving
- 100 D1 v1 Windows VMS
  - Folders: Application, InputFiles, OutputFiles
  - *config.py*: settings
  - *starttask.cmd*: unzip files to shared dir
  - *task.cmd*: on each VM, copy files from shared dir, run model, create zip files of output
  - Output files pushed back to Storage upon completion of each task
- Separate static input files from simulation inputs into separate folders / zip files
- Download files from storage

PEARL in de cloud

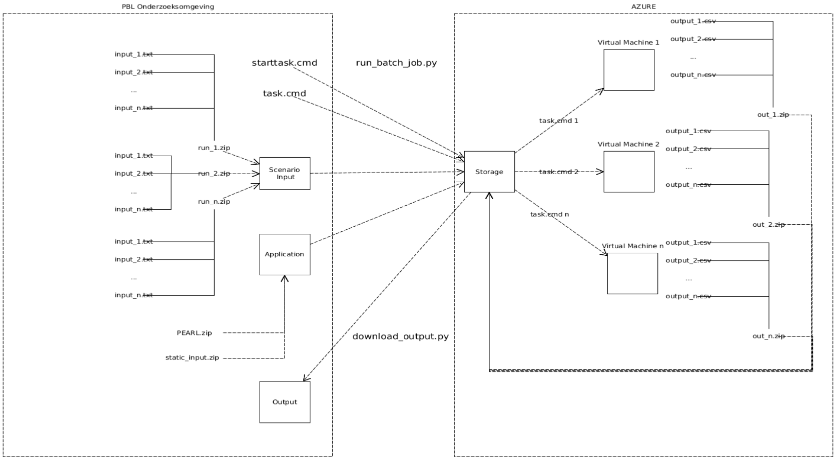


Name	Date modified	Type
__pycache__	26-4-2019 16:49	File folder
ApplicationFiles	25-4-2019 14:33	File folder
archive	25-4-2019 14:33	File folder
housing_scenarios	25-4-2019 14:00	File folder
InputFiles	29-4-2019 11:06	File folder
OutputFiles	29-4-2019 13:57	File folder
WA_PEARL_2019	5-4-2019 14:57	File folder
batch_download_output_files	26-4-2019 18:06	spyder.exe
batch_python_windows	29-4-2019 11:06	spyder.exe
config	26-4-2019 16:48	spyder.exe
config.pyc	26-4-2019 10:16	PYC File

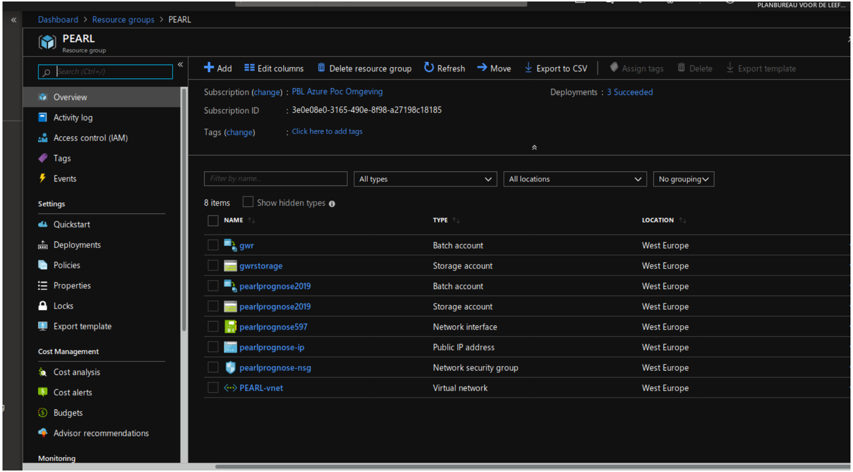


Name	Date modified	Type
7za.dll	3-4-2019 10:20	Application extension
7za	3-4-2019 10:19	Application
pearl_run	25-4-2019 14:30	Windows Batch File
Project_PEARL	26-4-2019 16:31	notepad++.exe
starttask	4-4-2019 13:55	Windows Command ...
static_input_files	25-4-2019 14:29	Compressed (zippe...
task	26-4-2019 12:42	Windows Command ...
WA_PEARL_2019	5-4-2019 13:58	Compressed (zippe...

PEARL in de cloud



PEARL in de cloud



Subscription (change) : PBL Azure Poc Omgeving

Deployments : 3 Succeeded

Subscription ID : 3e0e08e0-3165-490e-8f98-a27198c18185

Tags (change) : Click here to add tags

8 items □ Show hidden types

NAME	TYPE	LOCATION
gwr	Batch account	West Europe
gwrstorage	Storage account	West Europe
pearlprognose2019	Batch account	West Europe
pearlprognose2019	Storage account	West Europe
pearlprognose597	Network interface	West Europe
pearlprognose-ip	Public IP address	West Europe
pearlprognose-nsg	Network security group	West Europe
PEARL-vnet	Virtual network	West Europe

Results

Figure: Short distance arrivals

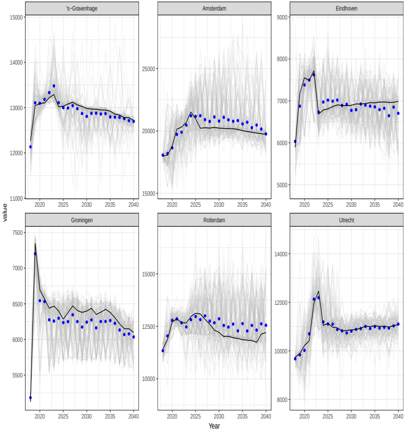
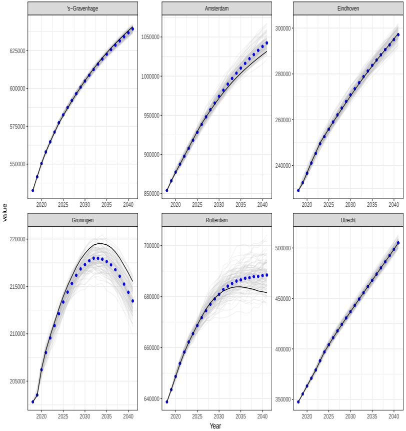


Figure: Population



Conclusions

- Two use cases
  1. Azure opens new possibilities for a computationally heavy model (systematic check of out-of-sample performance)
  2. With Azure we can do fully stochastic population projection (current method is *quasi-stochastic*, with a limited amount of runs)
- Running Azure from R is not so straight forward. Some of the usual suspects (installation of packages; dependencies...) caused problems. Working with Docker images a possible solution
- Ruben's python script does the job but could use some extensions: number of tasks is defined by zip-files in the input folder, no autoscaling options...
- Last but not least: computing power should not be a substitute for efficient code