Dénes Tóth
(toth.denes@kogentum.hu)

LeaRn, eaRn, retuRn

A tour in R & data analytics

CEU, Budapest
March, 02, 2020

KOGENTUM
egyszerüen komplex

Content

- Bio
- Learn
  - Research
  - Self-education
  - R as a data analytic tool
  - R as a software
  - R as a community
- Earn
  - Education
  - Logistics
  - Deduplication
- Return
  - BURN, satRday
  - Teaching
  - Open-source

Bio

- Education:
  - Univ. Of Economics (1998-2003):
    - Management + Applied Statistics
  - ELTE, Psychology (2000-2005):
    - Cogn. + developmental psychology
  - Psychology PhD (2005-2009 [PhD: 2013])
- Work:
  - Hungarian Academy of Sciences (2006-2017)
  - Kogentum (2013-)

Bio

- *Lesson*:
  - Data science is more like soccer, not volleyball or (horse) polo


SFU SPIRIT
SEE YOU TONIGHT
THE SIMON


LACOM n.v.
WWW.LACOM.BE
BOUW
SCHILDER
VANDEZANDE
Tel.
016 40 20 40
VANDERMEULEN
Gevelrestauratie
Verhuur van steigers
www.vandermeulen.be
ianfootball.be
www.belgianfo

Learn

Learn

- Research
  - Visual word recognition
  - Reading development, dyslexia (reading disorder)
  - Reproducibility of neurobiological signals

Notes on research

- Autonomous work (groups) preferred
- There is time to explore
- Extremely competitive
- Very complex right from the start, and gets more complex as time passes by
- Frustration tolerance is a must
- Packaging the results is important

Visual word recognition

800-1200 ms

400-600 ms

SAME

DIFFERENT

83 ms

Many other variants,
e.g. masked priming...

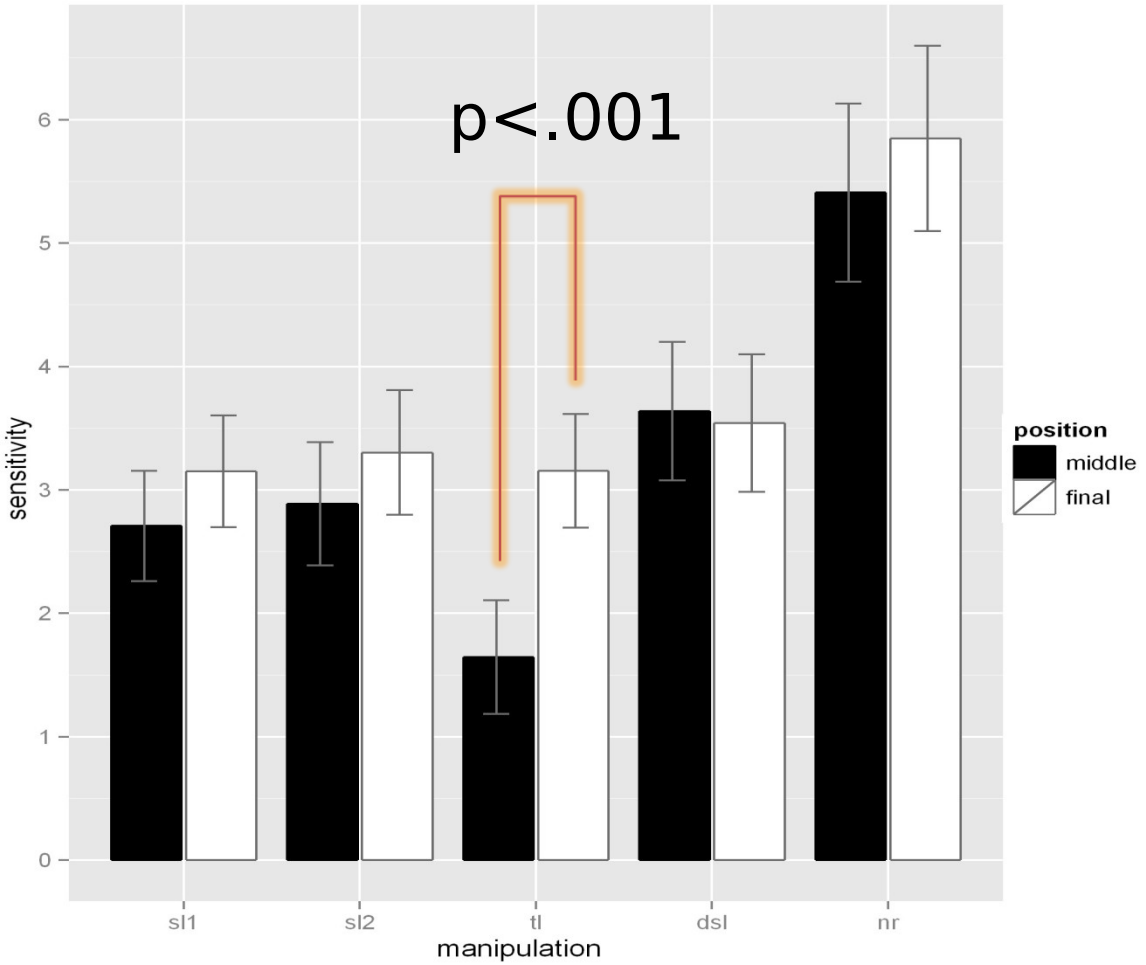
Results

Middle:

TL < SL1 = SL2 < DSL < NR

Final:

TL = SL1 = SL2 = DSL < NR



Visual word recognition

- The transposed letter effect:
  - birgam/BIGRAM vs binjam/BIGRAM?
  - Behavioural experiments:
    - Accuracy and response time
    - Stat: (generalized) mixed effects model
- The first experience with R: **lme4**
  - All pre- and postprocessing in MATLAB (uhh)
  - Modeling in R
- Example research paper

Dyslexia diagnosis – 3DM-H

Hamarosan szavakat fogsz látni.

Olvasd el az összes oszlopot felülről lefelé, amilyen gyorsan csak tudod.

Figyelj jól! Ha minden szót elolvastál, ismét új szavak jönnek.

Olvass, amíg a teszt magától le nem áll.

A feladatra fél perced van.

Szólj, ha folytathatjuk!

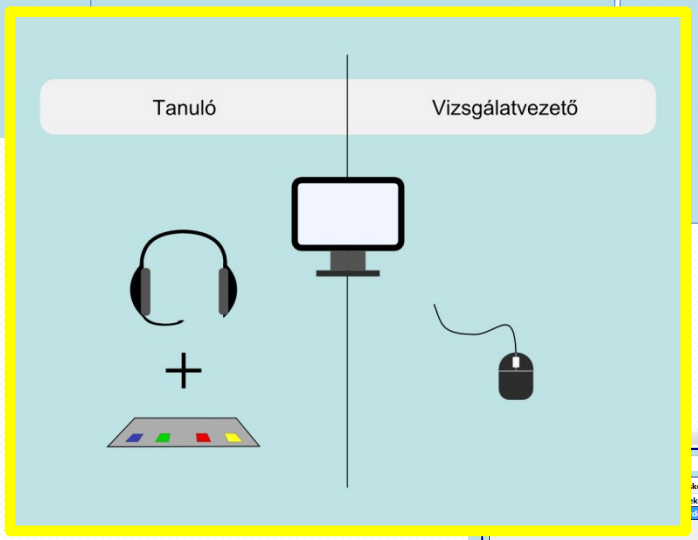
nap

kéz

rend

város

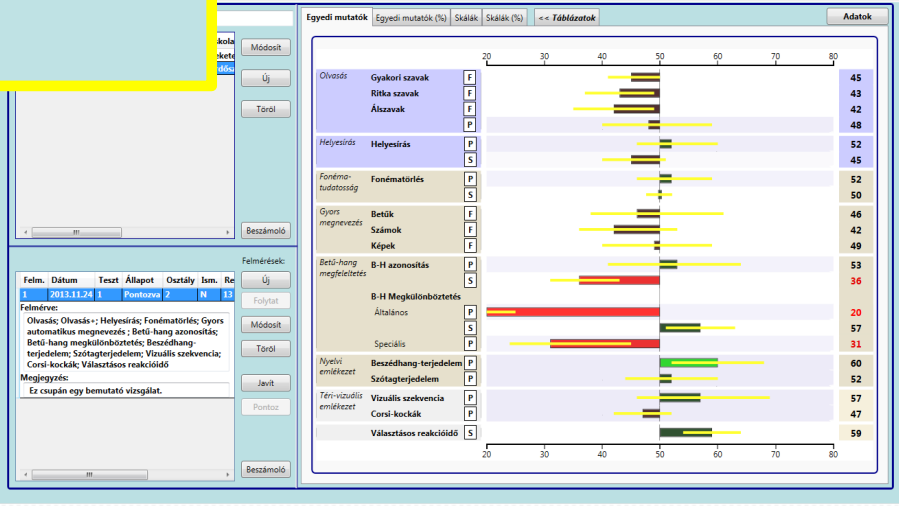
név	tűz	hős
kör	hit	dél
cél	gól	por
víz	fül	tej
láb	mag	kút



mé

Pontozás:

Helyes | Javított | Hibás



Dyslexia diagnosis – 3DM-H

- 3DM:
  - Differential Diagnosis of Dyslexia, Maastricht
  - 3DM-H: The Hungarian adoptation (now a commercial software)
- Psychometrics: how to create a valid and reliable psychological test?
- Softwares used (in research stage):
  - Presentation (NeuroBS): subtests
  - MATLAB: GUI for response evaluation and data uploads
  - R: all analyses
    - Cleansing, outlier detection (e.g., modi)
    - Item response theory (e.g., ltm)
    - Factor analysis (e.g., psych)
    - Calculation of standard scores (e.g., gamlss)

Reading development (ProRead)

Research Article

aps ASSOCIATION FOR PSYCHOLOGICAL SCIENCE

Psychological Science
21(4) 551–559

DOI: 10.1177/0956797610363406
http://pss.sagepub.com
SAGE

Orthographic Depth and Its Impact on Universal Predictors of Reading: A Cross-Language Investigation

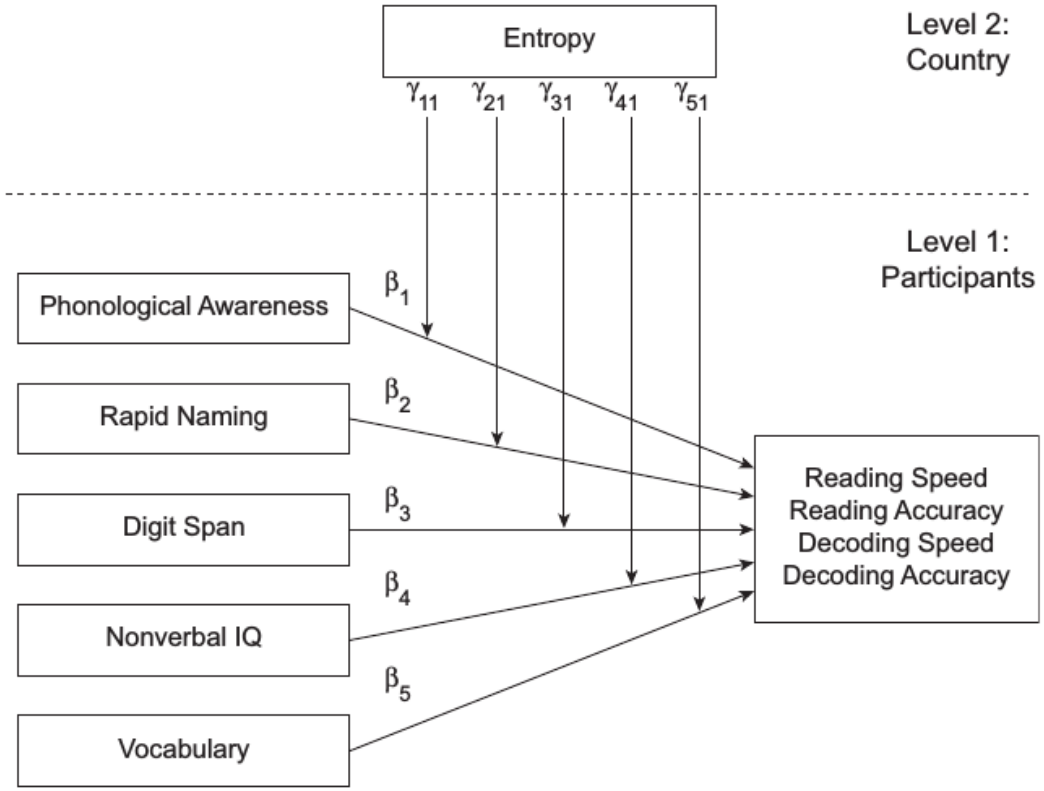
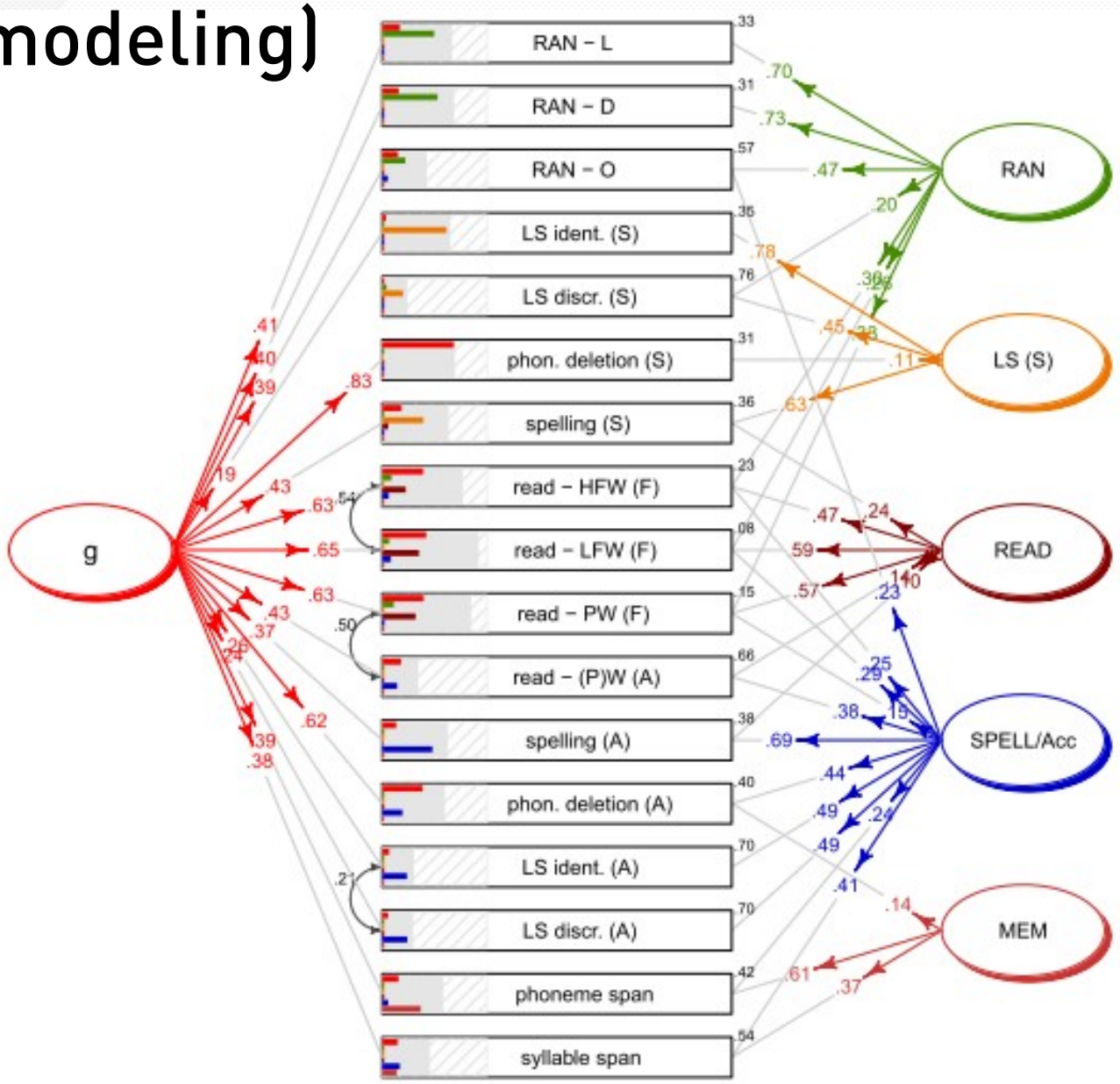


Fig. 1. Basic design of the study displaying the five factors that influence reading and decoding at the individual level (Level 1) and their potential modulation by script entropy (Level 2).

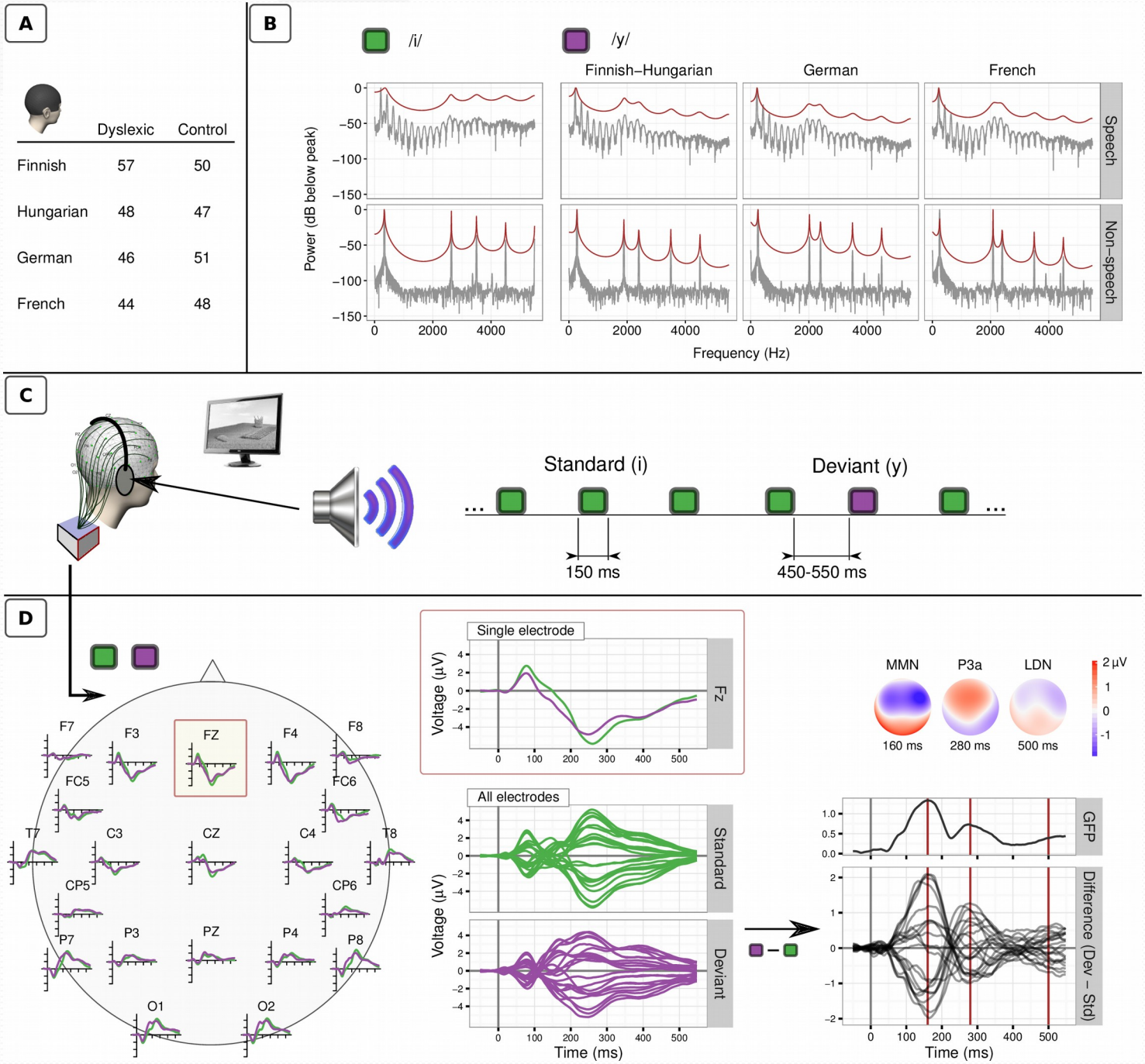
CFA (conf.factor.an.) and SEM (struct.eq.modeling)

- lavaan



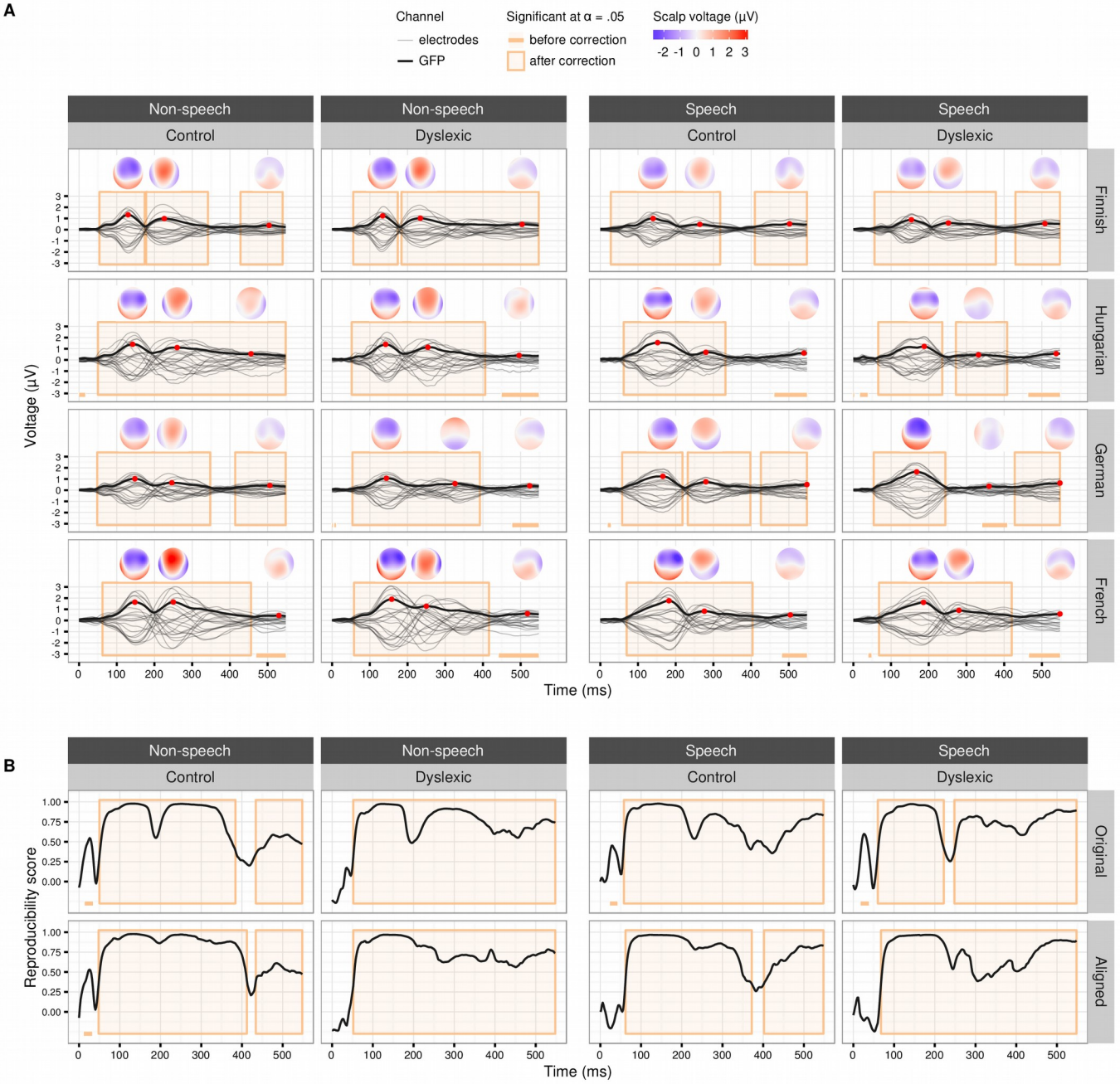
The NeuroDys project

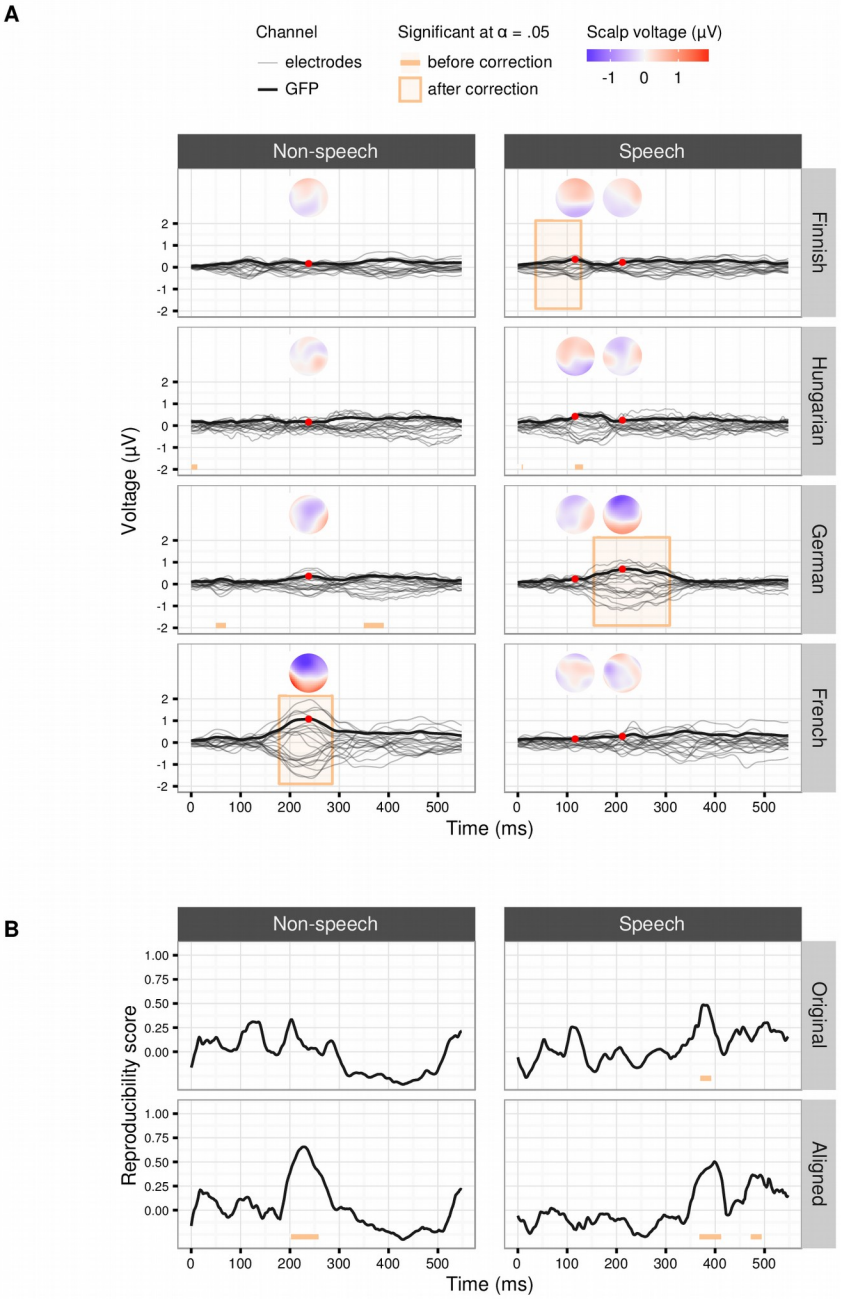
- Behavioural (N > 2000), neurobiological (ERP: N ~ 400) and genetic (N > 2000) background of dyslexia
- ERP: a mismatch negativity (MMN) experiment in four countries, 200 dysl + 200 control students
- Q: How reproducible is the MMN component & the DL-Contr difference?


A
Dyslexic
Control
Finnish 57 50
Hungarian 48 47
German 46 51
French 44 48
B
/i/
/y/
Finnish–Hungarian
German
French
Speech
Non-speech
Power (dB below peak)
Frequency (Hz)
C
Standard (i)
Deviant (y)
150 ms
450-550 ms
D
F7
F3
FZ
F4
F8
FC5
FC6
T7
C3
CZ
C4
T8
CP5
CP6
P7
P3
PZ
P4
P8
O1
O2
Single electrode
Fz
Voltage (μV)
All electrodes
Standard
Deviant
Time (ms)
MMN
P3a
LDN
160 ms
280 ms
500 ms
2 μV
GFP
Difference (Dev – Std)

eegR

- A self-developed R package to analyze EEG signals
- 


A
Channel
electrodes
GFP
Significant at α = .05
before correction
after correction
Scalp voltage (μV)
-2 -1 0 1 2 3
Non-speech
Control
Non-speech
Dyslexic
Speech
Control
Speech
Dyslexic
Finnish
Hungarian
German
French
Voltage (μV)
Time (ms)
B
Reproducibility score
Original
Aligned
Time (ms)


A
Channel
electrodes
GFP
Significant at α = .05
before correction
after correction
Scalp voltage (μV)
-1
0
1
Non-speech
Speech
Finnish
Hungarian
German
French
Voltage (μV)
Time (ms)
B
Non-speech
Speech
Original
Aligned
Reproducibility score
Time (ms)

Self-education

- What worked for me:
  - Try to solve real problems
  - Experiment with various statistical methods
  - RTFM (and other official docs)
  - Read the source code
  - Let the community help you („official" R mailing lists, stackoverflow, Cross Validated):
    - Important: learn how to ask for help
  - Be part of the community (e.g., join BURN)

Self-education

- Try to be up-to-date (e.g., r-bloggers, follow relevant developers) - but do not hop on the hype train
- Some books of Hadley Wickham (Advanced R, R packages) or from other reliable sources
- Use the CRAN Task Views

- Might work as well
  - R online courses (e.g., DataCamp, Coursera, edX)
  - The swirl package

R as a data analytic tool

- Developed by statisticians for statisticians (see its history)
- So good for data wrangling:
  - data.frame (and its offsprings, e.g., data.table or tibble)
  - missing values
  - indexing by names
- So good for visualization:
  - Base graphics, lattice, ggplot2 (with all its extensions)
- So good for modeling:
  - If you can not find an R package for a statistical method, it is very likely that you should not use it
- Even good for reporting:
  - knitr, pander, rmarkdown, and lots of other recent goodies
- Reproducible science

R as a software

- Not-that-steep learning curve at the start
  - Users with stat background: very different from SPSS and cousins
  - Users with comp. science background: lazy and non-standard evaluation, lots of quirks
- Requires much less effort later
- The important thing is to grasp the basics right
  - Everything that exists in R is an object.
  - Everything that happens in R is a function call.
  - Base objects, indexing, functions

R as a software

- Became easier to follow coding „best practices”:
  - Use a proper Integrated Development Environment (IDE)
  - Structure your projects
  - Use version control
  - Use consistent style and naming in your scripts
  - Document your functions
  - Take care of package dependencies
  - Write unit tests

R as a community

- Phantastic:
  - Very supportive
  - Very large user base
  - Open-source at its heart

Earn

Education

- Use data science to improve education efficiency of secondary schools in Australia

The start

- First interview:
  - Do you have any expertise in SEM modeling (structural equation modeling)?
- First impression:
  - OK, but what about the data?

Pre-production phase

- First task: create static (pdf) reports
  - Import data from various file formats (txt, csv, xls, xlsx, pdf) provided by the schools
  - Data content: student enrolments, student attendance, teacher assessments, external tests (e.g., NAPLAN), certificates
  - Problems: badly formatted files, tables with multiple (different) headers, missing and non-matching student IDs, no domain knowledge

Pre-production phase

- Solution:
  - Create a package which extracts the data from the raw files
  - Explore the data
  - Write some helper functions and scripts which transform the imported data to proper data tables
  - Analyze data and create charts
- Most used packages:
  - data.table, ggplot2

Prototyping phase

- Task: Create a Shiny application by which a user can run simple analyses on in-memory, pre-processed data.
- Solution:
  - Create two more packages; one for analytics/visualization and one for the Shiny app
  - User feedback:
    - Users do not understand how to use the application.
- ***Lesson:***
  - **Never underestimate the complexity of simplicity.**

Post-prototype phase

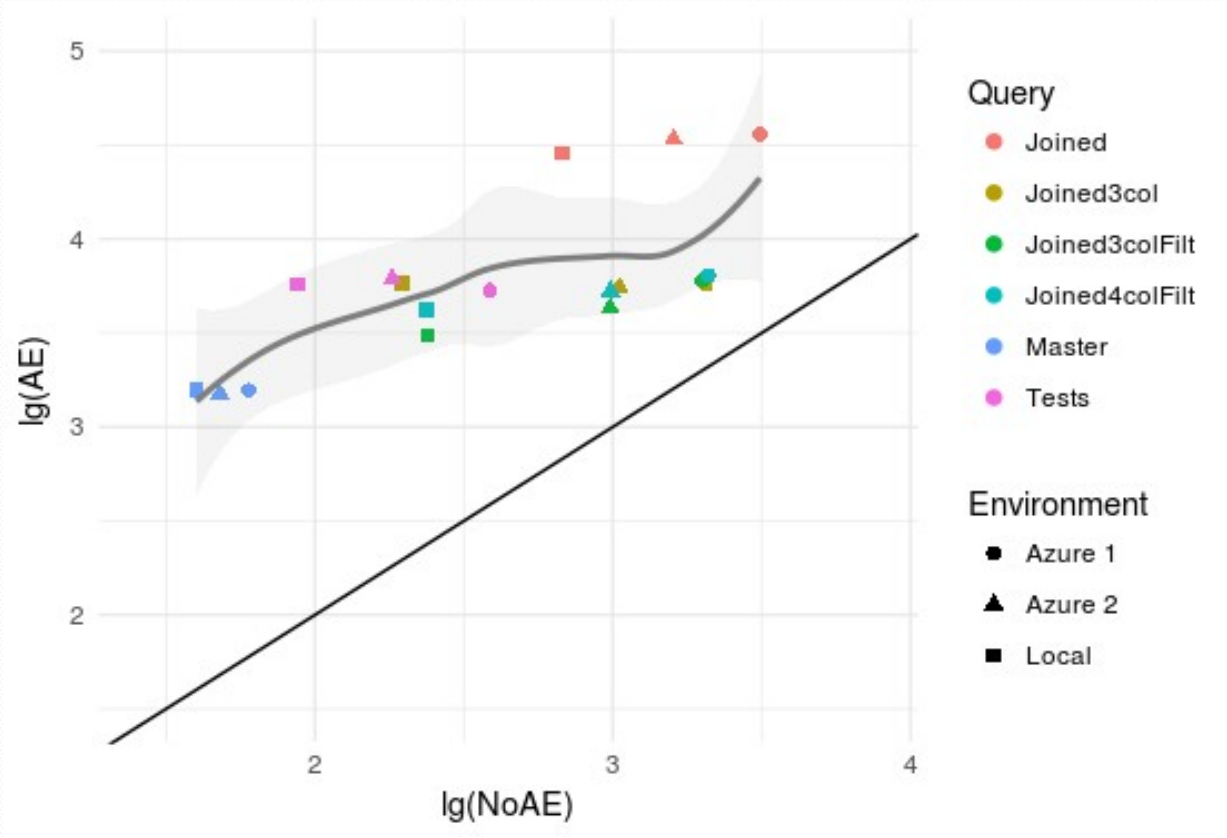
- Decisions to be made:
  - Infrastructure (own server, school server, cloud, hybrid)?
  - Cloud provider (AWS/MS Azure/Google Cloud Platform)?
  - Web framework (R Shiny, something standard + OpenCPU)?
  - Meet (mostly unknown) security/privacy requirements
  - Reporting framework (MS Office, WYSIWYG HTML editors)
  - Development and deployment framework/cycle

Post-prototype phase

- Decisions:
  - Fully cloud-based infrastructure
  - MS Azure
  - R Shiny served by ShinyProxy
  - SQL Database Service with MS Always Encrypted
  - Reports: Integration with MS Office via a custom add-in
  - GitLab + OpenProject
- ***Lessons:***
  - **Feel free to disregard the framework of your prototype.**
  - **Do not believe in rumors and self-advertisements. Check it out on your own or consult with experts with hands-on experience.**

MS Always Encrypted

- Extreme overhead (5-60x performance degradation)



ShinyProxy by OpenAnalytics

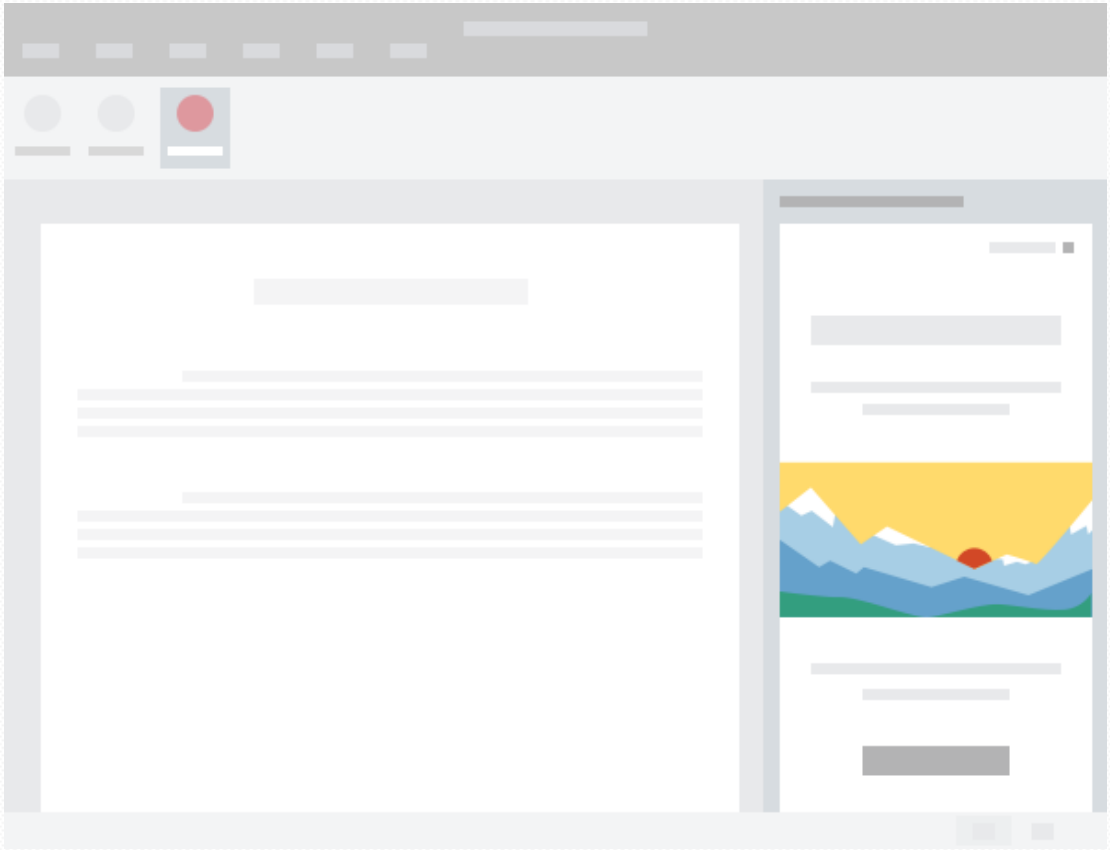
What is ShinyProxy?

ShinyProxy is your favourite way to **deploy Shiny apps in an enterprise context**. It has built-in functionality for LDAP authentication and authorization, makes securing Shiny traffic (over TLS) a breeze and has no limits on concurrent usage of a Shiny app.

- Free and open source, just like R :)
- You deploy exactly the same Shiny application that you develop
- It can be easily customized, e.g. extended by custom authentication module
- Fast and reliable
- If higher-level scalability needed, supports Docker Swarm out of the box and running in Kubernetes is also possible

Office add-in + WOPI

- You get the full power of Office Online, but implementing it is not a breeze…



Challenges - workflow

- Workflow of a lone wolf data scientist
  - Endless data mungling
  - Explore super fast, try to deliver early
  - Write functions and scripts because you do not want to type a lot
  - Do-and-forget
  - The only documentation is the report that you produce in the end
  - One-off visualizations with proper tweaks
- Workflow of a software development team
  - Specification
  - Version Control
  - Documentation
  - Testing
  - Automatic deployment
  - Tracking user activity (logging)
  - Finally you even write some code, but mostly debug yours' and others'

Challenges - app

- Questions of interest:
  - Even with a modest set of analytic procedures, the number of potential analysis setups is extremely high due to potential filtering and subgrouping
  - Example: how girls belonging to the 2014 entry cohort progressed in the Numeracy NAPLAN domain?
- Target users:
  - No data analytic or statistical skills
  - What is the best way to guide the user to formulate meaningful questions that you can respond to?

Challenges - app

- Performance bottleneck: gather the data
- Data coverage:
  - Far from being perfect
  - Hard to evaluate beforehand
- Expected usage pattern:
  - Infrequent data uploads (max. 3-4 times per year)
  - Highly overlapping analytic questions
- ***Lessons:***
  - **Think first, code second. BTW, the former is much harder.**

The app (as of now)

- A cloud based web application which provides:
  - Advanced dashboard-like functionality in which the user formulates the questions in a flexible step-by-step manner and the answer is delivered via interactive charts and tables
  - Full integration with MS Word Online to create reports

Tips (Shiny)

- You need working knowledge of JS and CSS, or even better an experienced web dev.
- Tips:
  - Package instead of app.R/server.R/global.R
  - Use modules
  - Keep the reactive flow and the actual data preparations/calculations/visualizations separated (use functions for the latter)
  - Do not insist on Shiny and reactivity; sometimes it is just a lot easier to write a JS widget with simple event handlers
  - Do not let the application crash
  - Create tests, even if it is hard

Tips (general)

- *Never hide hard-wired parameters* in the R code - use a configuration file or add them as options
- Use *yaml* with anchors for configuration
- Always provide *examples* and *proper documentation* for exported functions
- Write *unit tests* (e.g., **tinytest**, **testthat**)
- Prefer packages with less *dependencies*

Tips (general)

- Write *modular* code and avoid large, monolithic packages
- If you do the same thing twice in your code, write a *function*.
- If you call the same set of functions again and again, or the same functions in various packages, wrap them in a separate *package*.
- Prefer *readability* over speed, but always benchmark the speed of your functions (e.g., `bench::mark`)
- Do always *type-check* your arguments within your functions (e.g., **checkmate**)

Logistics

Logistics

- Solve a complex vehicle routing problem for a company with >150 trucks and >6000 shops to serve per week

The start

- Question:
  - They want to know how many drivers they need to serve the shops. Do you know how to solve this problem?
- Response:
  - Nope, but we have a full month to figure this out.

Pre-production phase

- Read the literature
- Find an open-source library with the required set of features:
  - happened to be two *Java* libraries (**graphhopper** and **jsprit**)
- Do all data pre- and postprocessing in *R*, and customize the *Java* libraries to be able to answer the business questions
- Create the reports in *R*

Production

- Implement a SAAS solution which:
  - Considers various constraints (truck capacity, opening hours of shops, max. work time, load-dependent consumption etc.)
  - Automatically creates optimized routes for future dates (based on predicted orders)
  - Automatically adjusts the routes for the actual day (based on the actual orders)
  - Allows manual modifications of routes
  - Has interfaces to several other CRM and logistic systems
  - Creates Excel reports

Solution

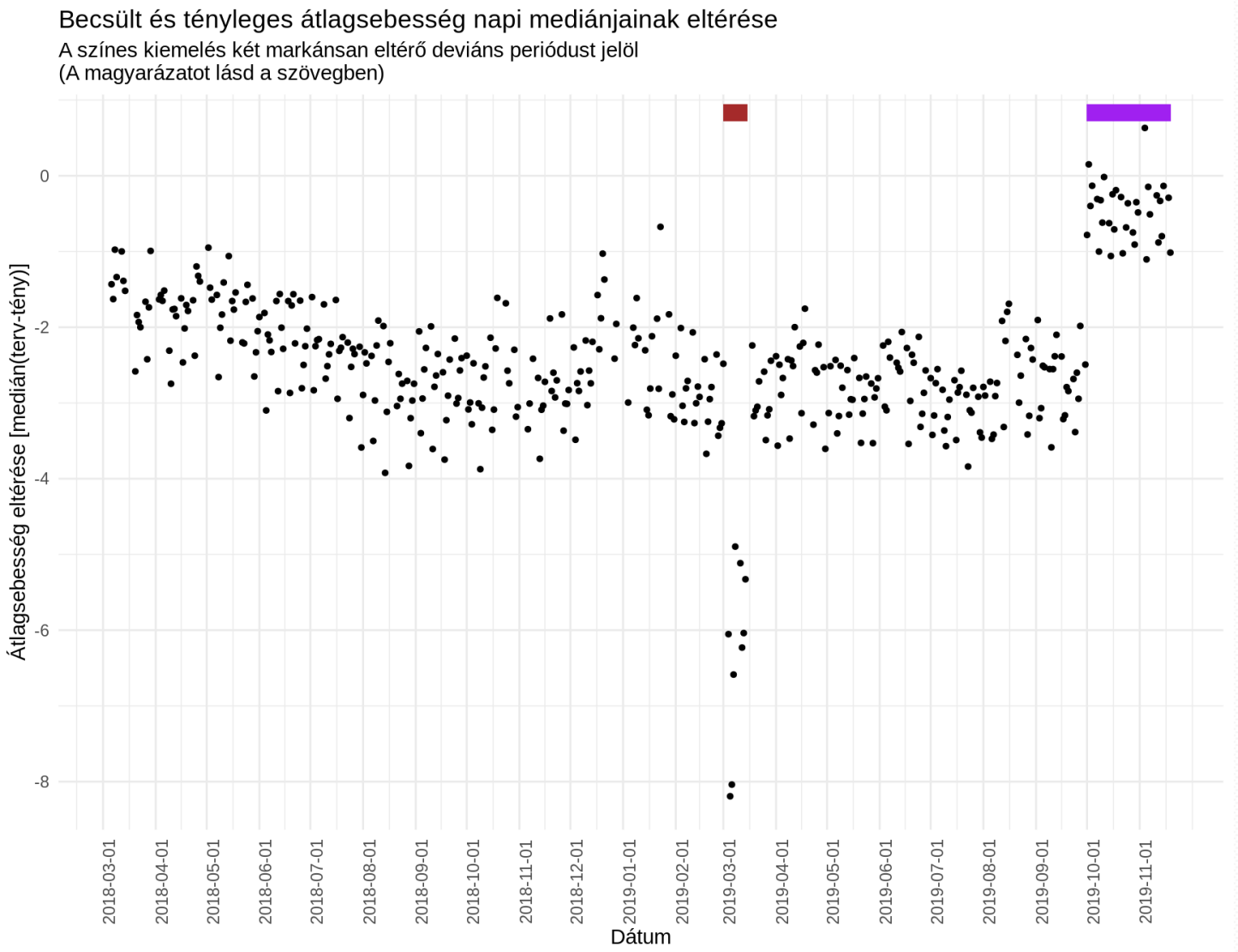
- Scala (Play) + JavaScript + MongoDB
- My tasks:
  - algorithm development
  - prototyping in Java
  - convert business demands
- ***Lessons:***
  - **R is a Swiss Army knife, but one does not use a Swiss Army knife for clearcutting**
  - **No need to reinvent the wheel**

Extensions – travel time estimation

- Task:
  - Could we improve the accuracy of estimated travel times? (GraphHopper's default is not that precise)
- Input:
  - GPS tracking data (only destination endpoints)
- Solution:
  - Data preparation (linkage)
  - Explore potential external effects
  - ML with postal regions and start time (xgboost)
  - Create report
- New task: nice results, could we implement it?

Extensions – travel time estimation

- Bang:



Extensions – travel time estimation

- Solution:
  - Dig deep into GraphHopper internals
  - Two-step model:
    - OSM-edges: speed of various route types (elasticnet)
    - Destinations: ML with postal regions and start time (xgboost)
- **Lessons:**
  - **Always expect unforeseen issues**

Deduplication

Deduplication

- Context:
  - A large society which has to collect data (of highly varying quality) on several entity types from various external sources
  - Consequence: large amount of duplicated entities in the central database
- Task:
  - Develop a concept of a software which can automatically deduplicate the main entity type

Concept

- Steps:
  - Read the literature
  - Meet with the key persons (IT, business), familiarize with the business domain
  - Explore the database (almost no documentation)
  - Analyze the data in *R* (data.table) → understand the problem
  - Provide and ask for feedback
  - Dirty implementation of algorithms
  - UI prototyes
  - Write the concept (>70 pages)

Follow-up

- Rival 1:
  - Engineers
  - Solution not domain- or client-specific
- Rival 2:
  - Data scientists
  - Open-source libraries (Python), fancy methods
  - Domain-, but not client-specific
- *Lesson*:
  - **Understand the domain and the data**
  - **Develop feasible solution(s)**

Follow-up

- Phase 0 request:
  - Please document our database

Earn - summary

- General lesson (consultancy):
  - Knowledge/ability to learn: default
  - Expertise: +
  - Social aspects/soft skills: +
  - Deliver on time: +
  - Perfectionism is **negative**

Return

BURN

- Budapest Users of R Network
- >1500 members

satRday

Tweet Follow @satRdays_org

Latest News

NEWS

Video records are now available (Sep 17 2016)

Ustream recorded and live streamed all conference talks, from which we received a HD version and I cut into pieces -- so that you can rewatch the talks at any time. But don't forget: the goal of the satRdays series is to enable networking among R users, so make sure to attend the next conference instead of waiting for the videos to be uploaded :)

NEWS

Workshop and talk materials uploaded (Sep 11 2016)

Thanks to our awesome speakers, almost all conference talks and workshop materials, including the slides, are now available. You can also rewatch the whole conference in the archive of the live stream, but the HD version of the talks will be also soon uploaded here -- working on the final cuts right now.

NEWS

Quick summary on the conference (Sep 8 2016)

SAT R DAY

CONFERENCE

- Latest News
- About the Conference
- Sponsors
- Important Dates
- Registration
- Live stream of talks

Teaching

- ELTE, BA-MA-PhD
  - https://tdeenes.gitbooks.io/rintro_ma/content/
- Business clients
- ***Lesson***:
  - **Keep it simple**
  - **Focus on „real” problems**

Open-source

- At least:
  - Acknowledge the package(s) you use
  - Raise issues, report bugs
  - Respect licenses
- If you can:
  - Provide help on mailing lists and forums
  - Create pull requests to open-source packages
  - Publish packages

KOGENTUM

egyszerűen komplex