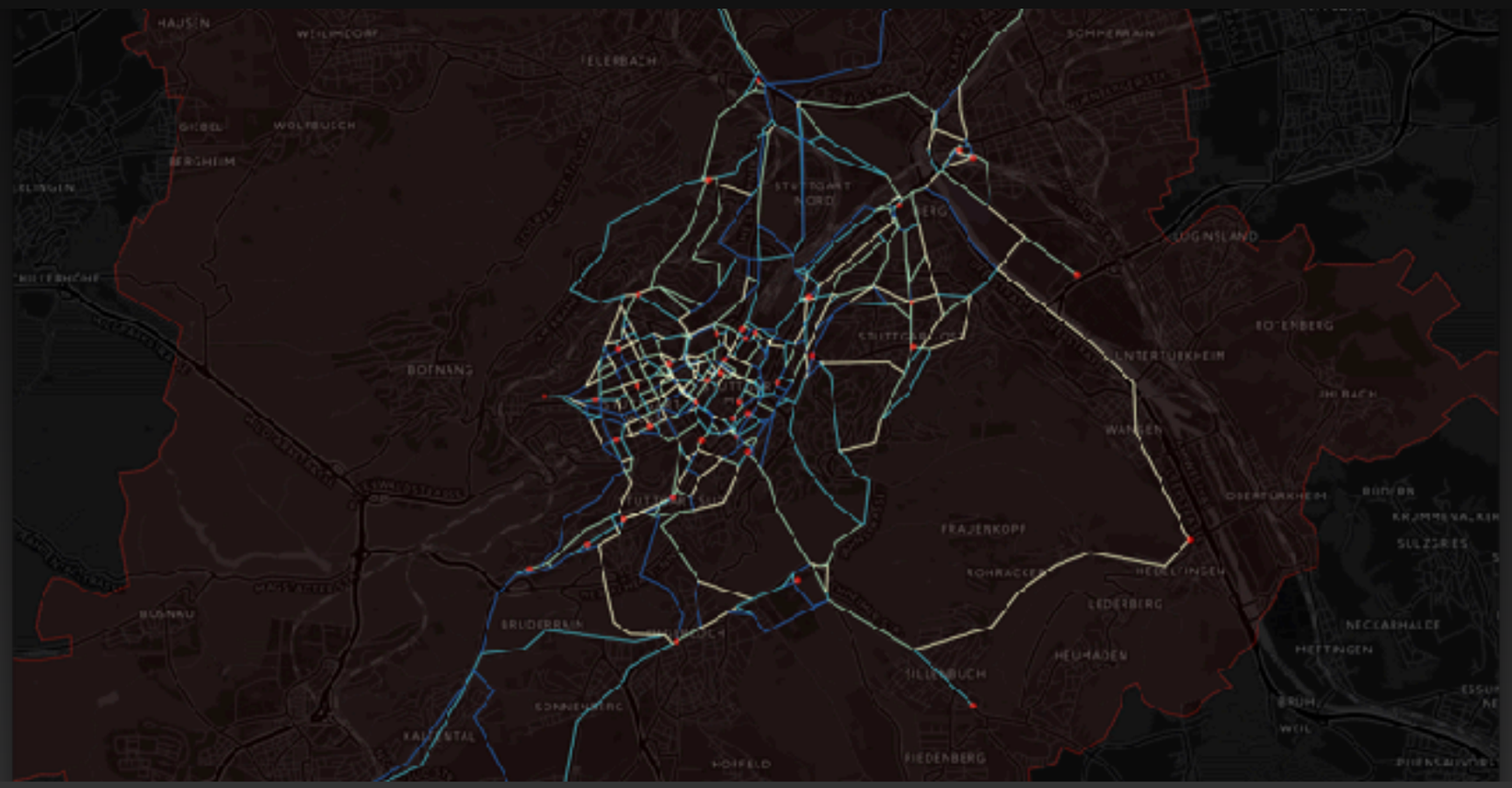
Reproducible Bike Ride in Hamburg

…or elsewhere in Europe



Carles CG

Data Scientist

Budapest BI 2017, 14th November

bit.ly/bike_rideR

Reliable Dynamics

Reproducibility

+

Application in a case

BIKE SHARING USAGE IN HAMBURG

THE DATA

The map shows the bike sharing usage of StadtRAD, the bike sharing system in Hamburg – Germany. The data is available on the open data platform from Deutsche Bahn, the public railway company in Germany. The last new StadtRAD station was put into operation in May 2016, that is why a have chosen to display the usage of June 2016. The brighter the lines, the more bikes have been cycled along that street.

THE PROCESSING

From data processing and spatial analysis to visualization the whole project was done in R. I have used the leaflet and shiny package to display the data interactively. The bikes themselves don't have GPS, so the routes are estimated on a fastest route basis using the awesome CycleStreets API. The biggest challenge has been the aggregation of overlapping routes. I found the overline function from the stplanr package very helpful. It converts a series of overlayed lines and aggregates their values for overlapping segments.

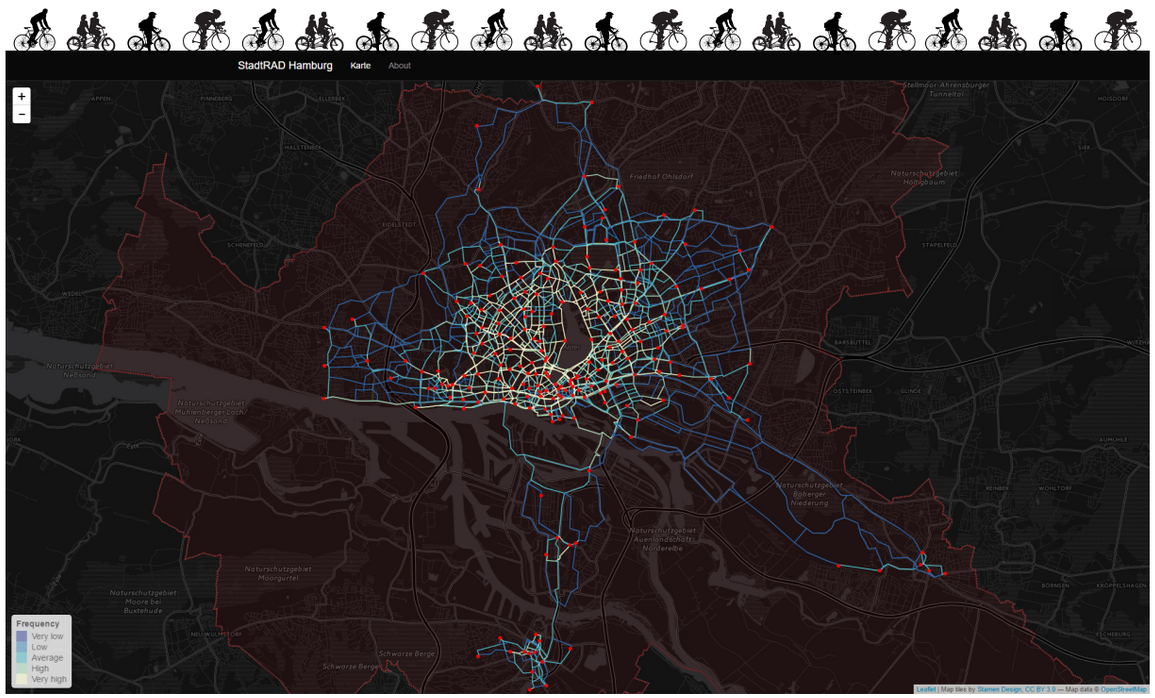
The raw data file from Deutsche Bahn is quite huge so I struggled to import the data into R to be able to process it. In the end the read.csv.sql function from the sqldf package did the job. This way I did not need to import the whole file and just could filter out the bike rides for Hamburg.

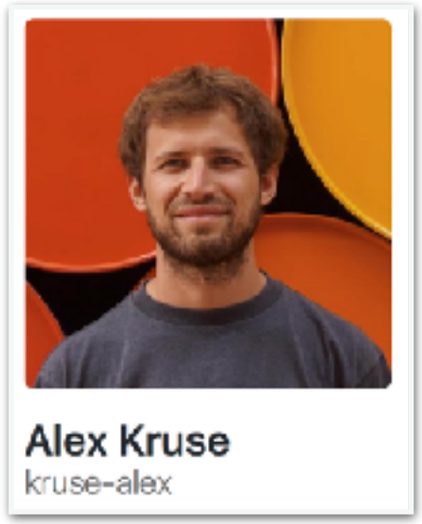
The code could easily be used to map other spatial data, for example the car sharing data from car2go which is available via their API. This might be a future project.

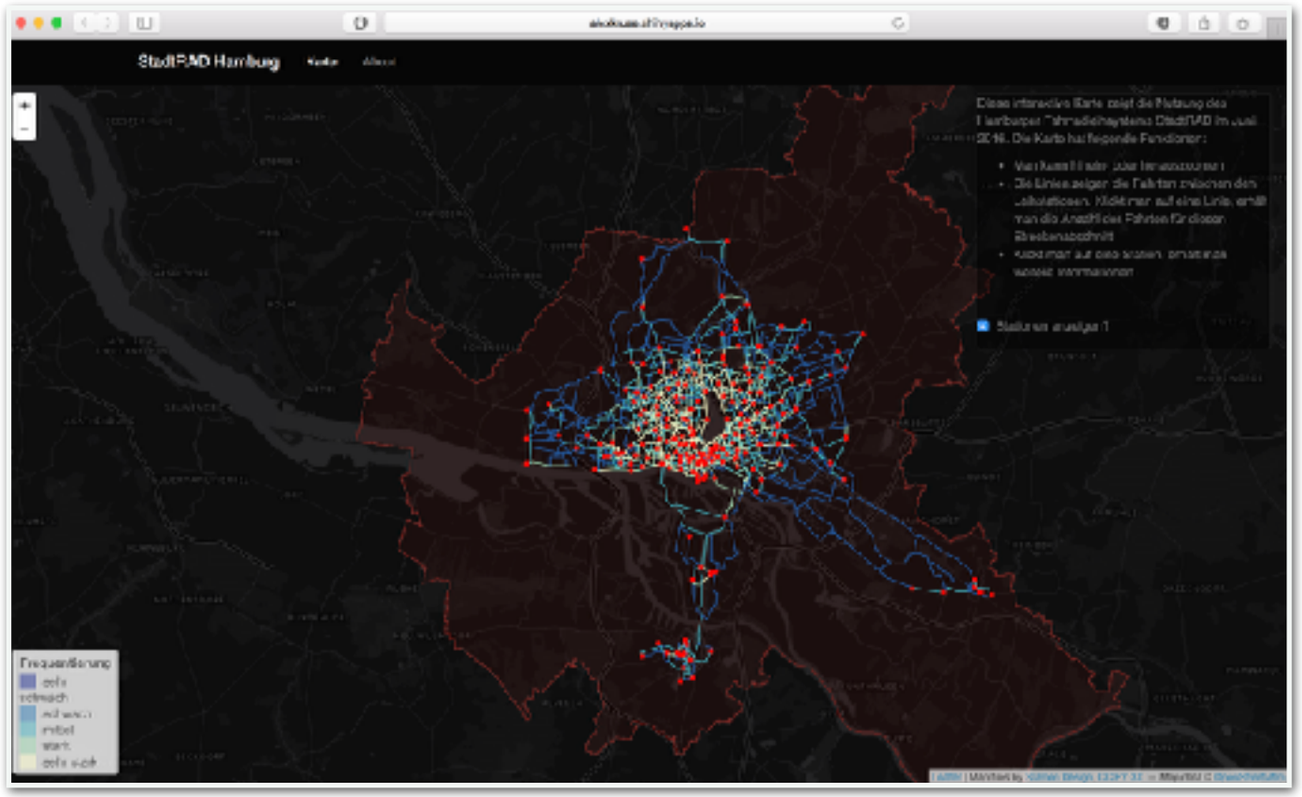
THE MAP

As a cycling ethusiast and Hamburg native I have been riding the streets of Hamburg for a long time now. Over the years I found my favorite cycle routes throughout the city but also know the tight and problematic corners of Hamburg, where missing or overcrowded cycling paths bring you too close to other bikers, cars or pedastrians. It is amazing to see that the data set can proof some of my hypothesis about the current state of the bicylce infrastructure in Hamburg and even bring up new questions I have not even thought about before.

When you look at the map you can see a widely spread bike sharing network over big parts of the city but also notice some enclaving processes where missing stations disconnect bike riders from the high frequented and well connected city center. As the Elbe river seperates Hamburg in a northern and southern part it seems like bike sharing became a well accepted means of transportation to keep both parts of the city connected.








```
##############################################################################################
# IMPORT AND PROCESS DATA
##############################################################################################

# load packages
x = c("sqldf","dplyr","data.table","sp","rgdal","stplanr","reshape2","rmapshaper","leaflet","RColorBrewer")
lapply(x, require, character.only = T)

# import bike rentals
mydata = read.csv.sql("HACKATHON_BOOKING_CALL_A_BIKE.csv",
                      sql = "select * from file where CITY_RENTAL_ZONE = 'Hamburg' ", sep = ";")

# filter on time period
mydata$DATE_FROM = as.POSIXct(strptime(mydata$DATE_FROM, "%Y-%m-%d %H:%M:%S"))
mydata = filter(mydata, DATE_FROM >= "2016-06-01 00:00:00" & DATE_FROM <= "2016-06-30 23:59:59")

# aggregate doubles
mydata = transform(mydata, min = pmin(as.character(START_RENTAL_ZONE_GROUP), as.character(END_RENTAL_ZONE_GROUP)))
mydata = transform(mydata, max = pmax(as.character(START_RENTAL_ZONE_GROUP), as.character(END_RENTAL_ZONE_GROUP)))

# get lan/lat from stations
station = read.csv("HACKATHON_RENTAL_ZONE_CALL_A_BIKE.csv", sep = ";")
station = filter(station, CITY == "Hamburg")

# merge station coordinates with bike rentals
mydata = merge(mydata, station, by.x = "min", by.y = "RENTAL_ZONE_GROUP", all.x = T)
mydata = merge(mydata, station, by.x = "max", by.y = "RENTAL_ZONE_GROUP", all.x = T)

# count bike rides for each route (combine lat/lon)
mydata$start = paste(mydata$RENTAL_ZONE_Y_COORDINATE.x, mydata$RENTAL_ZONE_X_COORDINATE.x, sep = " ")
mydata$dest = paste(mydata$RENTAL_ZONE_Y_COORDINATE.y, mydata$RENTAL_ZONE_X_COORDINATE.y, sep = " ")
mydata = mydata %>% group_by(start, dest) %>% summarise(count = n())

# split lat/lon into two columns
mydata$id = rownames(mydata)
mydata = melt(mydata, id.vars = c("id","count"))
test = data.frame(do.call('rbind', strsplit(as.character(mydata$value),' ', fixed=T)))
mydata = cbind(mydata, test)
mydata = select(mydata, x1, x2, id, count)
colnames(mydata) = c("lat","lon","id","count")
```

```
##############################################################################################
# GET ROUTES FROM CYCLESTREETS API
##############################################################################################

# transform to SPDF
dt = as.data.table(mydata)
lst_lines = lapply(unique(dt$id), function(x){
  Lines(Line(dt[id == x, .(lon, lat)]), ID = x)
})

spl_lst = SpatialLines(lst_lines)
spl_df = SpatialLinesDataFrame(spl_lst, data.frame(mydata$count))

# get routes from cyclestreet API
spl_df = line2route(spl_df, "route_cyclestreet", plan = "fastest")

# add ride count for each route
spl_df@data$count = mydata$count

# simplify SLDF
spl_df = ms_simplify(input = spl_df, keep = 0.01)

# aggregate ride counts on overlapping routes
spl_df = overline(spl_df, attrib = "count", fun = sum)

##############################################################################################
# CREATE LEAFLET MAP
##############################################################################################

# load hamburg shape for the map
hhshape = readOGR(dsn = ".", layer = "HH_ALKIS_Landesgrenze")

# create color palette
qpal = colorQuantile(rev(brewer.pal(4, "YlGnBu")), NULL, n = 4)

# create leaflet map
leaflet(spf_df) %>%
  addPolygons(data = hhshape, stroke = T, smoothFactor = 0.05, fillOpacity = 0.05, color = "red", weight = 1) %>%
  addPolylines(color = qpal(spf_df@data$count),opacity = 1, weight = 1.1) %>%
  addCircleMarkers(lng = station$RENTAL_ZONE_X_COORDINATE, lat = station$RENTAL_ZONE_Y_COORDINATE,
                   fillOpacity = 100, color = "red", stroke = F, radius = 3) %>%
  addLegend(position = 'bottomleft',colors =  rev(brewer.pal(4, "YlGnBu")),
            labels = c("Very low","Low","Average","High","Very high"), title = 'Frequency')
```

github/kruse-alex

@krusealex2013

1. Look at GitHub

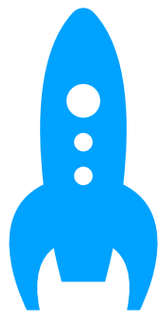
kruse-alex / **bike_sharing**

2. Found the script file!

Branch: **master** **bike_sharing** / **stadtrad_processing.R**

3. Download original data & play with it

```
##                    City Number_entries
## 1               Hamburg         6431973
## 2                  <NA>         1301794
## 3  Frankfurt am Main            1201123
## 4                Berlin         1000297
## 5               München          765202
## 6                Kassel          544558
## 7             Stuttgart          458271
## 8                  Köln          448928
## 9             Darmstadt          251104
## 10              Marburg          179197
```



Bingo!

Disclaimer

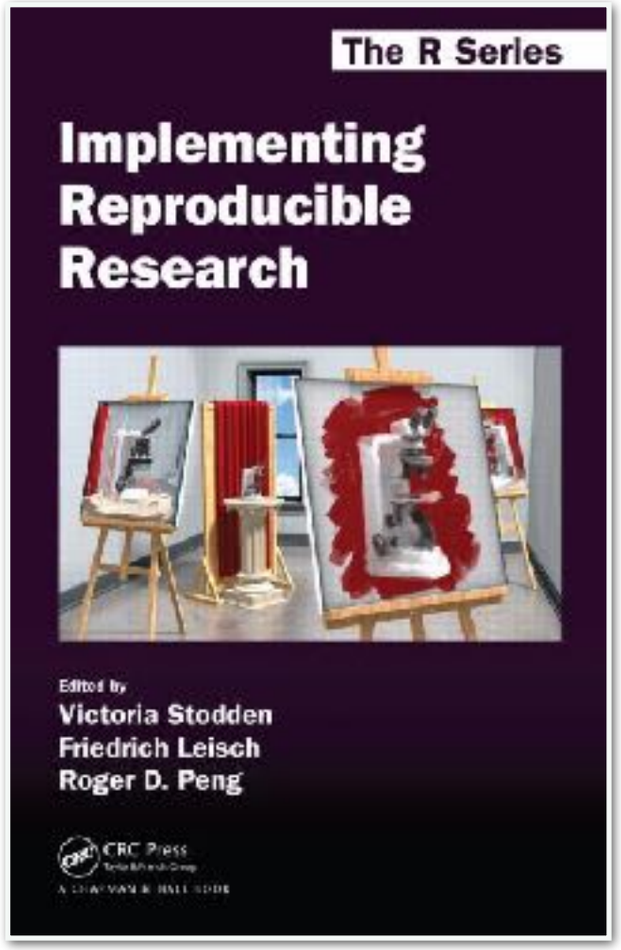
Data Replication & Reproducibility

PERSPECTIVE

Reproducible Research in Computational Science

Roger D. Peng

Computational science has led to exciting new developments, but the nature of the work has exposed limitations in our ability to evaluate published findings. Reproducibility has the potential to serve as a minimum standard for judging scientific claims when full independent replication of a study is not possible.



PLOS | COMPUTATIONAL BIOLOGY

PERSPECTIVE

Good enough practices in scientific computing

Greg Wilson[1☯*], Jennifer Bryan[2☯], Karen Cranston[3☯], Justin Kitzes[4☯], Lex Nederbragt[5☯], Tracy K. Teal[6☯]

1 Software Carpentry Foundation, Austin, Texas, United States of America, 2 RStudio and Department of Statistics, University of British Columbia, Vancouver, British Columbia, Canada, 3 Department of Biology, Duke University, Durham, North Carolina, United States of America, 4 Energy and Resources Group, University of California, Berkeley, Berkeley, California, United States of America, 5 Centre for Ecological and Evolutionary Synthesis, University of Oslo, Oslo, Norway, 6 Data Carpentry, Davis, California, United States of America

☯ These authors contributed equally to this work.
* gvwilson@software-carpentry.org

OPEN ACCESS Freely available online

PLOS | BIOLOGY

Community Page

Best Practices for Scientific Computing

Greg Wilson[1*], D. A. Aruliah[2], C. Titus Brown[3], Neil P. Chue Hong[4], Matt Davis[5], Richard T. Guy[6¤], Steven H. D. Haddock[7], Kathryn D. Huff[8], Ian M. Mitchell[9], Mark D. Plumbley[10], Ben Waugh[11], Ethan P. White[12], Paul Wilson[13]

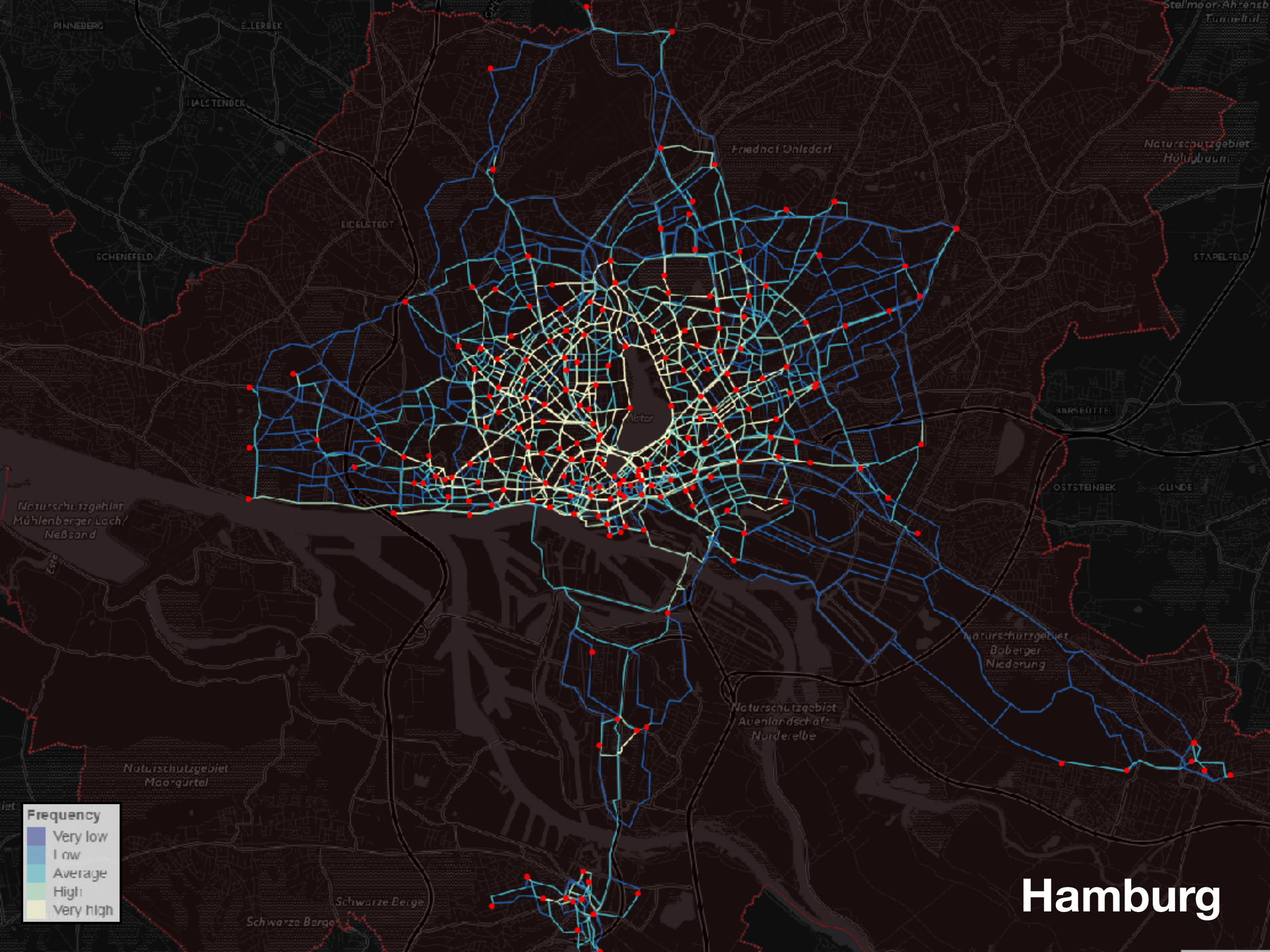
1 Mozilla Foundation, Toronto, Ontario, Canada, 2 University of Ontario Institute of Technology, Oshawa, Ontario, Canada, 3 Michigan State University, East Lansing, Michigan, United States of America, 4 Software Sustainability Institute, Edinburgh, United Kingdom, 5 Space Telescope Science Institute, Baltimore, Maryland, United States of America, 6 University of Toronto, Toronto, Ontario, Canada, 7 Monterey Bay Aquarium Research Institute, Moss Landing, California, United States of America, 8 University of California Berkeley, Berkeley, California, United States of America, 9 University of British Columbia, Vancouver, British Columbia, Canada, 10 Queen Mary University of London, London, United Kingdom, 11 University College London, London, United Kingdom, 12 Utah State University, Logan, Utah, United States of America, 13 University of Wisconsin, Madison, Wisconsin, United States of America

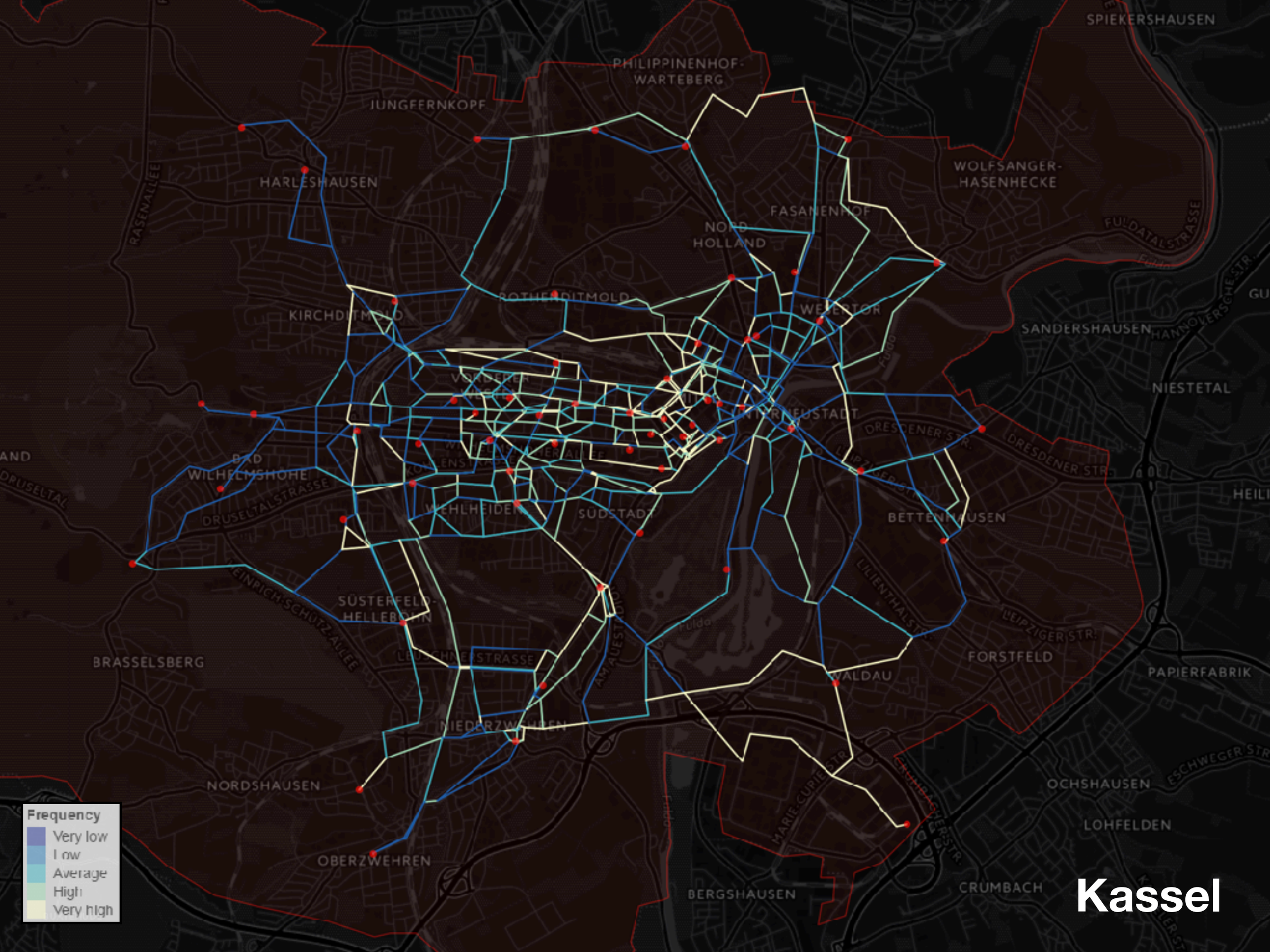
Who am I?

Reliable Dynamics

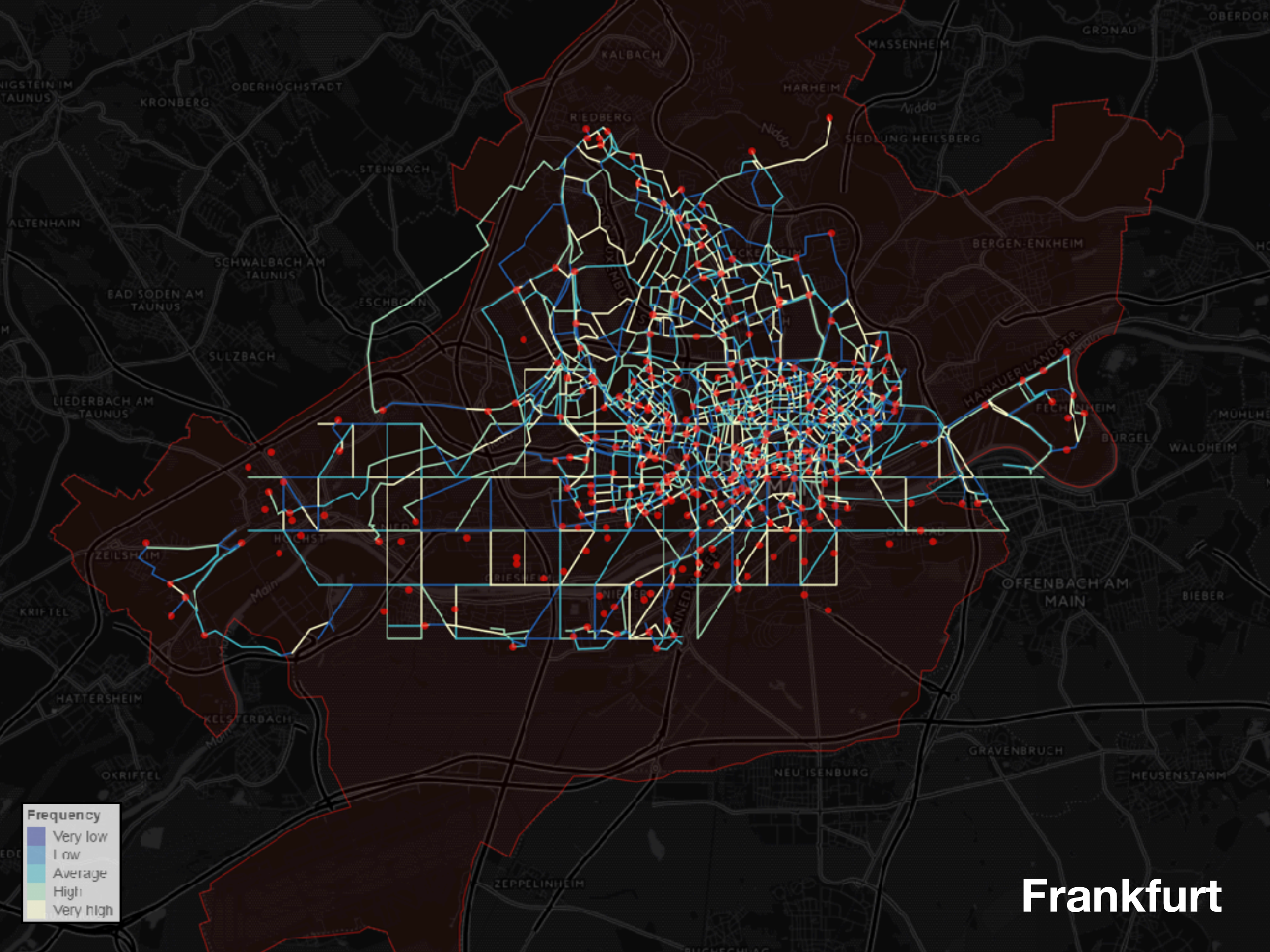
Carles CG
Data Scientist & Consultant

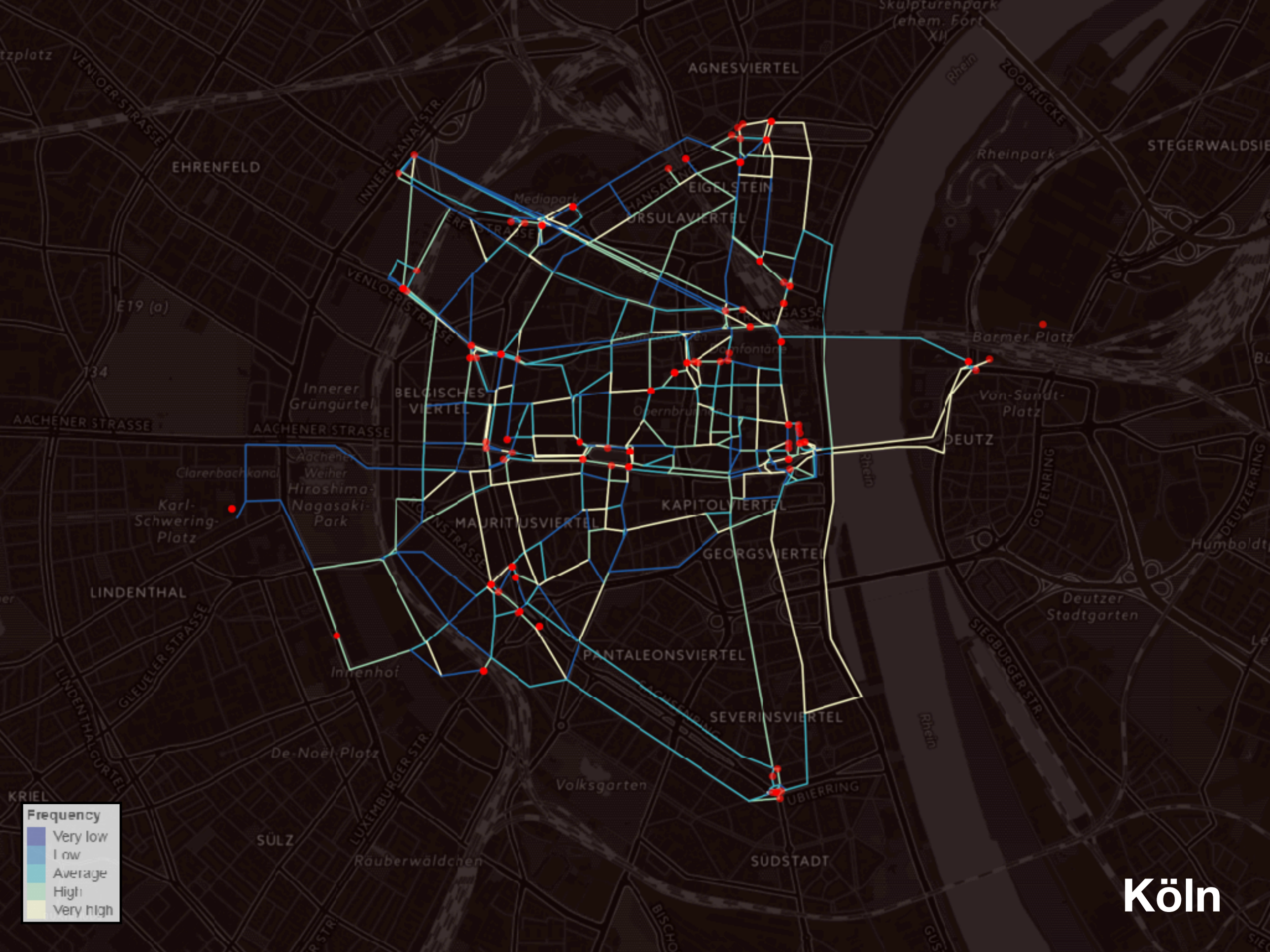
carles@reliabledyanimcs.com
@carles_

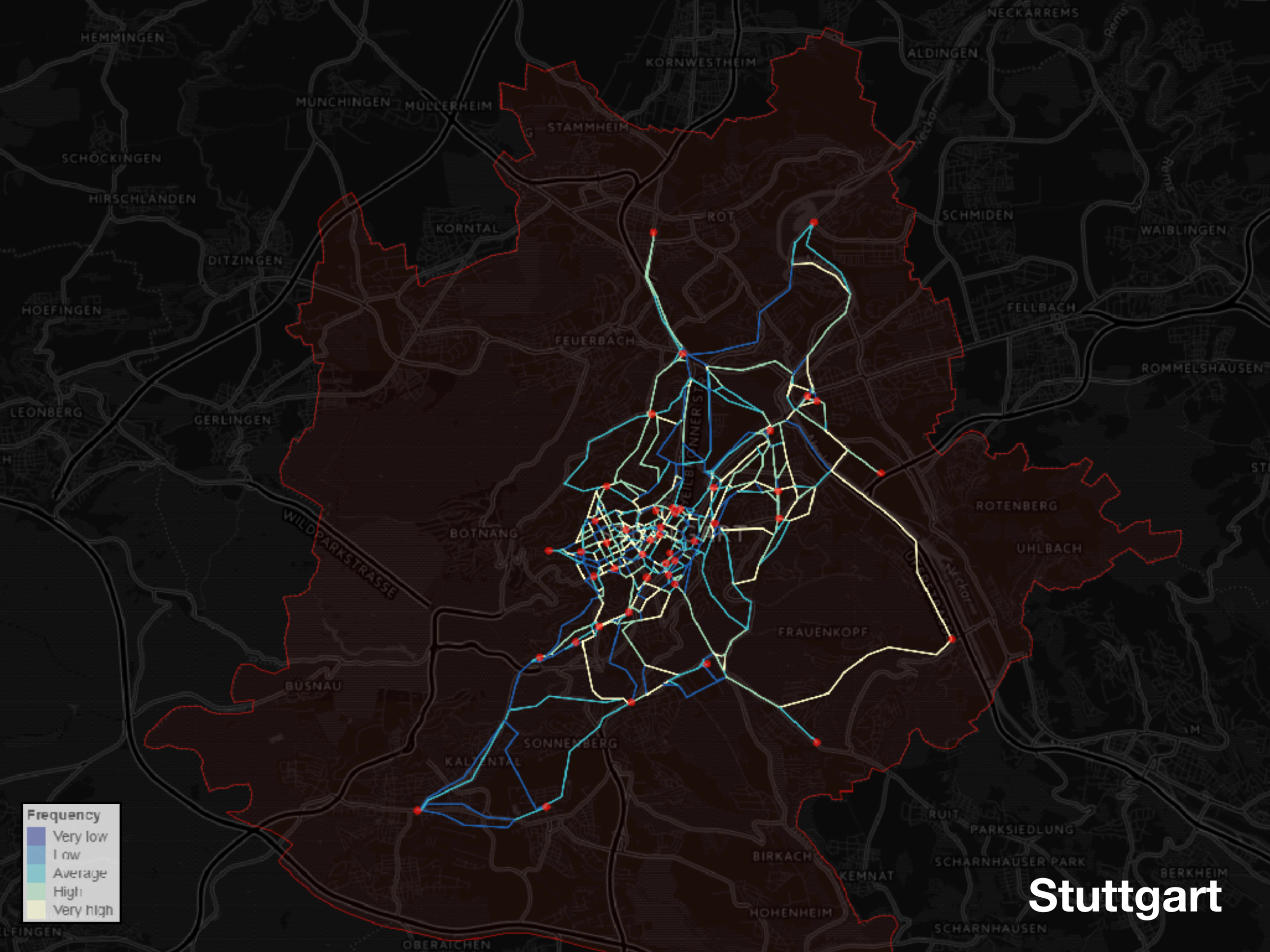

Frequency
Very low
Low
Average
High
Very high
Hamburg


Frequency
Very low
Low
Average
High
Very high
Kassel
SPIEKERSHAUSEN
PHILIPPINENHOF-
WARTEBERG
JUNGFERNKOPF
HARLESHAUSEN
WOLFSANGER-
HASENHECKE
FASANENHOF
NORD-
HOLLAND
KIRCHDITMOLD
WESERTOR
SANDERSHAUSEN
NIESTETAL
BAD
WILHELMSHÖHE
WEHLHEIDEN
SÜDSTADT
BETTENHAUSEN
FORSTFELD
PAPIERFABRIK
SÜSTERFELD-
HELLEBERG
BRASSELSBERG
WALDAU
NORDSHAUSEN
OCHSHAUSEN
LOHFELDEN
OBERZWEHREN
BERGSHAUSEN
CRUMBACH


Frequency
Very low
Low
Average
High
Very high
Berlin


Frequency
Very low
Low
Average
High
Very high
Frankfurt
KALBACH
HARHEIM
MASSENHEIM
GRONAU
KRONBERG
OBERHÖCHSTADT
STEINBACH
SCHWALBACH AM TAUNUS
BAD SODEN AM TAUNUS
ESCHBORN
SULZBACH
LIEDERBACH AM TAUNUS
BERGEN-ENKHEIM
SIEDLUNG HEILSBERG
HANAUER LANDSTR.
HÖCHST
ZEILSHEIM
KRIFTEL
HATTERSHEIM
OKRIFTEL
KELSTERBACH
NEU ISENBURG
ZEPPELINHEIM
OFFENBACH AM MAIN
BIEBER
GRAVENBRUCH
HEUSENSTAMM
WALDHEIM
Nidda
Main


Frequency
Very low
Low
Average
High
Very high
Köln
AGNESVIERTEL
EHRENFELD
EIGELSTEIN
URSULAVIERTEL
BELGISCHES VIERTEL
Innerer Grüngürtel
AACHENER STRASSE
Barmer Platz
Von-Sandt-Platz
DEUTZ
KAPITOLVIERTEL
MAURITIUSVIERTEL
GEORGSVIERTEL
PANTALEONSVIERTEL
SEVERINSVIERTEL
SÜDSTADT
LINDENTHAL
SÜLZ
KRIEL
Rheinpark
Deutzer Stadtgarten
Volksgarten
Rhein
Mediapark
Innenhof
Räuberwäldchen
Hiroshima-Nagasaki-Park
Karl-Schwering-Platz
De-Noël-Platz
Aachener Weiher
Clarenbachkanal
Skulpturenpark (ehem. Fort XII)
Humboldt
E19 (a)
134


Frequency
Very low
Low
Average
High
Very high
Stuttgart
HEMMINGEN
MÜNCHINGEN
MÜLLERHEIM
KORNWESTHEIM
ALDINGEN
NECKARREMS
STAMMHEIM
SCHÖCKINGEN
HIRSCHLANDEN
KORNTAL
ROT
SCHMIDEN
WAIBLINGEN
DITZINGEN
HOEFINGEN
FEUERBACH
FELLBACH
ROMMELSHAUSEN
LEONBERG
GERLINGEN
BOTNANG
WILDPARKSTRASSE
ROTENBERG
UHLBACH
FRAUENKOPF
BÖBLINGEN
SONNENBERG
KALTENTAL
RUIT
PARKSIEDLUNG
SCHARNHAUSER PARK
BERKHEIM
BIRKACH
KEMNAT
HOHENHEIM
SCHARNHAUSEN
OBERAICHEN

Reproducibility

+

Application in a case

Extended… but How?

- Archive raw, pre/post process and final data
- Rewrite code with the tidyverse principles
- Benchmark reading and munging code
- Extend the analysis
- Documentation

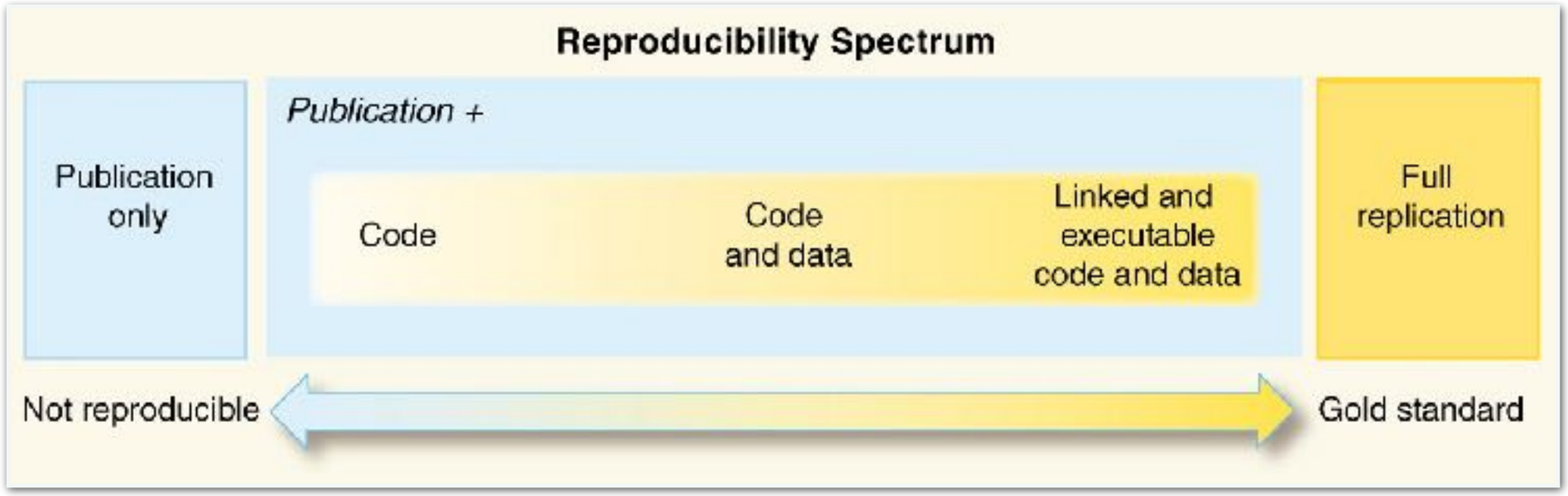
Reproducibility… but why?

"*Reproducibility is the ability to take the code and data from a previous publication, rerun the code and get the same results*"

https://simplystatistics.org/2017/03/02/rr-glossy/

- Make it easier for your future self. Data might be expanded in the future!
- Review the basis that lead to decision
- Transparency
- Avoid manual errors
- Learn new skills
- Science! Reproducibility vs. Replication

Science!



"***Replication*** *This is the act of repeating an entire study, independently of the original investigator without the use of original data (but generally using the same methods).*

Reproducibility *A study is reproducible if you can take the original data and the computer code used to analyze the data and reproduce all of the numerical findings from the study.*"

https://simplystatistics.org/2016/08/24/replication-crisis/

Reproducibility gone wrong

Growth in a Time of Debt

By CARMEN M. REINHART AND KENNETH S. ROGOFF*

American Economic Review: Papers & Proceedings 100 (May 2010): 573–578
http://www.aeaweb.org/articles.php?doi=10.1257/aer.100.2.573

```
# 23-class classification problem

skf=StratifiedKFold(labels,8)

if trainsvm:
    pred=N.zeros(len(labels))
    for train,test in skf:
        clf=LinearSVC()
        clf.fit(data[train],labels[train])
        pred[test]=clf.predict(data[test])
```

Results:
93% accuracy

```
data[:,train]
data[:,test]
```

Results:
53% accuracy

http://www.russpoldrack.org/2013/02/anatomy-of-coding-error.html

(Some)

Principles of reproducibility

Data
Code
Environment
Documentation

Soft

- How important is the output of the analysis?
- Team effort vs. cowboy coder
- How much time should we invest to make it till some degrees reproducible?

Data Management

Archiving short vs long term

Version Control services

Academia services?

Private internal repository

(avoid silos, AirBnB case)

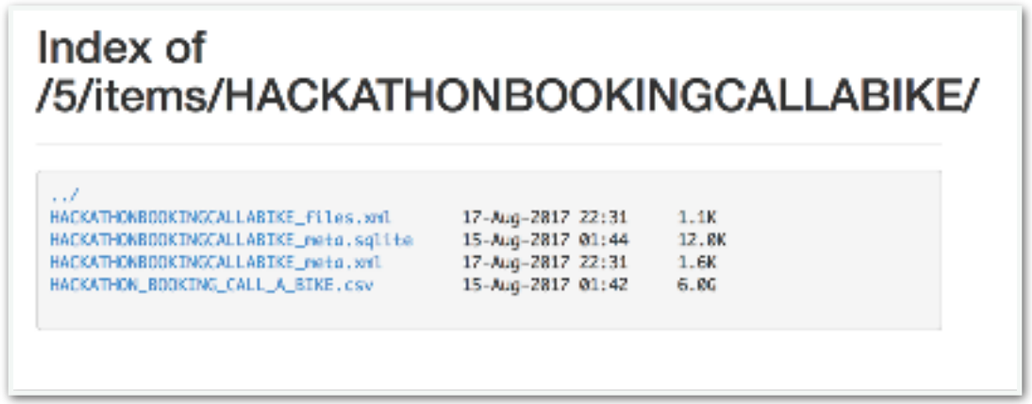
Data archive services

Archive.org, DataHub, Zenovo, …

Case

1. Corrupt CSV file after unzipping

   Had to upload the raw CSV file (6Gb) to archive.org to avoid unzipping problems



2. API

   External computations save intermediate results, before & after the API.

Case

API key **never** hardcoded!
Two solutions

1. Control your *.Renviron*

2. Check out the package “secret" by *Gábor Csárdi [aut, cre], Andrie de Vries [aut]*

R Software

Software

The environment is R

Always include session information in the documentation.

```
sessionInfo()
```

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.4.1  backports_1.1.0 magrittr_1.5    rprojroot_1.2
##  [5] tools_3.4.1     htmltools_0.3.6 yaml_2.1.14     Rcpp_0.12.13
##  [9] stringi_1.1.5   rmarkdown_1.7   knitr_1.17      stringr_1.2.0
## [13] digest_0.6.12   evaluate_0.10.1
```

[Advanced]
Package "*containerit*"
Automatically archiving reproducible studies with docker.

- Use Rmarkdown, jupiter notebooks or any other form of literate programming

- Use always relative paths in favour of absolute paths

Relative

```
file.path("./data/BOOKING_CALL_A_BIKE.RData")
```

```
## [1] "./data/BOOKING_CALL_A_BIKE.RData"
```

Absolute

```
library(tools)
file_path_as_absolute(x = "./data/BOOKING_CALL_A_BIKE.RData")
```

```
## [1] "/Users/RDynamics/Documents/R_folder/bike_sharing/data/BOOKING_CALL_A_BIKE.RData"
```

- Package versioning

    1. Packrat
    2. Checkpoint

```
library(checkpoint)
checkpoint("2017-07-01")
```

- Out of scope but important too

  1. Unit testing (code and data)
  2. Code coverage
  3. Continuous Integration / Continuous Deployment
  4. …

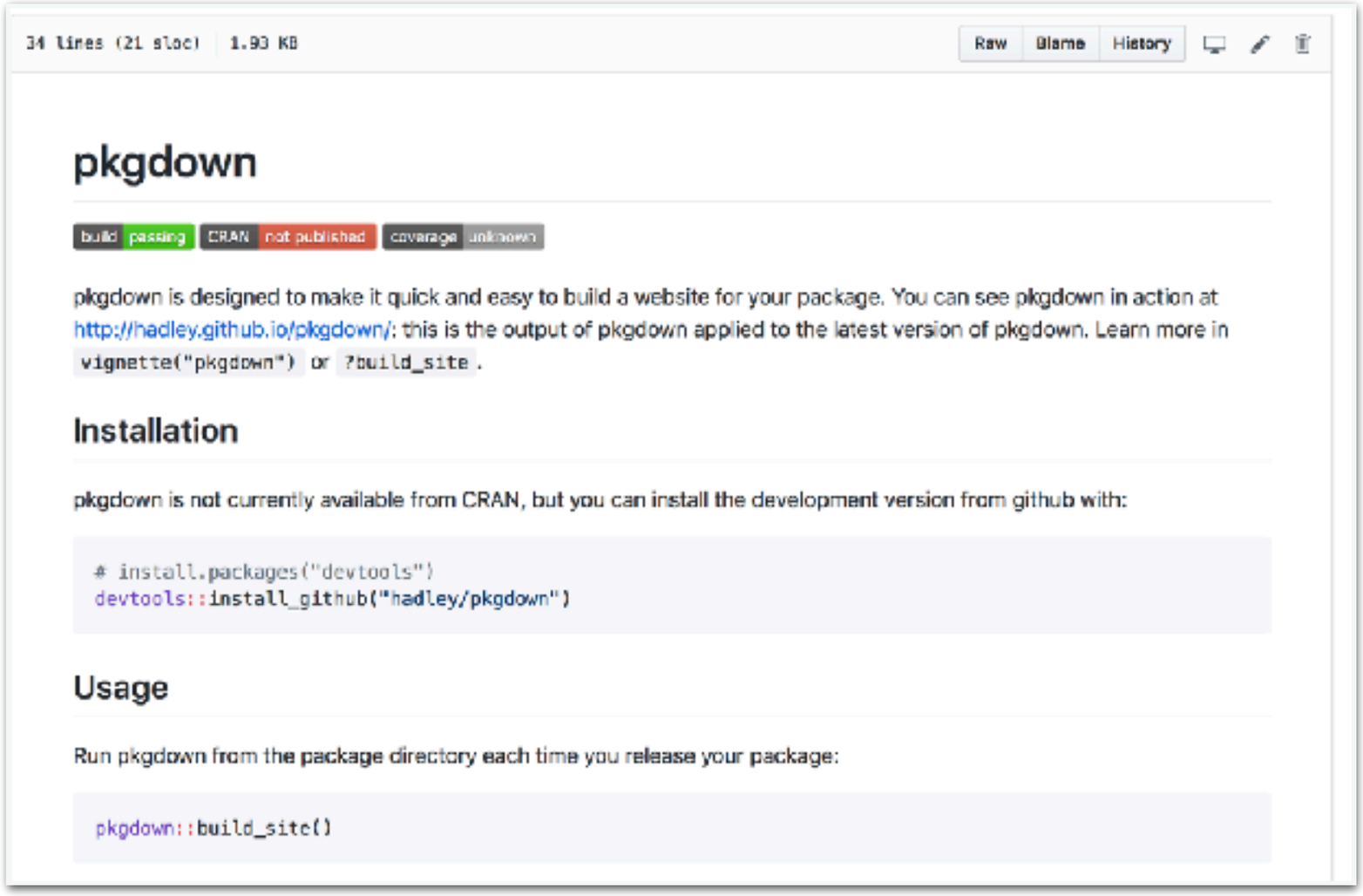
Collaboration

Collaboration

- Version control platforms

Check legal! How delicate is your data?

Rich README.md with an overview of the analysis



- Folder structure

Box 3. Project layout

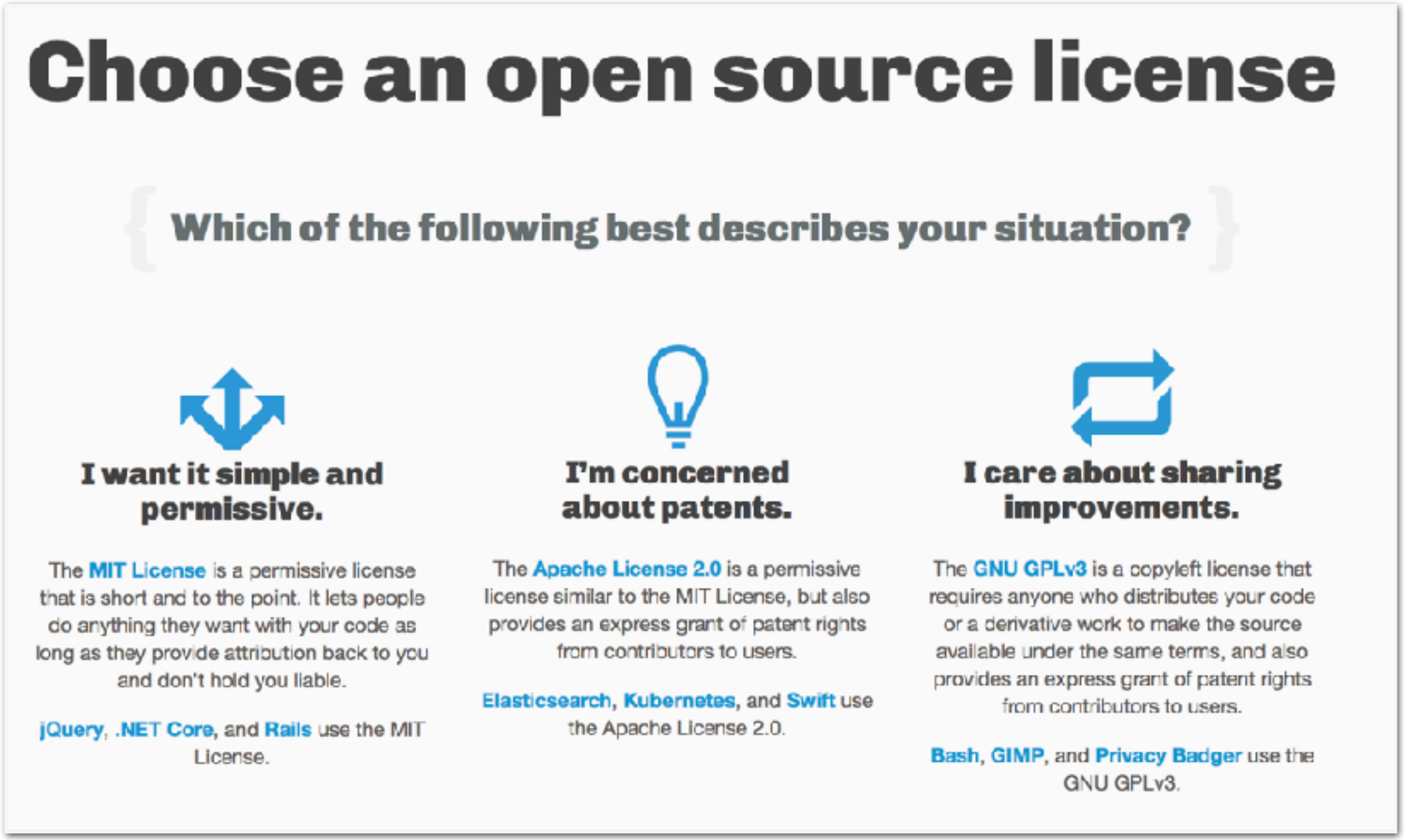
```
.
|-- CITATION
|-- README
|-- LICENSE
|-- requirements.txt
|-- data
|    |-- birds_count_table.csv
|-- doc
|    |-- notebook.md
|    |-- manuscript.md
|    |-- changelog.txt
|-- results
|    |-- summarized_results.csv
|-- src
|    |-- sightings_analysis.py
|    |-- runall.py
```

”Good enough practices in scientific computing"

- Naming conventions

5 minutes video by Rasmus Bååth - User2017!

- Licensing


Choose an open source license
Which of the following best describes your situation?
I want it simple and permissive.
The MIT License is a permissive license that is short and to the point. It lets people do anything they want with your code as long as they provide attribution back to you and don't hold you liable.
jQuery, .NET Core, and Rails use the MIT License.
I'm concerned about patents.
The Apache License 2.0 is a permissive license similar to the MIT License, but also provides an express grant of patent rights from contributors to users.
Elasticsearch, Kubernetes, and Swift use the Apache License 2.0.
I care about sharing improvements.
The GNU GPLv3 is a copyleft license that requires anyone who distributes your code or a derivative work to make the source available under the same terms, and also provides an express grant of patent rights from contributors to users.
Bash, GIMP, and Privacy Badger use the GNU GPLv3.

Out of scope…but important!

- Calculations were done in Azure Data Science Virtual Machine on CentOS.

Since I didn't use docker...the GIS packages have *funny* Unix library dependencies.

```
sudo yum update
sudo yum install gdal
sudo yum install proj-devel
sudo yum install proj-nad
sudo yum install proj-epsg
sudo yum install geos-devel
```

- MRAN is set up to 2017-07-01

```
library(checkpoint)
checkpoint("2017-07-01")
```

- Reproducible presentation

  Xaringan (RMarkdown presentation)

  Rpres from RStudio

  http://rmarkdown.rstudio.com/ioslides_presentation_format.html

To do

- Reproducible presentation: this presentation is not reproducible!

  Xaringan (RMarkdown presentation)

  Rpres from RStudio

  Slidify

- The code of the analysis is not publish (yet) on GitHub

Summary

Data
Code
Environment
Documentation

Thank you!
Q&A

Reliable Dynamics

Carles CG
Data Scientist & co-founder

carles@reliabledyanimcs.com
@carles_

Extra

More sources

- ["A Simple Explanation for the Replication Crisis in Science"](https://simplystatistics.org/2015/12/11/instead-of-research-on-reproducibility-just-do-reproducible-research/)

- ["Good enough practices in scientific computing"](http://journals.plos.org/ploscompbiol/article/file?id=10.1371/journal.pcbi.1005510&type=printable)

- Package [‘rrtools'](https://github.com/benmarwick/rrtools/blob/master/README.md).

- [Reproducibility guide](https://github.com/ropensci/reproducibility-guide)