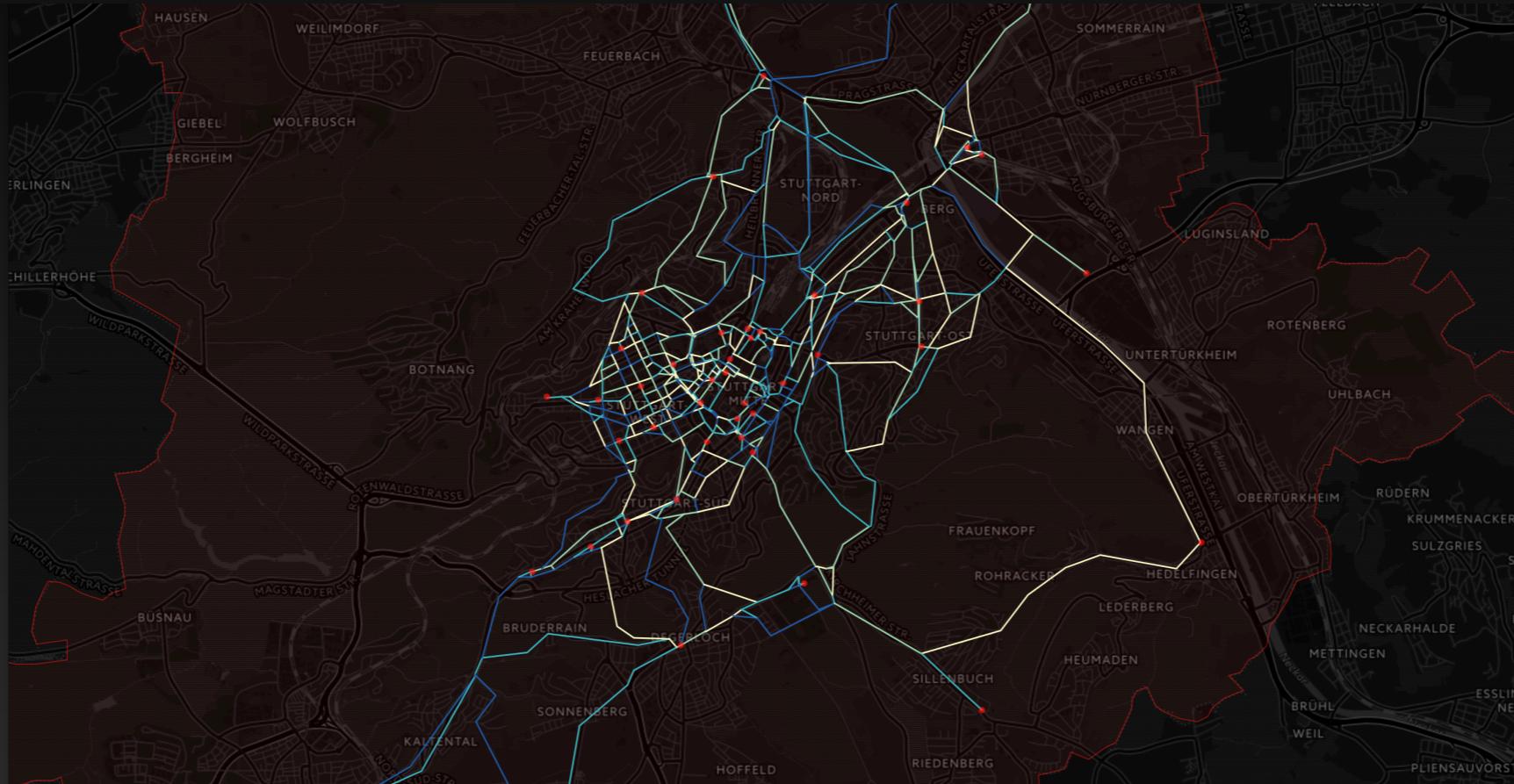


# Reproducible Bike Ride in Hamburg

*...or elsewhere in Europe*



**Carles CG**  
*Data Scientist*

**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 spplanr package very helpful. It converts a series of overlapping lines and aggregates their values for overlapping segments.

**THE MAP**

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.

As a cycling enthusiast 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 pedestrians. It is amazing to see that the data set can proof some of my hypothesis about the current state of the bicycle 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 separates 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.



```
#####
## PROCESS DATA
#####

# Load packages
x = c("sqldf", "dplyr", "sp", "rgdal", "stplanr", "reshape2", "rmapshaper", "leaflet", "colorBrewer")
lapply(x, require, character.only = TRUE)

# import bike rentals
mydata = read.csv("LOCATION_BOOKING_CALL_A_DATE.csv",
                  sql = "SELECT * FROM Rte 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 lat/labs from stations
station = read.csv("STATION_RENTAL_ZONE_CALL_A_BIKE.csv", sep = ",")  

station = filter(station, CITY == "Hamburg")

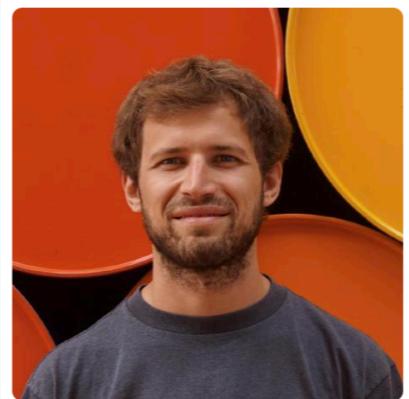
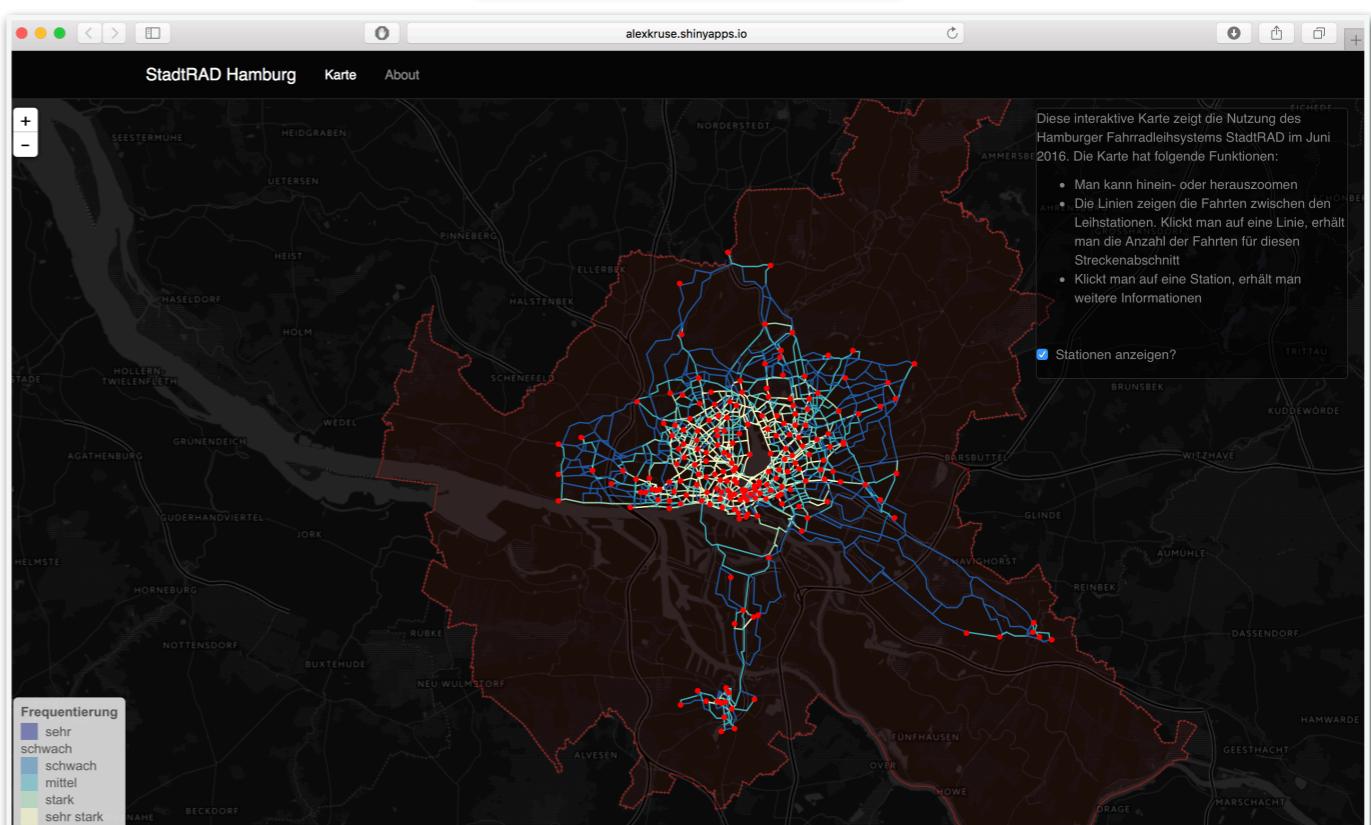
# merge station coordinates with bike rentals
mydata = merge(mydata, station, by.x = "min", by.y = "RENTAL_ZONE_GROUP", all.x = TRUE)
mydata = merge(mydata, station, by.x = "max", by.y = "RENTAL_ZONE_GROUP", all.x = TRUE)

# count bike rides for each route (combine lat/lon)
mydata$start = paste0(mydata$RENTAL_ZONE_X_COORDINATE.x, mydata$RENTAL_ZONE_Y_COORDINATE.x, sep = ",")  

mydata$dest = paste0(mydata$RENTAL_ZONE_X_COORDINATE.y, mydata$RENTAL_ZONE_Y_COORDINATE.y, sep = ",")  

mydata = mydata %>% group_by(start, dest) %>% summarise(count = n())

# split lat/lon into two columns
mydata$id = rownames(mydata)
mydata$lat = cbind(mydata$id, c("id", "count"))
test = data.frame(mydata[, c("id", "count")])
mydata = cbind(test, test)
mydata = select(mydata, id, x, id, count)
colnames(mydata) = c("lat", "lon", "id", "count")
```



Alex Kruse  
kruse-alex

# 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.



PERSPECTIVE

## Good enough practices in scientific computing

Greg Wilson<sup>1\*</sup>, Jennifer Bryan<sup>2</sup>, Karen Cranston<sup>3</sup>, Justin Kitzes<sup>4</sup>, Lex Nederbragt<sup>5</sup>, Tracy K. Teal<sup>6</sup>

**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.  
\* [gwilson@software-carpentry.org](mailto:gwilson@software-carpentry.org)

OPEN ACCESS Freely available online

Community Page

## Best Practices for Scientific Computing

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

**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



The R Series

## Implementing Reproducible Research



Edited by  
**Victoria Stodden**  
**Friedrich Leisch**  
**Roger D. Peng**

CRC Press  
Taylor & Francis Group  
A CHAPMAN & HALL BOOK

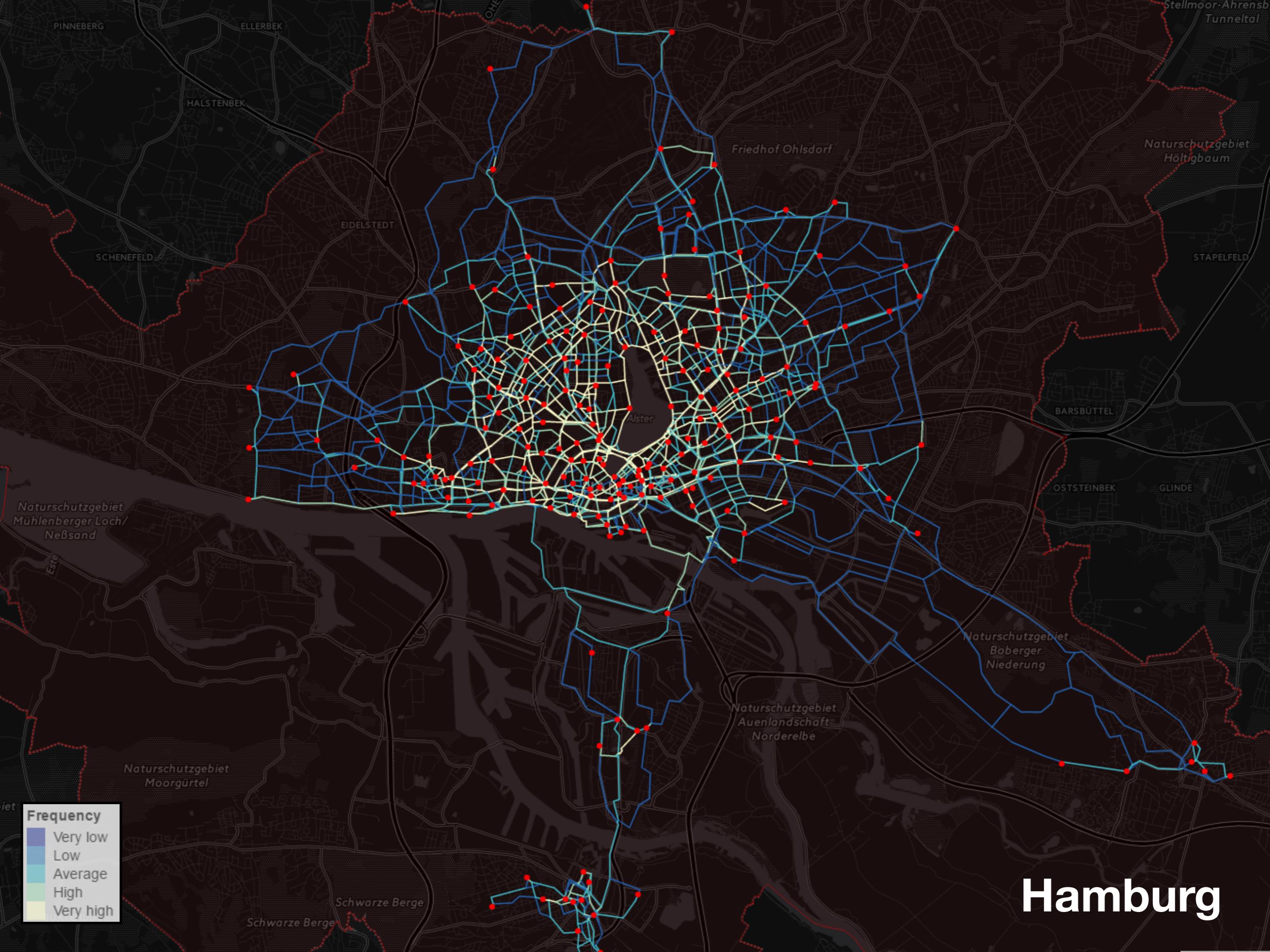
# Who am I?

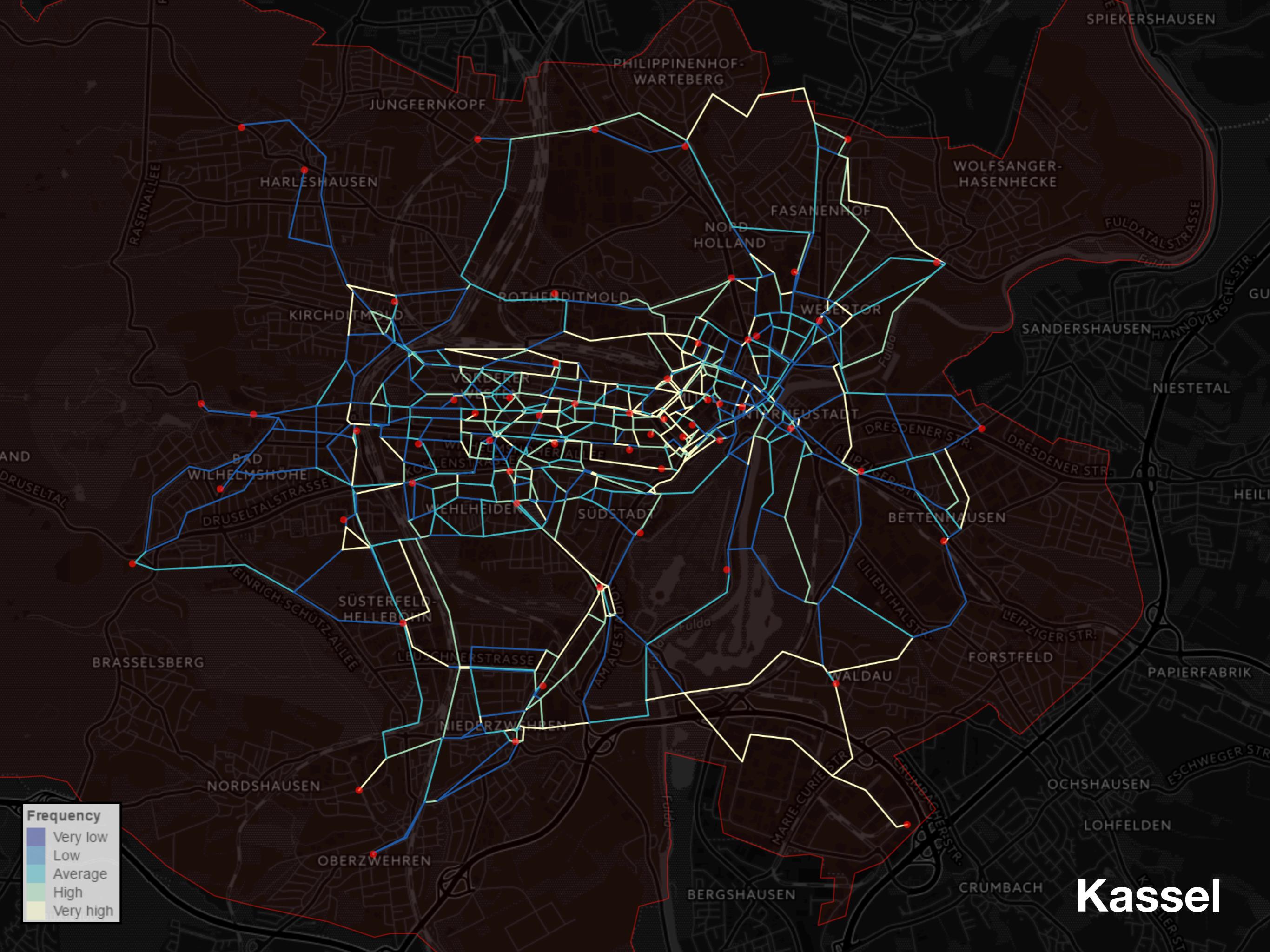
**Reliable  
Dynamics**

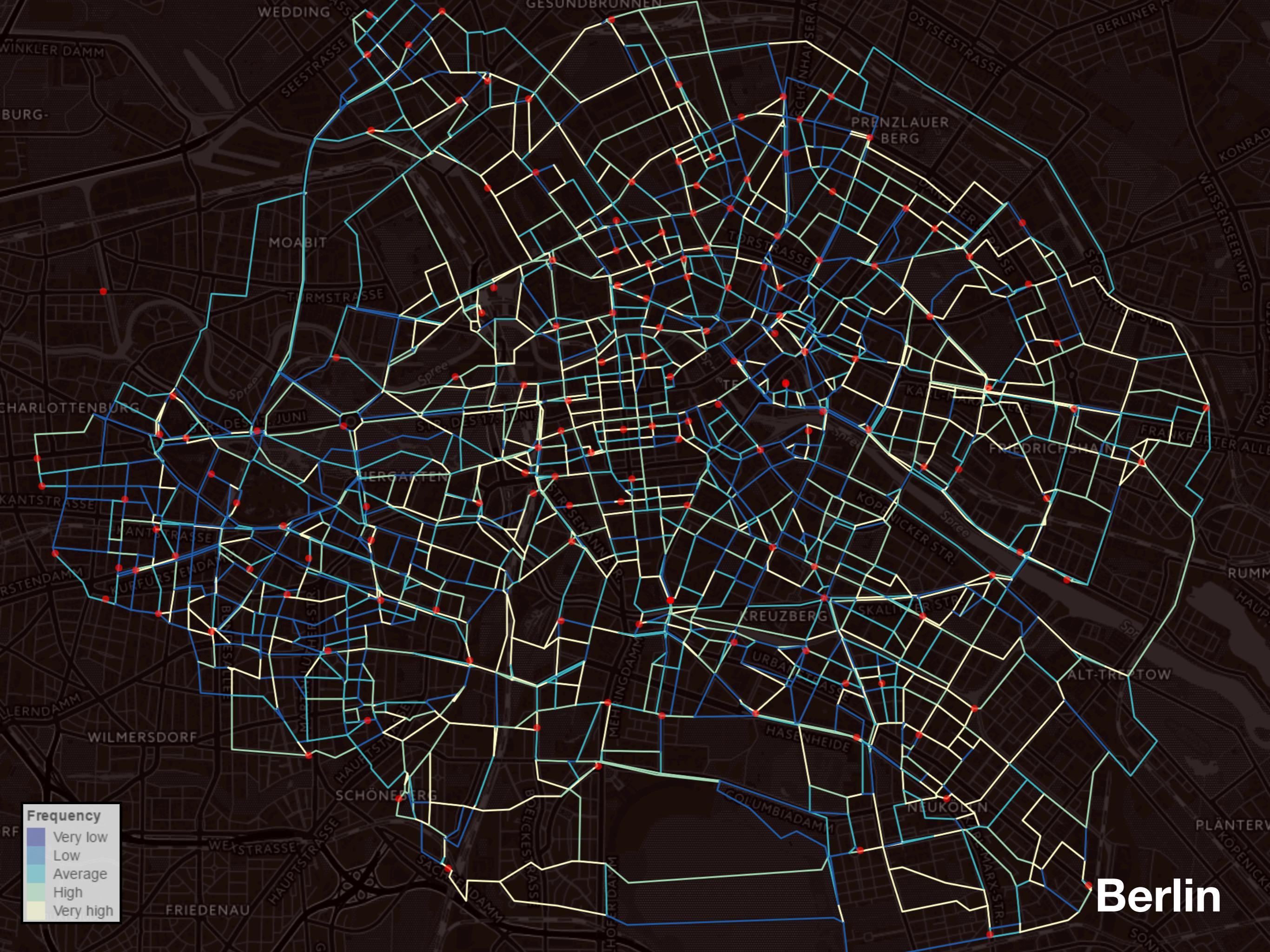


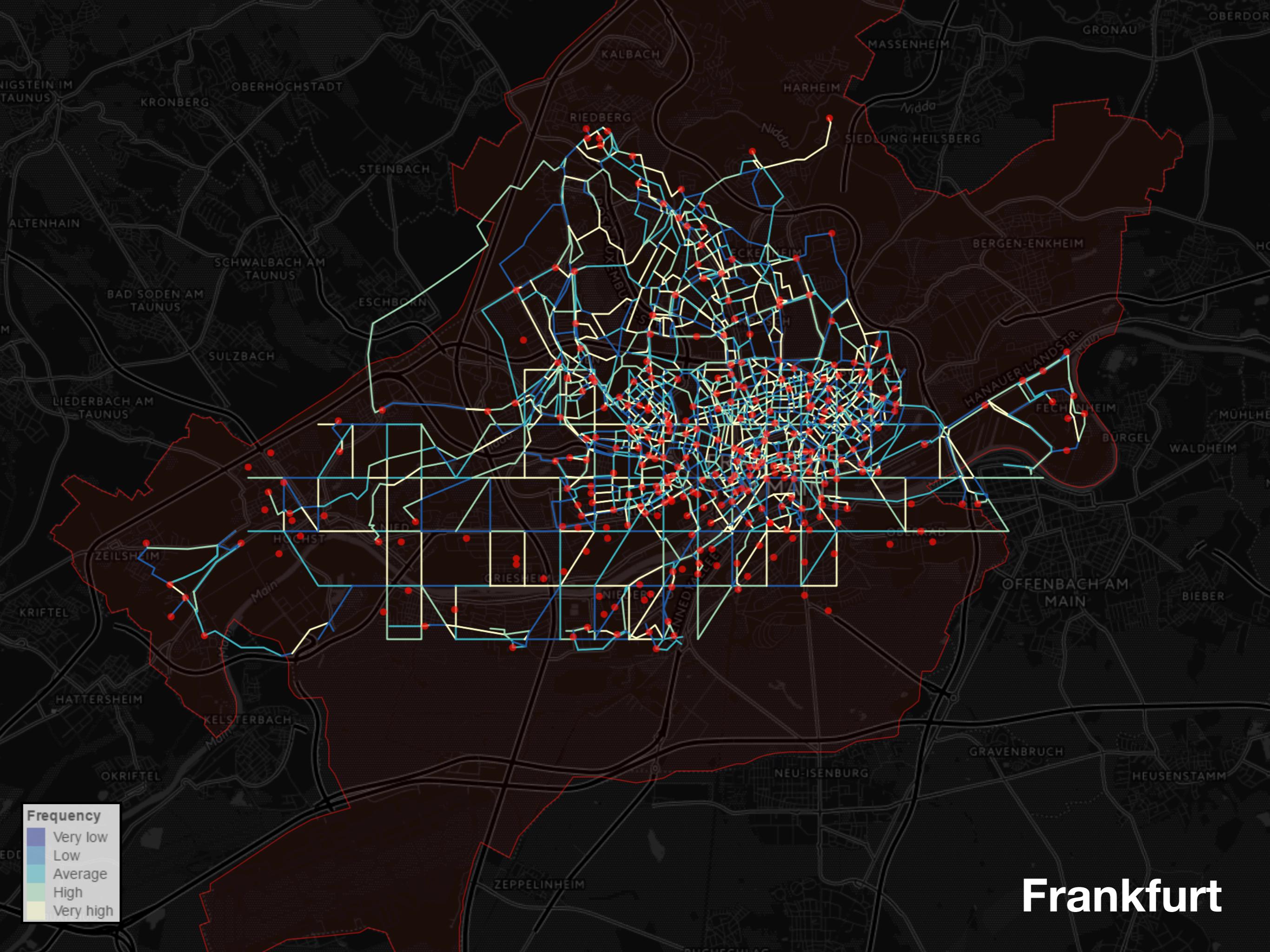
*Carles CG  
Data Scientist & Consultant*

[carles@reliabledyanimcs.com](mailto:carles@reliabledyanimcs.com)  
[@carles\\_](https://twitter.com/carles_)



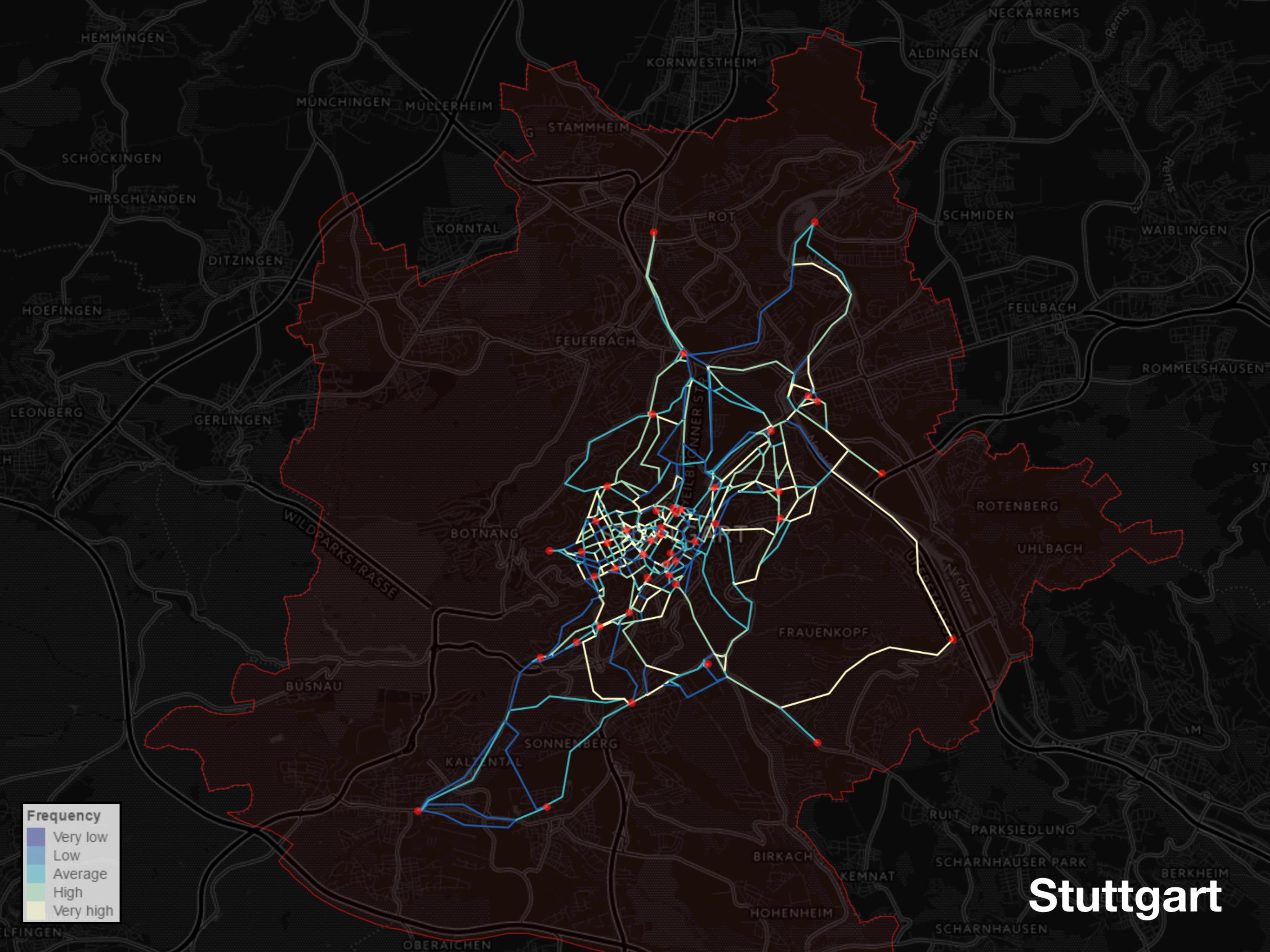








# Köln



# **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

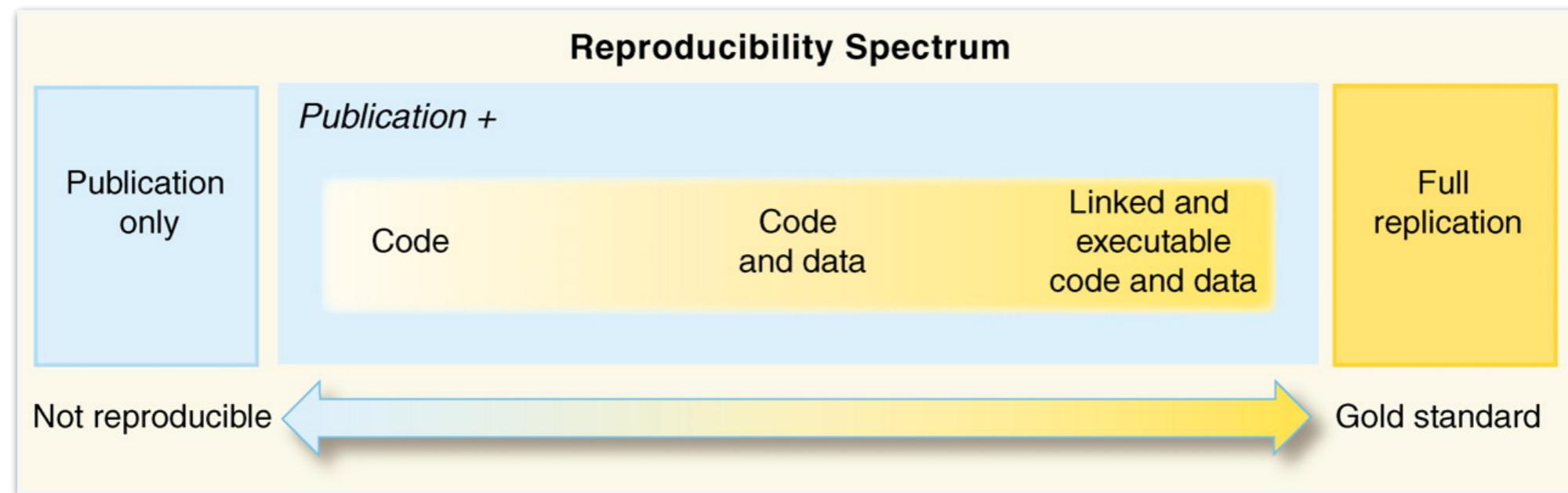
# Reproducibility... but why?

*“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."

# 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])

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

**Results:**  
93% accuracy

**Results:**  
53% accuracy

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

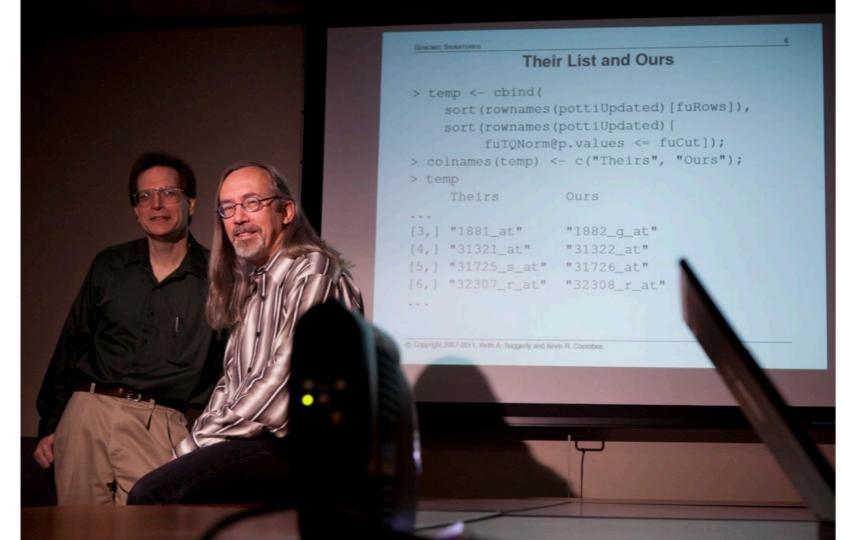
The New York Times

RESEARCH

## How Bright Promise in Cancer Testing Fell Apart

By GINA KOLATA JULY 7, 2011

f t p e 75



GENOMIC SIGNATURES Their List and Ours

	Theirs	Ours
[3, ]	"1881_at"	"1882_g_at"
[4, ]	"31321_at"	"31322_at"
[5, ]	"31725_g_at"	"31726_at"
[6, ]	"32307_r_at"	"32308_r_at"

© Copyright 2007-2011, Keith A. Baggerly and Kevin R. Coombes

Keith Baggerly, left, and Kevin Coombes, statisticians at M. D. Anderson Cancer Center, found flaws in research on tumors. Michael Stravato for The New York Times

*(Some)*  
Principles of  
reproducibility

# Data Code Environment Documentation

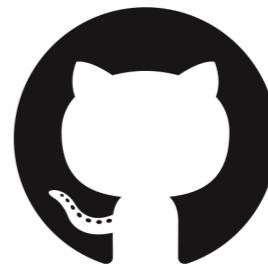
## Soft

- 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)*

[airbnb / knowledge-repo](#)

Data archive services  
*Archive.org, DataHub, Zenodo, ...*



# Case

## 1. Corrupt CSV file after unzipping

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

Index of  
/5/items/HACKATHONBOOKINGCALLABIKE/

.. /		
HACKATHONBOOKINGCALLABIKE_files.xml	17-Aug-2017 22:31	1.1K
HACKATHONBOOKINGCALLABIKE_meta.sqlite	15-Aug-2017 01:44	12.0K
HACKATHONBOOKINGCALLABIKE_meta.xml	17-Aug-2017 22:31	1.6K
HACKATHON_BOOKING_CALL_A_BIKE.csv	15-Aug-2017 01:42	6.0G

## 2. API

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

The screenshot shows the CycleStreets.net API (v2) interface. At the top, there's a green banner with the logo 'Cyclestreets.net' featuring a cyclist silhouette and a red 'BETA' badge. Below the banner are navigation links: 'Blog', 'My area', 'Journey planner', 'Photo', 'Help', 'Journey Planner', 'How it works', 'OSM', 'Developers', 'API', 'Photomap', and 'Photo licensing'. A large red header 'CycleStreets API (v2)' is centered. Below it, a search bar says 'Details for: \* Introduction' with a dropdown arrow and a 'Go!' button. To the right of the search bar is a link 'API overview'.

# Case

API key **never** hardcoded!

Two solutions

1. Control your *.Renviron*

## .Renviron

To set global variables and or set API constants i.e.

```
Sys.getenv('CYCLESTREET')
```

```
# Execute the command at the R console  
file.edit('~/.Renviron')
```

```
# And then add your keys  
CYCLESTREET=this_is_my_ip_secret
```

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

Andrie de Vries Retweeted



David Smith @revodavid · Jul 6

Don't put API keys or other secure data in R scripts or packages. Use the "secret" package instead — [@RevoAndrie](#) at [#user2017 #rstats](#)

Use the `secret` package

- On CRAN now
  - Published 2017-06-17
  - <https://cran.r-project.org/package=secret>
  - Maintainer: Gábor Csárdi
- Functionality
  - Create a vault
  - Add users
  - Encrypt and decrypt secrets
  - Share secrets

# 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.



Edzer Pebesma

@edzerpebesma

Follow

Daniel Nüst [@nordholmen](#) presenting containerit, creates a docker img from an R session to archive reproducibly [@o2r\\_project](#) [@cboettig](#)

- Use Rmarkdown, jupyter notebooks or any other form of literate programming
- Use always relative paths in favour of absolute paths



```
Relative
file.path("./data/BOOKING_CALL_ABIKE.RData")
## [1] "./data/BOOKING_CALL_ABIKE.RData"

Absolute
library(tools)
file_path_as_absolute(x = "./data/BOOKING_CALL_ABIKE.RData")
## [1] "/Users/RDynamics/Documents/R_folder/bike_sharing/data/BOOKING_CALL_ABIKE.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

The screenshot shows a GitHub README page for the `pkgdown` package. At the top, it displays statistics: 34 lines (21 sloc) and 1.93 KB. To the right are buttons for Raw, Blame, History, and icons for copy/paste and delete. Below this, the title `pkgdown` is shown in bold. Underneath the title are status badges: build (passing), CRAN (not published), and coverage (unknown). A descriptive paragraph explains that `pkgdown` is designed to make it quick and easy to build a website for your package, with a link to its live demo at <http://hadley.github.io/pkgdown/>. It also provides instructions for installation and usage.

34 lines (21 sloc) | 1.93 KB

Raw Blame History

## pkgdown

build passing CRAN not published coverage unknown

pkgdown is designed to make it quick and easy to build a website for your package. You can see `pkgdown` in action at <http://hadley.github.io/pkgdown/>: this is the output of `pkgdown` applied to the latest version of `pkgdown`. Learn more in `vignette("pkgdown")` or `?build_site`.

### Installation

`pkgdown` is not currently available from CRAN, but you can install the development version from github with:

```
# install.packages("devtools")
devtools::install_github("hadley/pkgdown")
```

### Usage

Run `pkgdown` from the package directory each time you release your package:

```
pkgdown::build_site()
```

- 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

The current state of naming conventions in R - UseR 2017 - YouTube



<https://www.youtube.com/watch?v=Pv5dfsHBBKE>

Jul 14, 2017 - Uploaded by rasmusab

This is a lightning talk I held at the UseR 2017 conference in Brussels. I talk about the current state of naming ...

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.

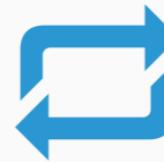
[jQuery](#), [.NET Core](#), and [Rails](#) use the MIT License.



## 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](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](mailto:carles@reliabledyanimcs.com)  
[@carles\\_](https://twitter.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>)