CODECHECK certificate 2025-027

https://doi.org/10.5281/zenodo.17059066

Item	Value
Title of checked publication	ClockBoard: A zoning system for urban analysis
Author(s)	Robin Lovelace , Martijn Tennekes , Dustin Carlino
Reference	https://doi.org/10.5311/JOSIS.2022.24.172
Codechecker(s)	Subhan Ali
Date of check	2025-09-04 12:00:00
Summary	Successfully reproduced all main figures from the ClockBoard paper using the zonebuilder R package. All core functions (zb_zone, zb_doughnut, zb_segment, zb_quadrat) work as described. Package installs correctly from CRAN and GitHub. Interactive mapping requires coordinate transformation but works properly. Parameter flexibility confirmed through additional tests.
Repository	https://github.com/codecheckers/zonebuilder

Table 1: CODECHECK summary

Output	Comment	Size (b)
`codecheck/outputs/figure2_clockboard_basic.png`	Reproduction of Figure 2: Basic ClockBoard zoning system for London	141874
`codecheck/outputs/figure3a_doughnuts.png`	Reproduction of Figure 3a: Doughnut zones (concentric rings)	74454
`codecheck/outputs/figure3b_segments.png`	Reproduction of Figure 3b: Segment zones (clock divisions)	62981
`codecheck/outputs/figure4_rectangular_grid.png`	Reproduction of Figure 4: Rectangular grid comparison	47615
`codecheck/outputs/test_3_rings.png`	Parameter test: 3 rings instead of default configuration	88255
`codecheck/outputs/test_8_segments.png`	Parameter test: 8 segments instead of default 12	116346
`codecheck/outputs/test_interactive_leaflet.png`	Screenshot of interactive leaflet map showing zoning system with geographic context	2011264

Table 2: Summary of output files generated

Summary

This codecheck successfully reproduces the key figures and functionality described in the ClockBoard paper. The zonebuilder R package installs correctly from both CRAN and GitHub, and all core functions work as documented. The reproduction demonstrates the flexibility of the ClockBoard zoning system through various parameter configurations and confirms the package's utility for urban analysis applications.

CODECHECKER notes

The GitHub repository for the zonebuilder package is available at https://github.com/zonebuilders/zonebuilder. This check is based on the commit `67dafb6a1294dd4416ad36d17926ea3c44cf51de`.

The code was written in R and uses several key dependencies including sf for spatial operations, dplyr for data manipulation, and leaflet for interactive mapping. The package provides a comprehensive set of functions for creating clock-based zoning systems.

Installation and Setup

I installed the zonebuilder package from CRAN using:

```
install.packages("zonebuilder")
```

I also tested the development version from GitHub:

```
remotes::install_github("zonebuilders/zonebuilder")
```

Both installations completed successfully without dependency conflicts.

Source of Reproduction Code

The paper itself does not provide explicit code for reproducing the figures. Therefore, I adapted the reproduction code from:

1. Primary source: Example code in the package's GitHub README
2. Function documentation and help files (`?zb_zone`, `?zb_doughnut`, etc.)
3. The methodology described in the paper

The README provides clear examples using `london_c()` and `london_a()` with functions like `zb_zone()`, `zb_doughnut()`, `zb_segment()`, and `zb_quadrat()`, which I used as the basis for reproducing the paper's figures.

Reproduction Process

I systematically reproduced each figure from the paper using the following exact function calls:

Figure 2 (Basic ClockBoard)

```
library(zonebuilder)
library(tmap)

# Using built-in London functions from zonebuilder package
zones_basic <- zb_zone(london_c(), london_a())

# Create and save plot using zb_plot
png("codecheck/outputs/figure2_clockboard_basic.png", width = 800, height = 600)
zb_plot(zones_basic)
dev.off()
```

Figure 3a (Doughnut zones)

```r
# Create concentric rings only (no radial segments)
zones_doughnut <- zb_doughnut(london_c(), london_a(), n_circles = 5)

# Create and save plot
png("codecheck/outputs/figure3a_doughnuts.png", width = 800, height = 600)
zb_plot(zones_doughnut)
dev.off()
```

Figure 3b (Segment zones)

```r
# Create radial segments only (no concentric rings)
zones_segments <- zb_segment(london_c(), n_segments = 12)

# Create and save plot
png("codecheck/outputs/figure3b_segments.png", width = 800, height = 600)
zb_plot(zones_segments)
dev.off()
```

Figure 4 (Rectangular grid comparison)

```r
# Create rectangular grid
zones_grid <- zb_quadrat(london_a(), ncol = 10)

# Create and save plot
png("codecheck/outputs/figure4_rectangular_grid.png", width = 800, height = 600)
plot(zones_grid)
dev.off()
```

Parameter Flexibility Testing

I conducted additional tests to verify the system's flexibility:

Test: 8 segments configuration

```r
# Changed from default 12 to 8 segments
zones_8seg <- zb_zone(london_c(), london_a(), n_segments = 8)

png("codecheck/outputs/test_8_segments.png", width = 800, height = 600)
zb_plot(zones_8seg)
dev.off()
```

Test: 3 rings configuration

```r
# Modified the default to 3 rings
zones_3rings <- zb_zone(london_c(), london_a(), n_circles = 3)

png("codecheck/outputs/test_3_rings.png", width = 800, height = 600)
zb_plot(zones_3rings)
dev.off()
```

Test: Interactive mapping

```r
library(sf)
library(leaflet)

# Create zones with custom parameters for interactive display
zones_interactive <- zb_zone(london_c(), london_a(), n_circles = 4, n_segments = 8)

# Transform to WGS84 for leaflet compatibility
zones_lonlat <- st_transform(zones_interactive, 4326)

# Create interactive map (opens in browser)
leaflet(zones_lonlat) %>%
  addTiles() %>%
  addPolygons()

# Screenshot saved manually as test_interactive_leaflet.png
```

Manifest files

figure2_clockboard_basic.png

Comment: Reproduction of Figure 2: Basic ClockBoard zoning system for London

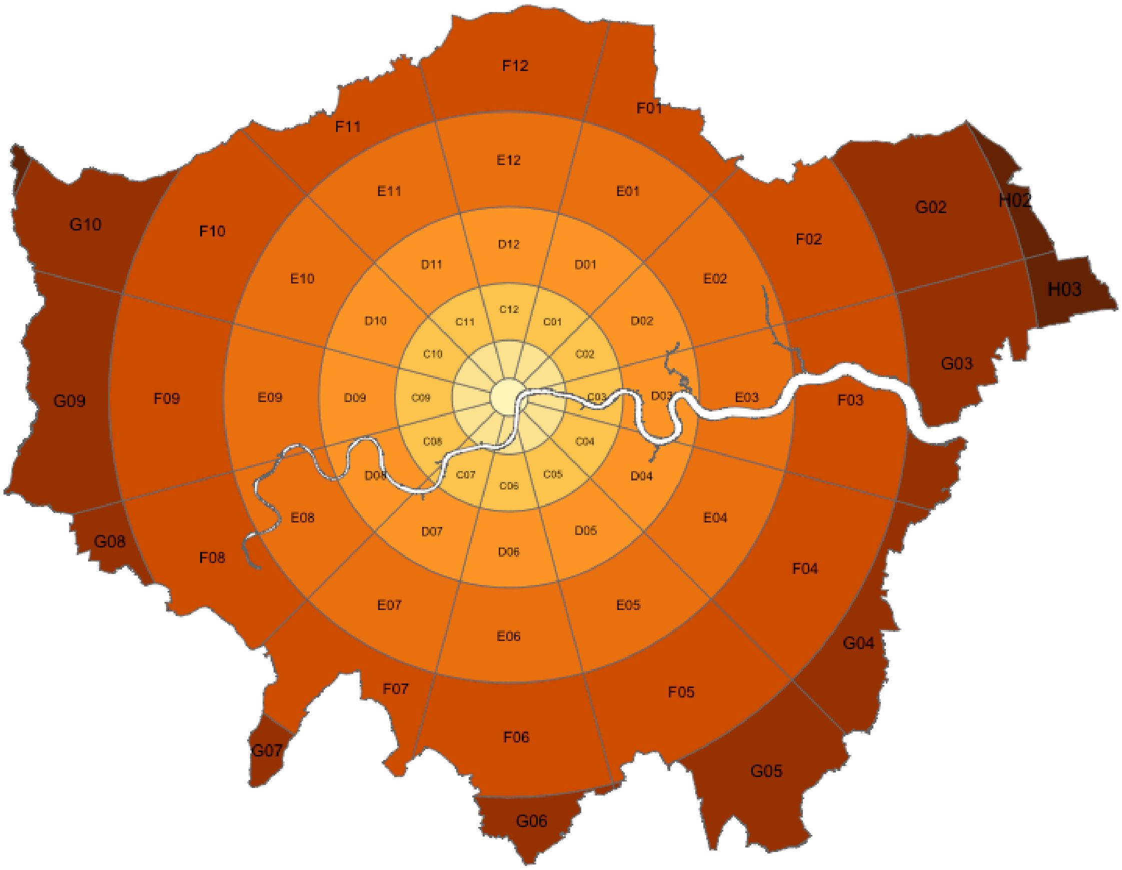


figure3a_doughnuts.png

Comment: Reproduction of Figure 3a: Doughnut zones (concentric rings)

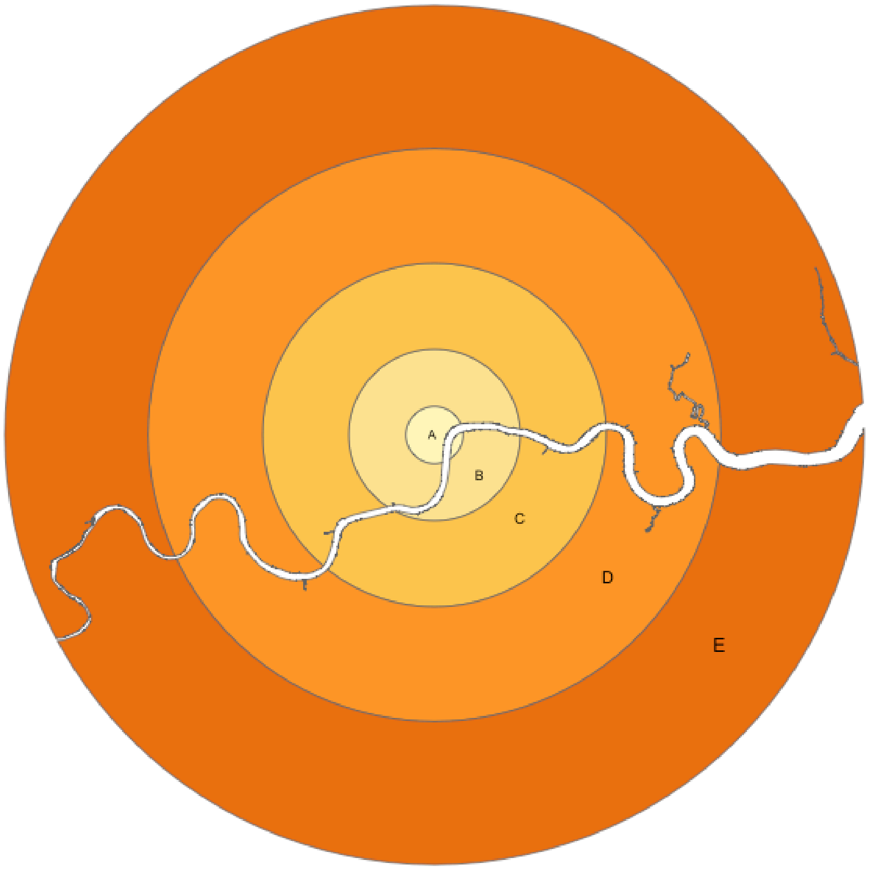


figure3b_segments.png

Comment: Reproduction of Figure 3b: Segment zones (clock divisions)

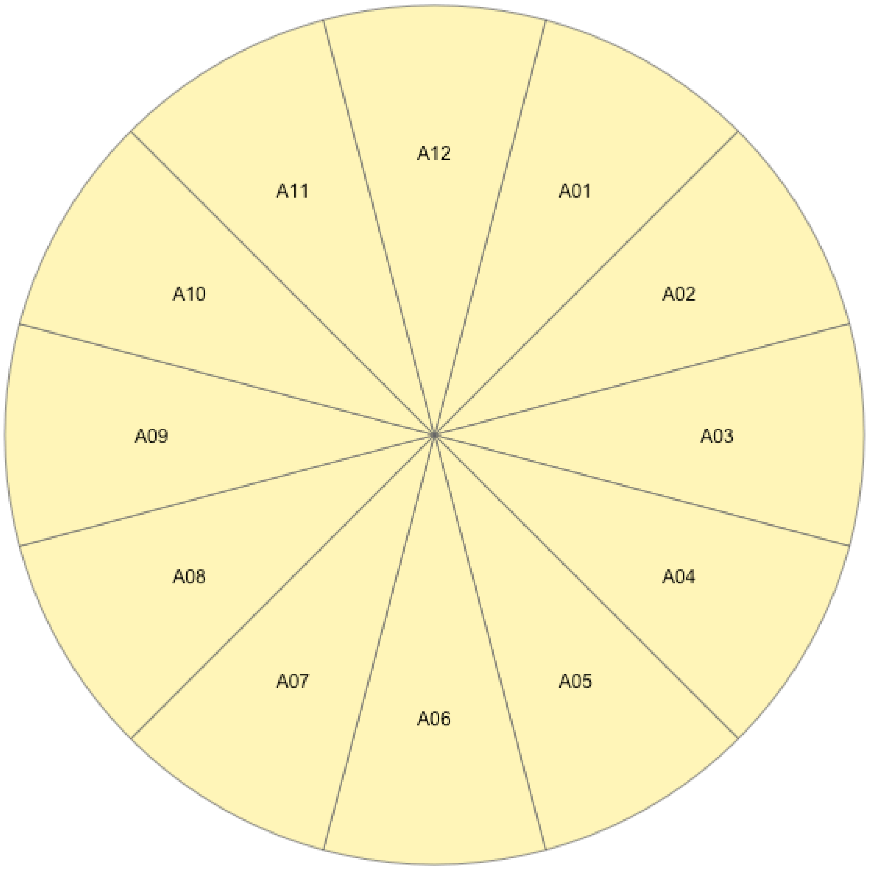
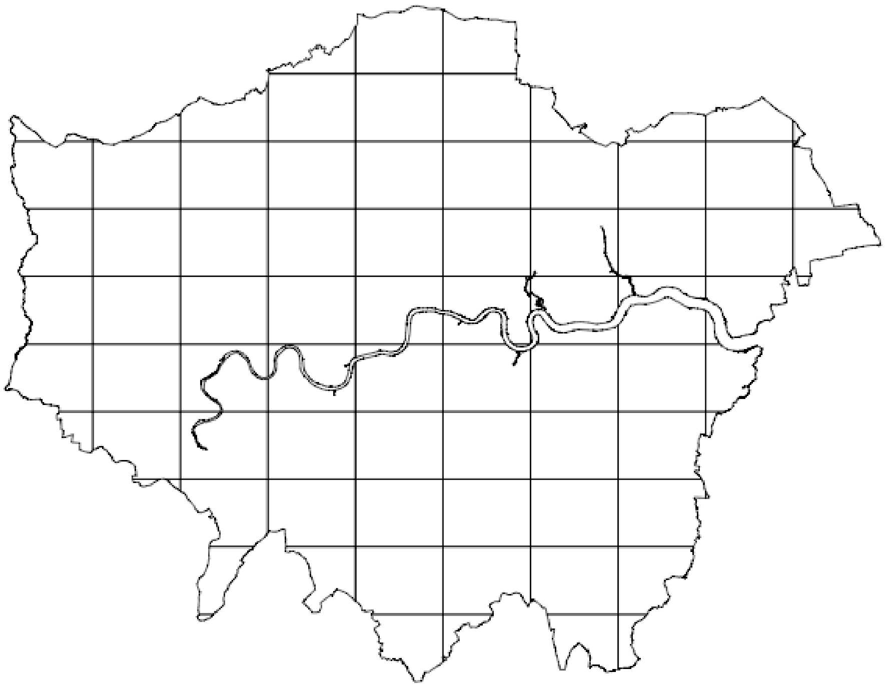


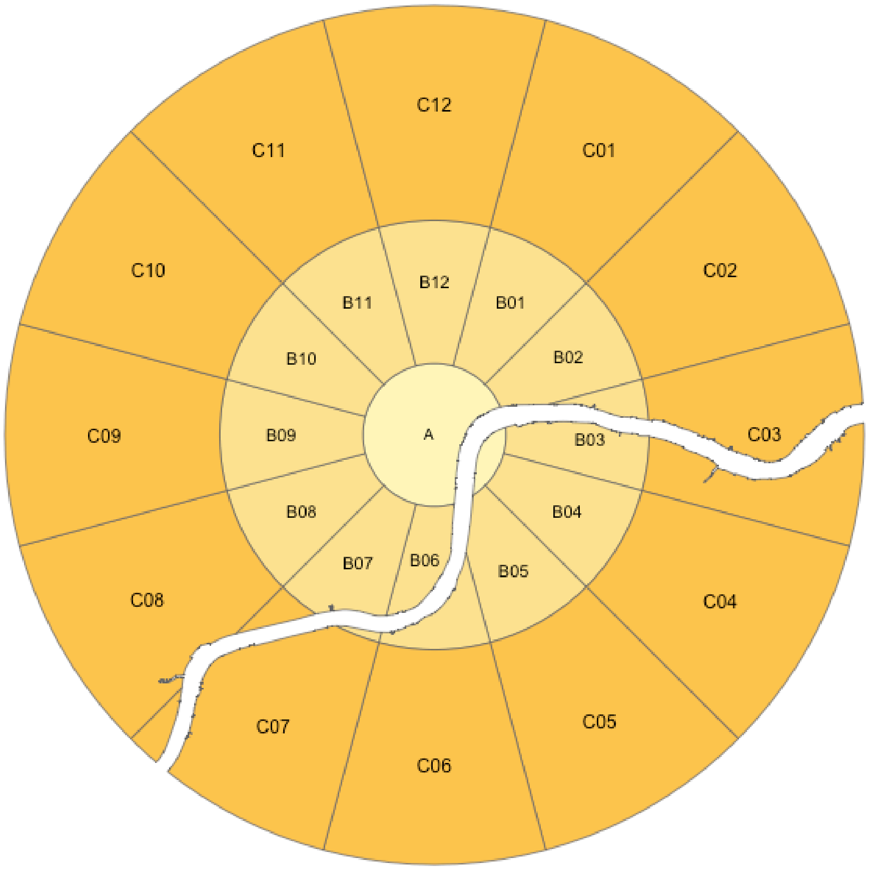
figure4_rectangular_grid.png

Comment: Reproduction of Figure 4: Rectangular grid comparison



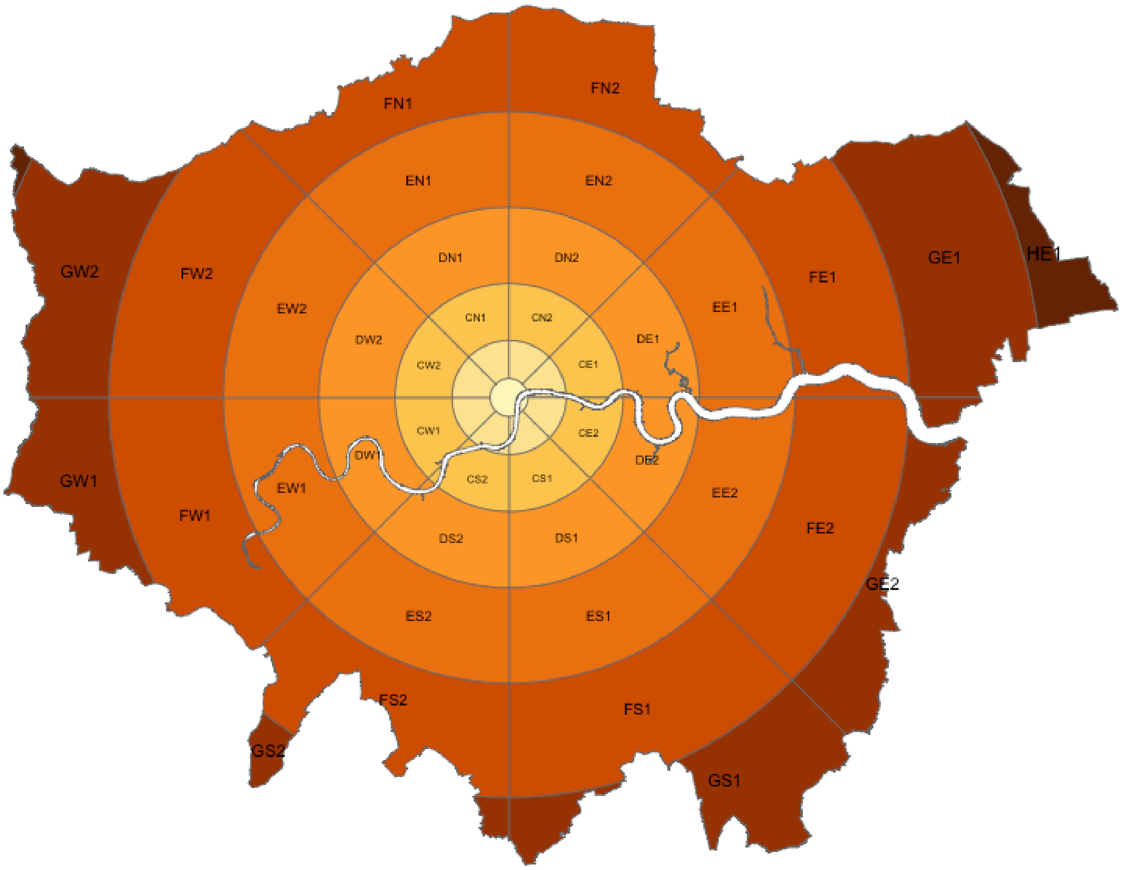
test_3_rings.png

Comment: Parameter test: 3 rings instead of default configuration



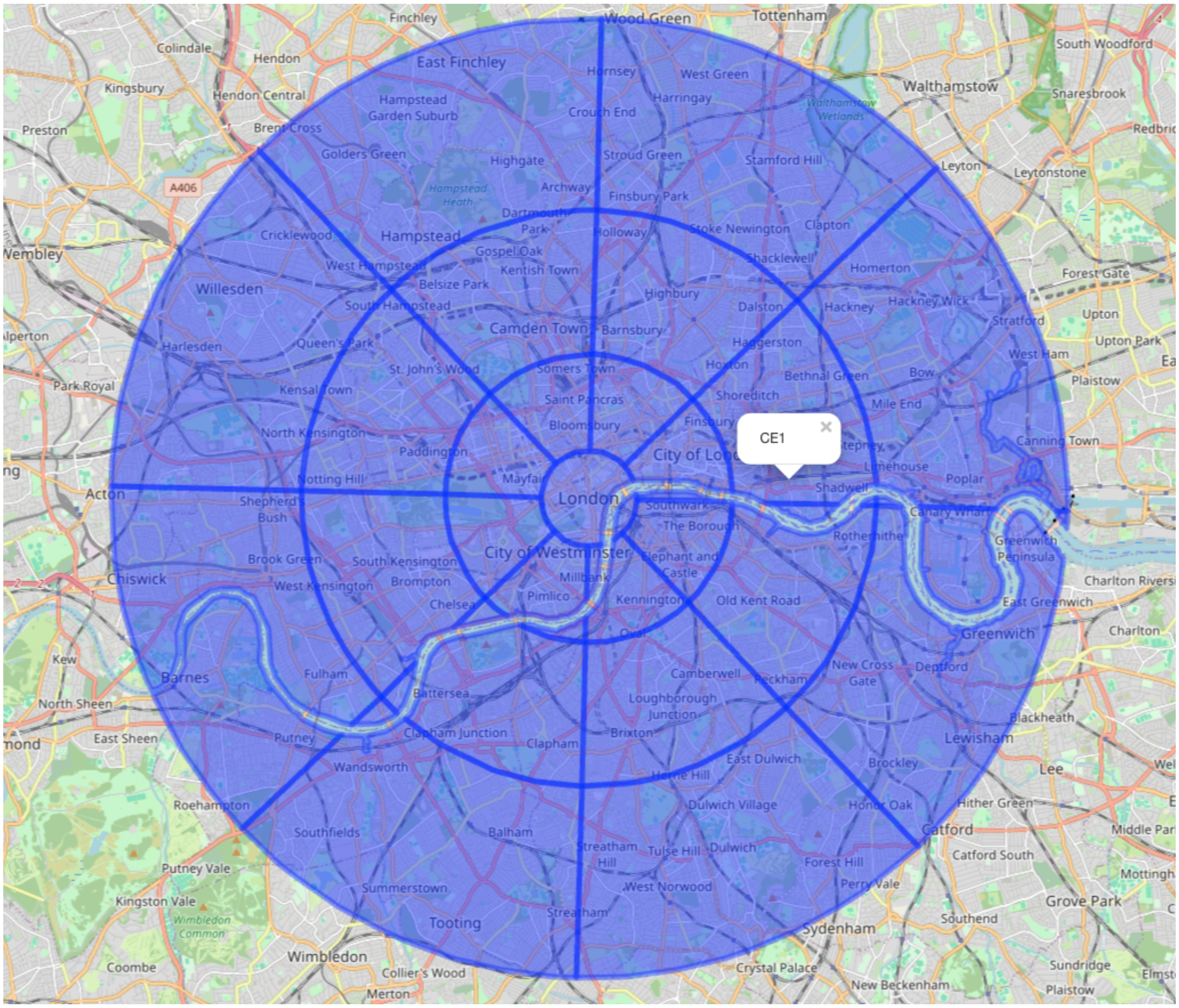
test_8_segments.png

Comment: Parameter test: 8 segments instead of default 12



test_interactive_leaflet.png

Comment: Screenshot of interactive leaflet map showing zoning system with geographic context



Acknowledgements

I would like to thank the authors Robin Lovelace, Martijn Tennekes, and Dustin Carlino for creating a well-documented and reproducible research package. The zonebuilder package represents excellent reproducible research practices with clear documentation and accessible code.

Citing this document

Subhan Ali (2025). CODECHECK Certificate 2025-027. Zenodo. https://doi.org/10.5281/zenodo.17059066

About CODECHECK

This certificate confirms that the codechecker could independently reproduce the results of a computational analysis given the data and code from a third party. A CODECHECK does not check whether the original computation analysis is correct. However, as all materials required for the reproduction are freely available by following the links in this document, the reader can then study for themselves the code and data.

About this document

This document was created using R Markdown using the `codecheck` R package. `make codecheck.pdf` will regenerate the report file.

```r
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS 15.6.1
## 
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib 
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
## 
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
## 
## time zone: Europe/Berlin
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets 
## [6] methods   base     
## 
## other attached packages:
##  [1] readr_2.1.5      tibble_3.2.1     xtable_1.8-4    
##  [4] yaml_2.3.10      rprojroot_2.0.4  knitr_1.50      
##  [7] codecheck_0.25.0 parsedate_1.3.2  R.cache_0.17.0  
## [10] gh_1.5.0        
## 
## loaded via a namespace (and not attached):
```

```
##  [1] xfun_0.52          rdflib_0.2.9       tzdb_0.5.0
##  [4] vctrs_0.6.5        tools_4.3.2        generics_0.1.3
##  [7] parallel_4.3.2     curl_6.4.0         fansi_1.0.6
## [10] pkgconfig_2.0.3    pdftools_3.5.0     R.oo_1.27.1
## [13] redland_1.0.17-18  readxl_1.4.3       lifecycle_1.0.4
## [16] compiler_4.3.2     atom4R_0.3-3       stringr_1.5.1
## [19] git2r_0.36.2       keyring_1.4.1      htmltools_0.5.8.1
## [22] crayon_1.5.2       pillar_1.9.0       whisker_0.4.1
## [25] tidyr_1.3.0        R.utils_2.13.0     cachem_1.1.0
## [28] zen4R_0.10.2       magick_2.8.7       tidyselect_1.2.0
## [31] zip_2.3.3          digest_0.6.33      stringi_1.8.3
## [34] dplyr_1.1.4        purrr_1.0.2        rsvg_2.6.2
## [37] fastmap_1.2.0      cli_3.6.2          magrittr_2.0.3
## [40] XML_3.99-0.18      crul_1.5.0         utf8_1.2.4
## [43] withr_2.5.2        osfr_0.2.9         bit64_4.6.0-1
## [46] roxygen2_7.3.2     rmarkdown_2.29     httr_1.4.7
## [49] bit_4.6.0          qpdf_1.4.1         cellranger_1.1.0
## [52] askpass_1.2.1      R.methodsS3_1.8.2  hms_1.1.3
## [55] memoise_2.0.1      evaluate_0.23      rlang_1.1.2
## [58] Rcpp_1.1.0         glue_1.6.2         httpcode_0.3.0
## [61] xml2_1.3.8         fauxpas_0.5.2      rorcid_0.7.0
## [64] vroom_1.6.5        jsonlite_1.8.8     plyr_1.8.9
## [67] R6_2.5.1           fs_1.6.3
```