HS19 Data Visualization Concepts

Exercise Session
Single View Data Visualization

Haiyan Yang

haiyan@ifi.uzh.ch
Visualization and MultiMedia Lab
Department of Informatics
University of Zürich

University of Zurich[UZH]

VISUALIZATION AND MULTIMEDIA LAB

Exercise Sessions

Exercise Sessions
General Overview

- Fundamental concepts of data visualization
  - Digitization, Color and perception, Multivariate data analysis and visualization, Spatial and geospatial data visualization, Trees, Graphs and networks
- Programming exercises using Python
  - Python familiarity for this course will be needed
- Takes place on specific Thursdays throughout the semester
  - 5 exercise sessions are scheduled, with exercise point distribution: 2 - 3 - 2 - 5 - 3
  - A minimum points of 5 must be achieved out of exercise 1, 2 and 4
- Contact (for exercises): Kate Gadola <kate.gadola@uzh.ch>
  - Discussions and questions will be handled in OLAT forum (technical questions) and VisGuides (theoretical questions)
  - For meeting in person, arrange first an appointment
  - Please use the platforms as much as possible such that all the students can benefit from them

Exercise Sessions
Session Structure

- Introduction to new theoretical and technical aspects related to new assignment
  - Including examples, code structure, demos, sample results if necessary, etc.
- Announcement of new assignment (one in every session)
  - Discussion of requirements, solution hints, etc.
  - Clarification of point requirement.
- Discussion
  - Including comments on submissions, answering questions regarding the grading, etc.
  - Questions regarding the newly assigned exercise can be handled shortly after the lecture (15mins) or on the OLAT later on.
- Takes place in the second half of the lecture on Thursdays

Exercise Sessions
Assignment Guidelines

- Assignments related only to Data Visualization course.
- For every assignment there will be a .zip file in OLAT, including:
  - Exercise's announcement (***what to do*** — *assignment requirements in details*)
  - Presentation of theoretical and technical aspects (***how to do*** — *technical guidance*)
  - Data files and code skeleton (with needed libraries and hints) will be provided when necessary
  - **Attention!** Uploaded .zip file will **NOT** include additional material presented in lab session , such as demos, code snippets, solutions from previous sessions, etc.
- Grading will be ***incomplete*** or ***complete***
  - Only a **complete** will result in the corresponding awarded points
- Exercises MUST be submitted before each deadline, otherwise it will be a FAIL.

Exercise Sessions
Assignment Submission

- Project files must be zipped and the .zip archive has to be named: dvc_exc2_MATRIKELNUMBER.zip
  - e.g. dvc_exc2_01234567.zip
- Follow exercise instructions and provide an answer sheet when necessary (e.g. a “readme.txt” file)
- No trash / debug output
- Use OLAT to submit files
  - Deadlines are typically Wednesdays at 23:59
- Use OLAT forum to ask questions about the technical questions from exercises

Instructions

Instructions

Dataset

GLOBAL UNIQUE IDENTIFIER	LAST EDITED DATE	TAXONOMIC ORDER	CATEGORY	COMMON NAME	SCIENTIFIC NAME	SUBSPECIES COMMON NAME	SUBSPECIES SCIENTIFIC NAME	OBSERVATION COUNT	E
URN:CornellLabOfOrnithology:EBIRD:OBS570779581	2018-01-24 14:41:34.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS563786537	2018-01-03 22:54:33.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS563333559	2018-01-03 22:54:33.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS563148351	2018-01-03 21:46:14.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS566009229	2018-01-10 16:17:03.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS563757539	2018-01-03 21:46:14.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS573616517	2018-05-01 13:28:03.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS570204089	2018-01-22 13:06:57.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS565026901	2018-01-07 13:01:03.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS565027324	2018-01-07 13:11:33.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS563757396	2018-01-06 16:34:31.0	19073	species	American Crow	Corvus brachyrhynchos			3	
URN:CornellLabOfOrnithology:EBIRD:OBS563135906	2018-01-06 16:34:34.0	19073	species	American Crow	Corvus brachyrhynchos			3	
URN:CornellLabOfOrnithology:EBIRD:OBS570775095	2018-01-24 14:20:02.0	19073	species	American Crow	Corvus brachyrhynchos			4	
URN:CornellLabOfOrnithology:EBIRD:OBS570066312	2018-01-23 06:08:07.0	19073	species	American Crow	Corvus brachyrhynchos			4	
URN:CornellLabOfOrnithology:EBIRD:OBS566304578	2018-01-11 11:32:41.0	19073	species	American Crow	Corvus brachyrhynchos			2	
URN:CornellLabOfOrnithology:EBIRD:OBS571034544	2018-01-25 16:59:50.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS570975325	2018-01-25 16:59:50.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS563307268	2018-01-03 16:14:41.0	19073	species	American Crow	Corvus brachyrhynchos			4	
URN:CornellLabOfOrnithology:EBIRD:OBS566313986	2018-01-11 13:40:10.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS571610540	2018-01-27 15:19:02.0	19073	species	American Crow	Corvus brachyrhynchos			1	
URN:CornellLabOfOrnithology:EBIRD:OBS572716949	2018-01-30 11:29:50.0	31039	species	American Goldfinch	Spinus tristis			1	
URN:CornellLabOfOrnithology:EBIRD:OBS563065389	2018-01-02 09:50:07.0	31039	species	American Goldfinch	Spinus tristis			1	

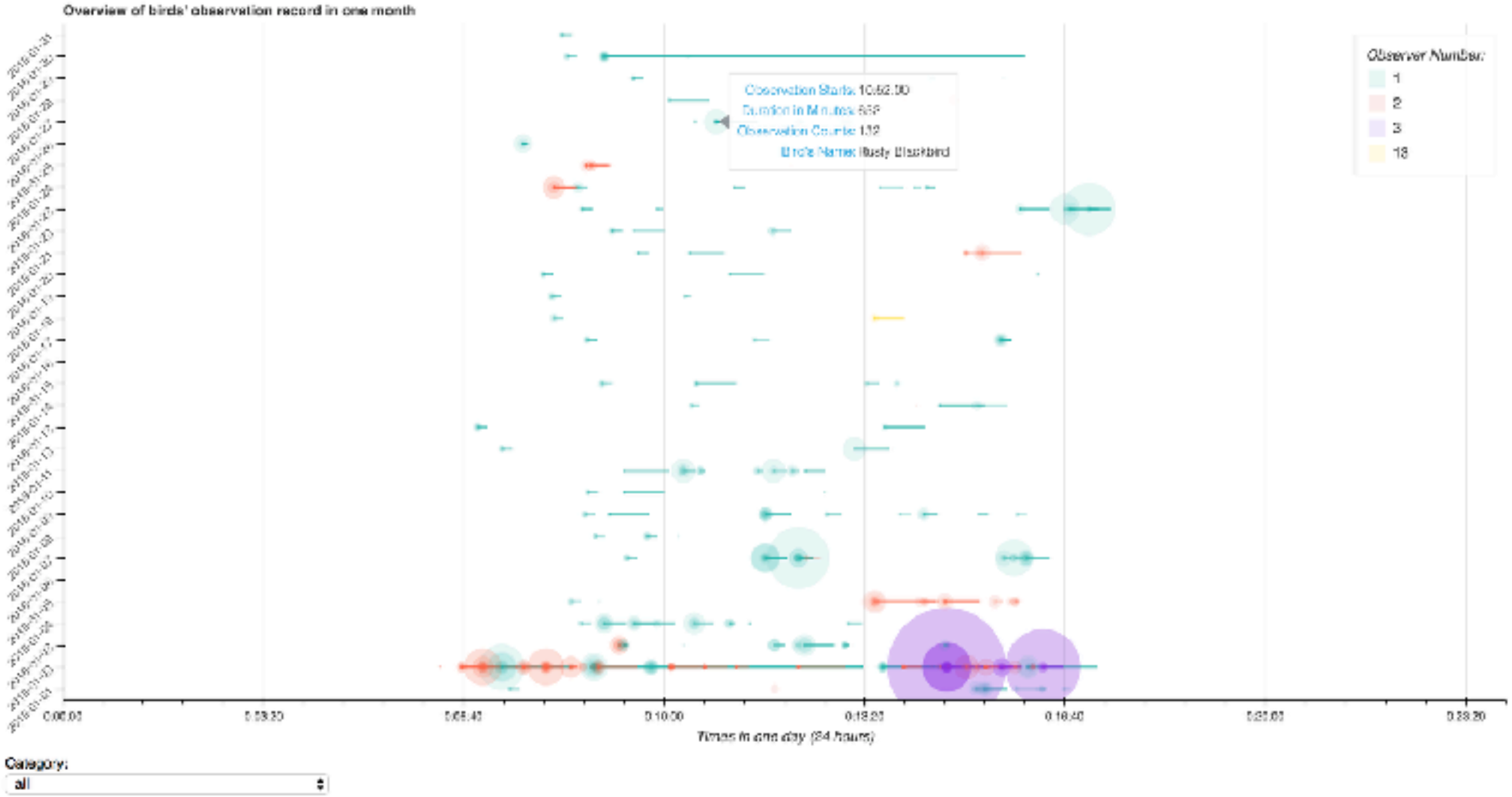
...

▪
▪
▪

Instructions

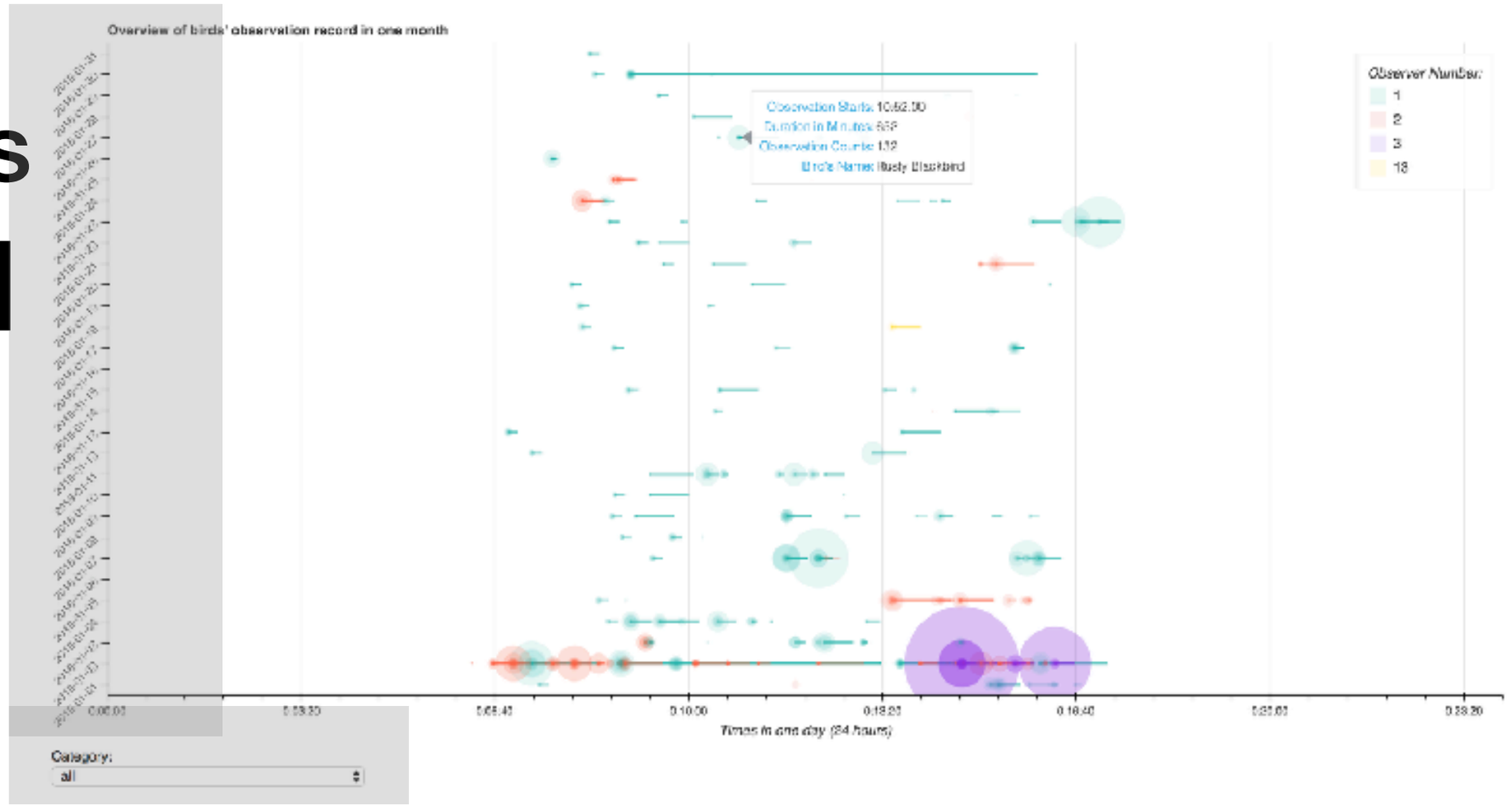
Data Visualization

- Final result should be something **similar** as shown below.
  - Detailed requirements and hints are described in the “exercise assignment” document as well as the code skeleton.



Instructions

Tasks - 1

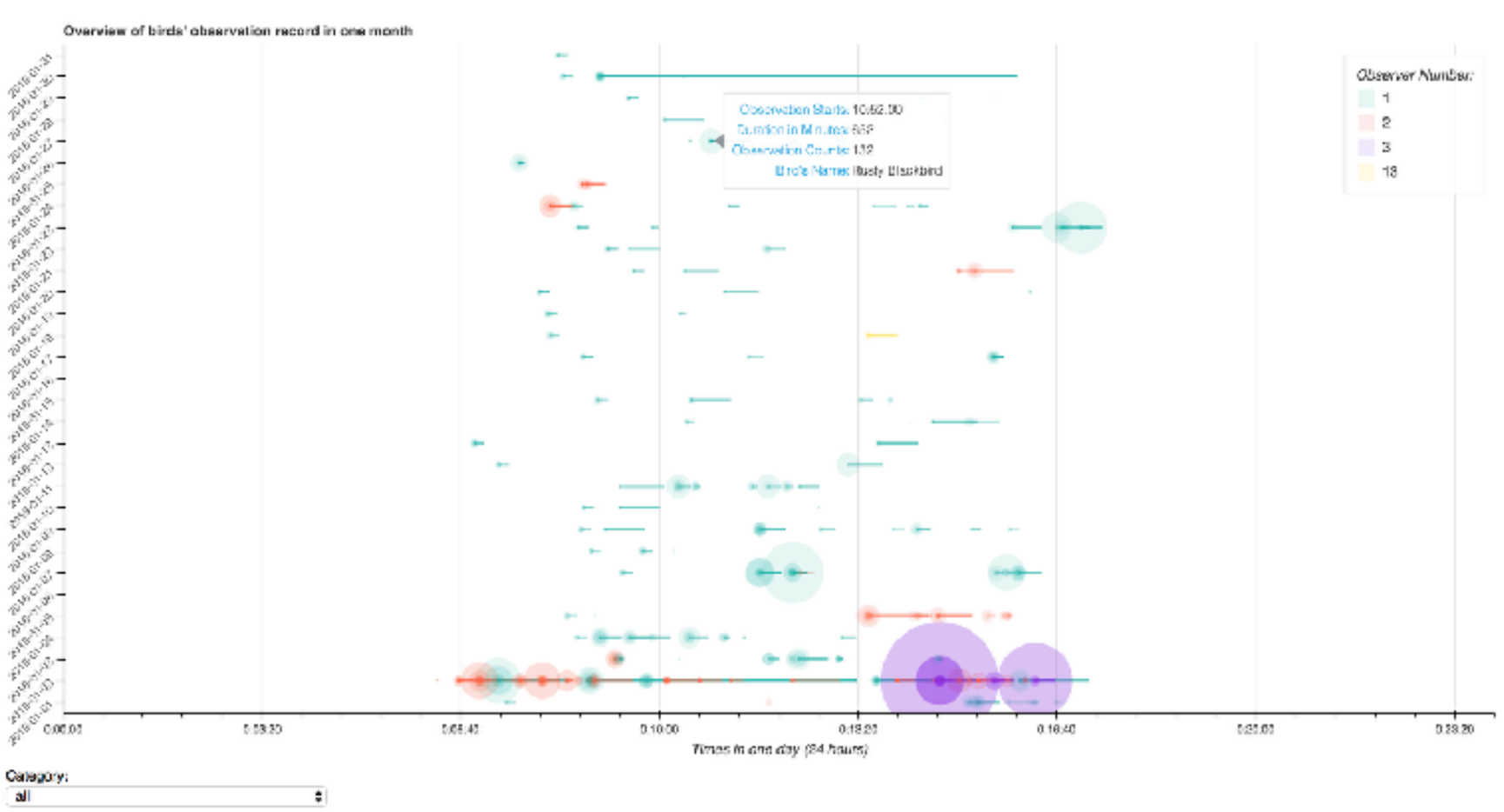


- Generate proper y axis labels, which is consistent through all the conditions
  - The result should be a list of array: (['2018-01-01', '2018-01-02', '2018-01-03',....]).
  - Can be extracted from the original data under the attribute of “OBSERVATION DATE”.
  - After extracting, there are still two days missing, try to add them into the list.
  - Finally sort the list.

- Add an interactive selector for selecting birds based on the category (or name, locality, protocol, etc.)
  - Define the “options” for the selector by extracting unique items inside the attribute category (or the one you choose)
  - Don’t forget to add “all” option into the options list, read the reference of how to add it effectively

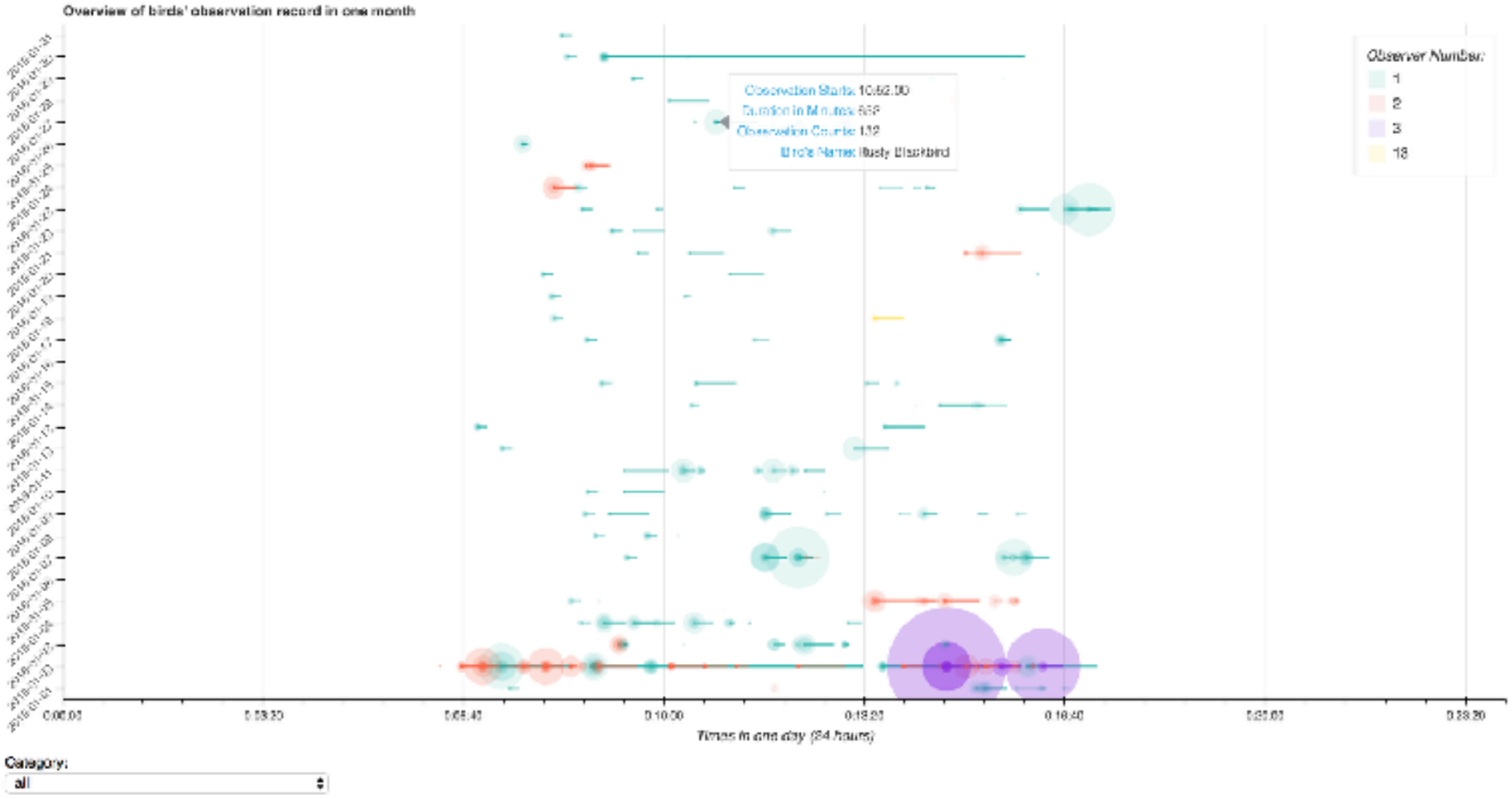
Instructions

Tasks - 2



- Define the datasource construction function by following the hints in the code skeleton
  - The input and output are described in the skeleton.
  - The provided code can also be modified, for example, choose different colorset from the bokeh color library: https://bokeh.pydata.org/en/latest/docs/reference/colors.html
  - The items to put into the data source can be modified accordingly too, based on what you want to visualize. Read the provided explanation about each item and then decide which ones to add and/or remove.

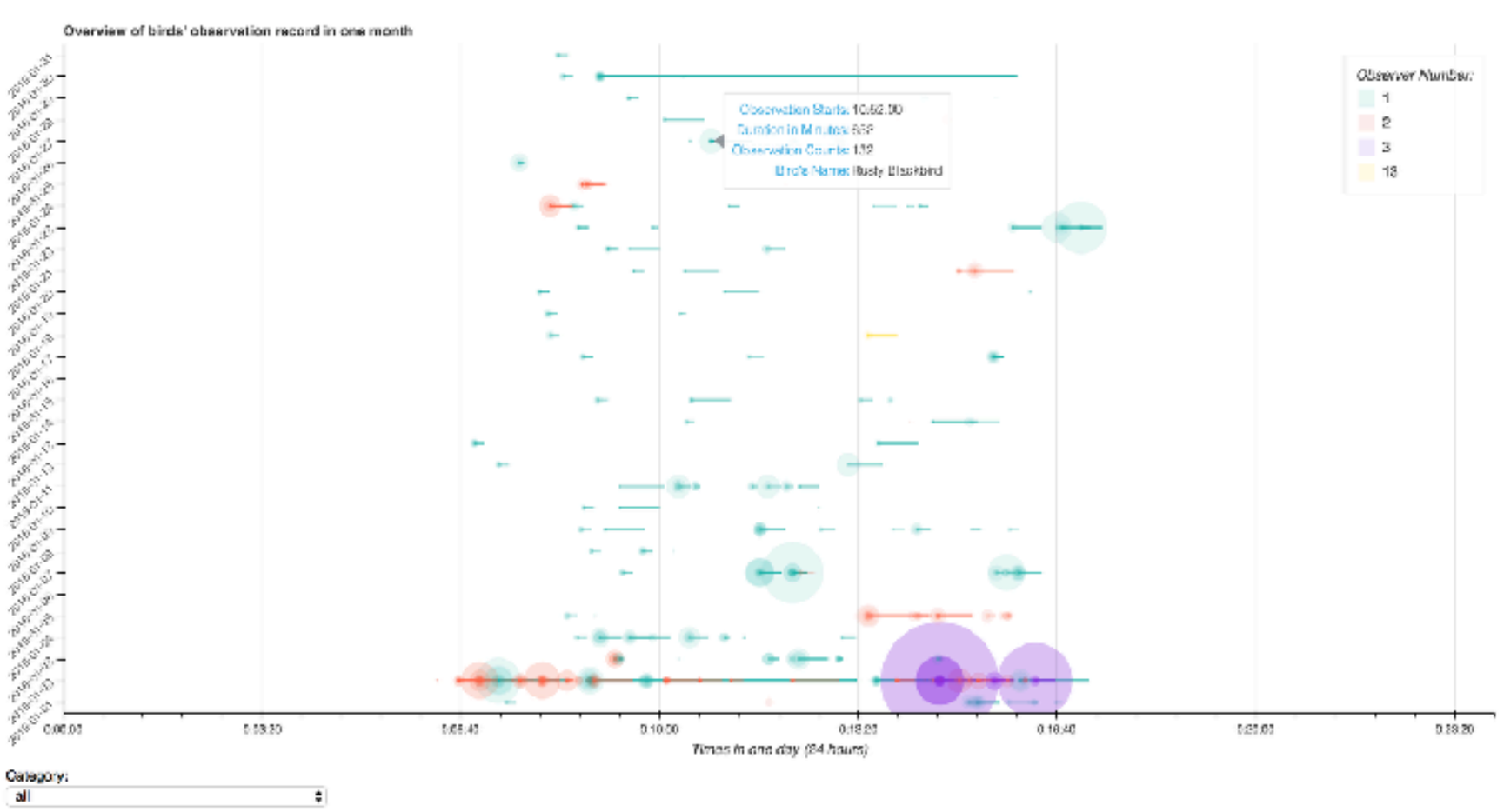
Instructions
Tasks - 3



- Define the update source function
  - The update function **always** takes three arguments: **attr**, **old**, **new** and updates the plot based on the selection controls. Bokeh requires and checks for these arguments in callbacks (i.e. the update function) to avoid potential user errors.
  - **attr** is set to the attribute you pass as the first argument of *on_change*,
  - **old** is the old attribute value,
  - **new** is the new attribute value.
  - But the actual input which will be used inside of the function is "**value**", the usage is just calling **select.value** to pass the current selected value from the selector.

Instructions

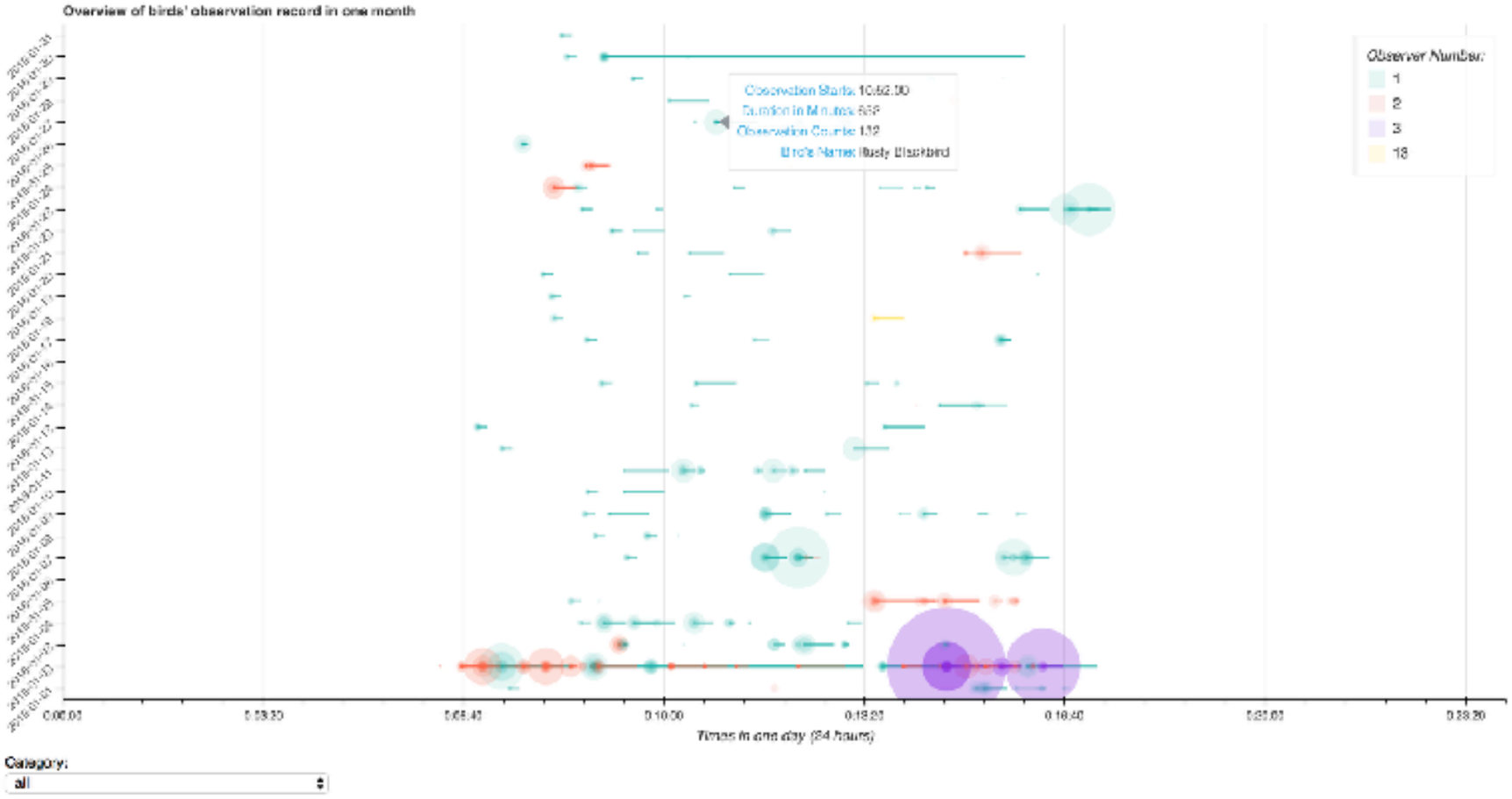
Tasks - 4



- Define the plot function which will be called whenever the selector is triggered, as well as when running the server initially
  - When run the bokeh server the first time, this function should already be called to be able to draw the initial plots.
  - The hbar plot should start each bar at the observation started time and end at the observation ended time.
  - The circle plot should position the circle also at the observation started time, with the circle size controlled by the number of observations.
  - use the color option to specify different colors for each bar and circle according to the number of observers, which is defined in the datasource construction function.
  - Read the hint for formatting the x axis into Hour-Minutes form from [0, 1440].
  - Add the hover tool properly, and make it only effective for the circle plot, i.e. p2.

Instructions

Tasks - 5



- Draw the figure and add bokeh server for interactive visualization
  - Draw the plots by calling the pre-defined functions
  - Add the plots and the selector into one layout/dashboard and run the application via bokeh server: https://bokeh.pydata.org/en/latest/docs/user_guide/server.html

Instructions
Grading Scheme

- Tasks
  - **Data visualization**: a combined horizontal bar chart and circle plot which visually encode multiple attributes from the dataset.
  - **Interactive selection**: add a selector based on meaningful attribute like category, name, locality, protocol, etc.

- Grading
  - Points: {0, 2, 3}
  - **3**: only if you complete **all** the tasks and submit the requirement documents **before deadline.**
  - **2**: if you can finish the visualization but unable to add the selector successfully.
  - **0**: if trash/no output or miss of the deadline.

Instructions
Code Skeleton

- Code provided
  - ***ALL*** necessary libraries are provided
  - Reference links and hints for each task, if any
  - Read this example as a starting point: https://bokeh.pydata.org/en/latest/docs/gallery/bar_intervals.html
- Code missing
  - See the code skeleton for details

Instructions
Do It Yourself

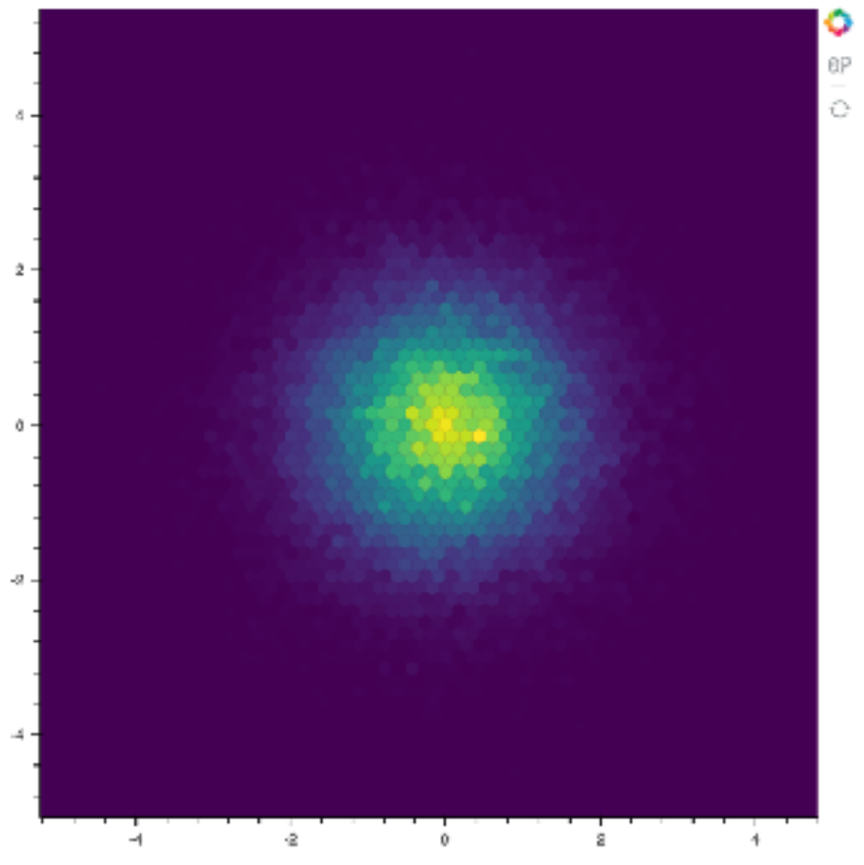
- Learn Python-Bokeh visualization techniques from online tutorials and documentation
- Read carefully the instructions in the skeleton and finish the missing parts
- **Write proper comments for your code**
- **Use forum to ask and/or answer questions**
- Make sure to produce desired results with Bug-Free code
- Submit it before deadline

Python-Bokeh

Visualization

Python-Bokeh Visualization
Bokeh Plots

- More plots can be found in:
  - http://bokeh.pydata.org/en/latest/docs/user_guide/plotting.html
  - https://bokeh.pydata.org/en/latest/docs/user_guide/categorical.html
  - http://bokeh.pydata.org/en/latest/docs/user_guide/tools.html



Python-Bokeh Visualization
More Tutorials

- Interactive Data Visualization in Python With Bokeh (https://realpython.com/python-data-visualization-bokeh/)
  - An online advanced tutorial
- Bokeh server (https://bokeh.pydata.org/en/latest/docs/user_guide/server.html)
  - The Bokeh server usage with some examples
- Bokeh documentation (https://bokeh.pydata.org/en/latest/docs/user_guide.html#userguide)
  - The official Bokeh documentation
  - With nice examples

Live Demo

Thank you