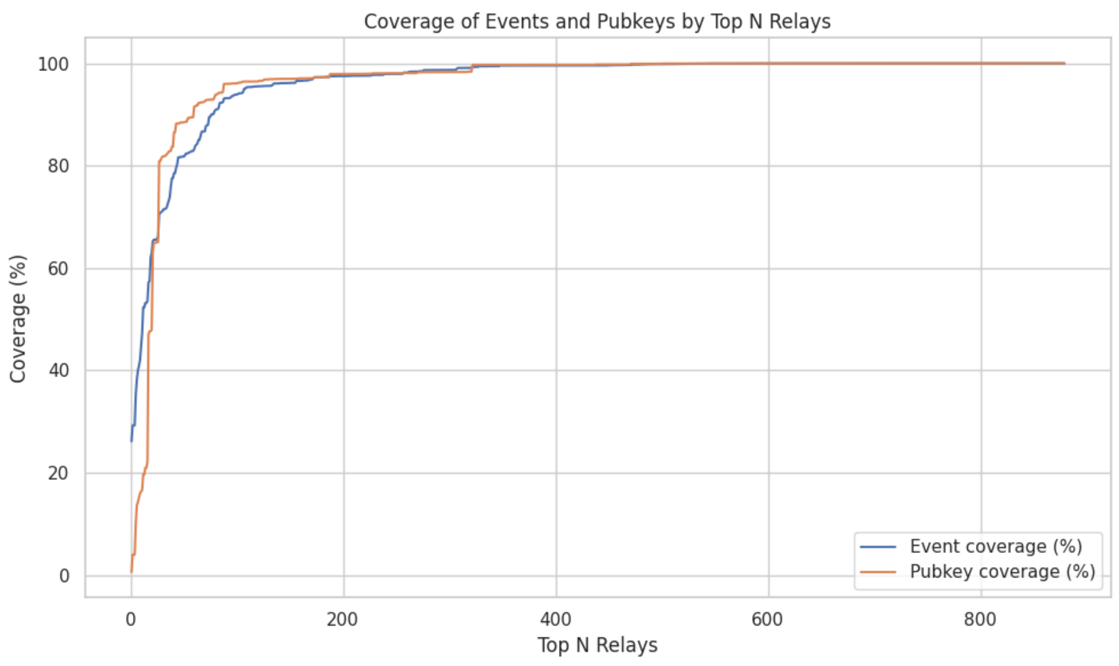
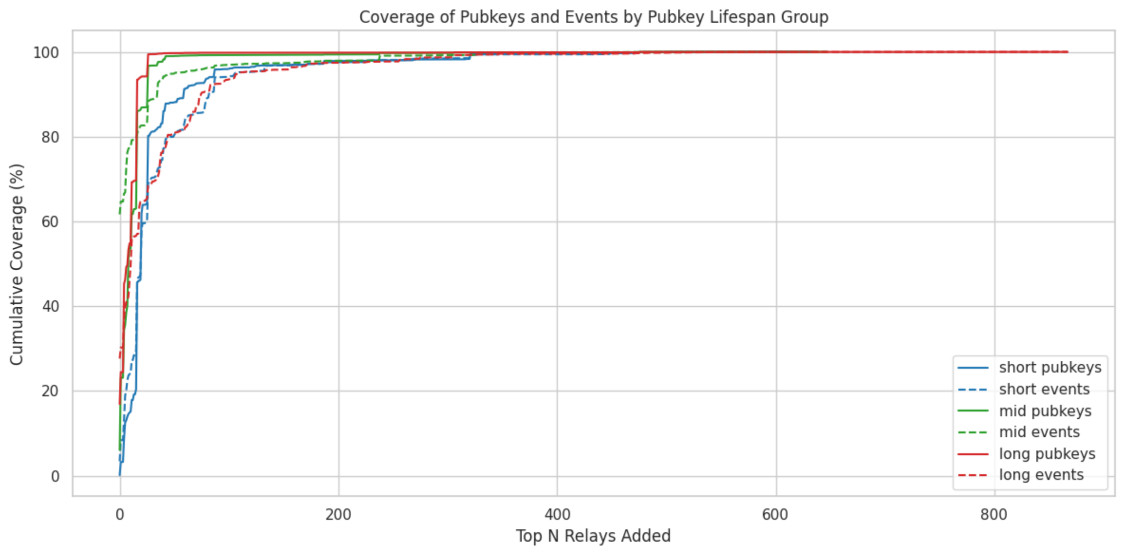
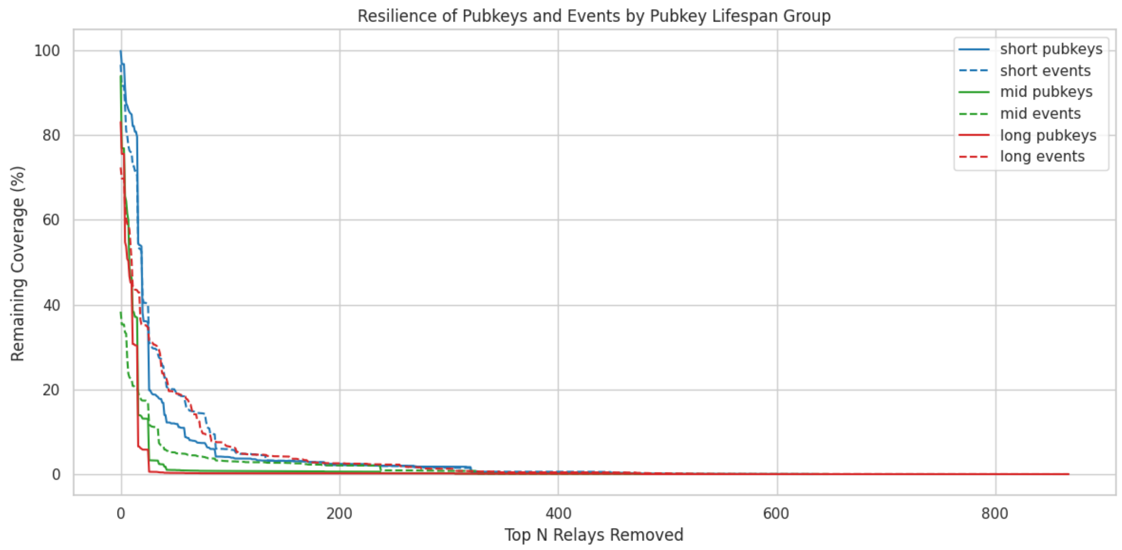
events_relays_overview

July 14, 2025







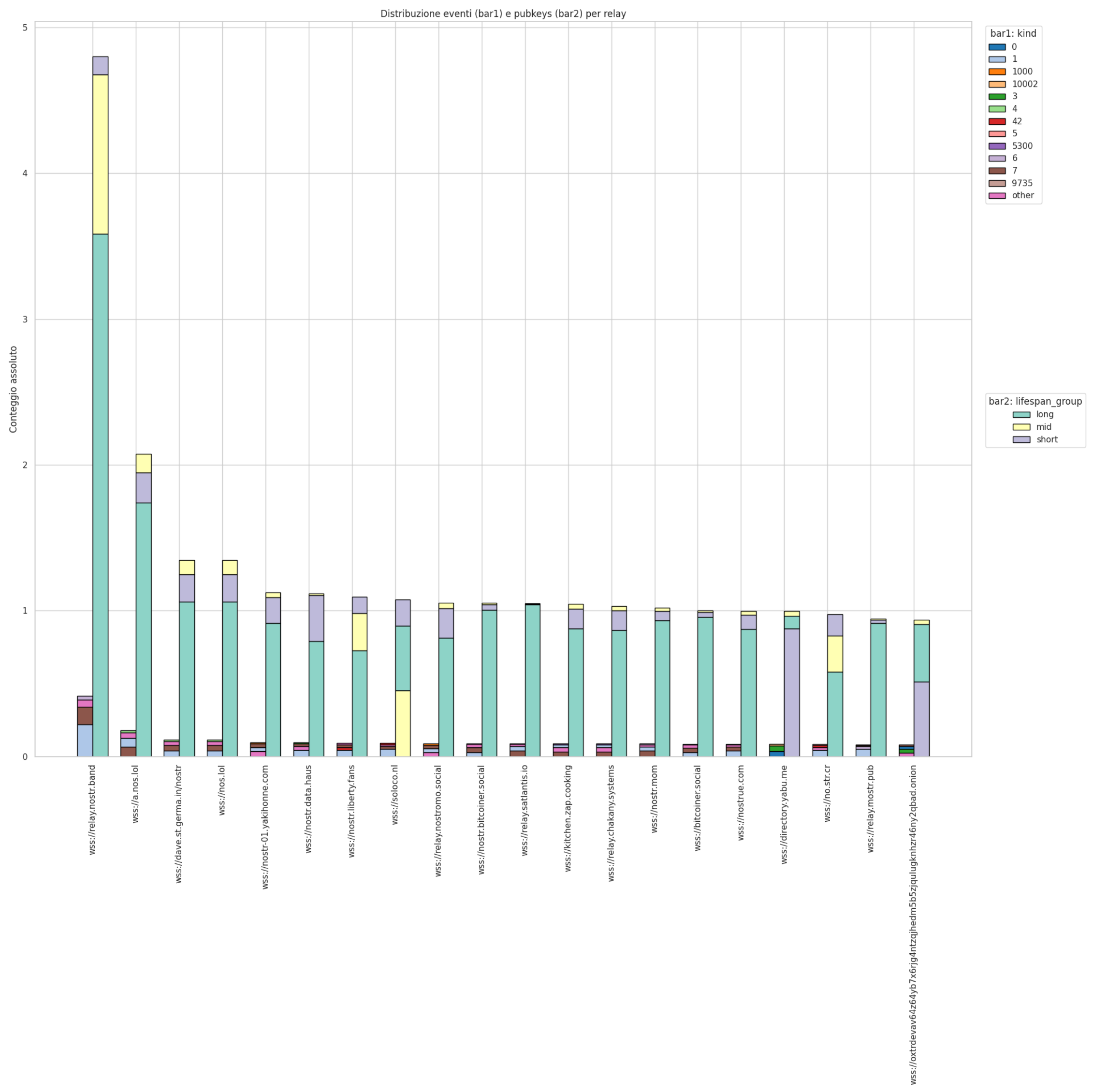
```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
↪_core/data.py:313, in convert_dataframe_to_pandas(data)
    306 try:
    307     # This is going to convert all columns in the input dataframe, even
↪though
    308     # we may only need one or two of them. It would be more efficient to
↪select
   (…)
    311     # interface where variables passed in Plot() may only be referenced
↪later
    312     # in Plot.add(). But noting here in case this seems to be a
↪bottleneck.
--> 313     return pd.api.interchange.from_dataframe(data)
    314 except Exception as err:

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/pandas/
↪core/interchange/from_dataframe.py:54, in from_dataframe(df, allow_copy)
     52     raise ValueError("`df` does not support __dataframe__")
---> 54 return _from_dataframe(df.__dataframe__(allow_copy=allow_copy))

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/pandas/
↪core/interchange/from_dataframe.py:75, in _from_dataframe(df, allow_copy)
     74 for chunk in df.get_chunks():
---> 75     pandas_df = protocol_df_chunk_to_pandas(chunk)
     76     pandas_dfs.append(pandas_df)
```

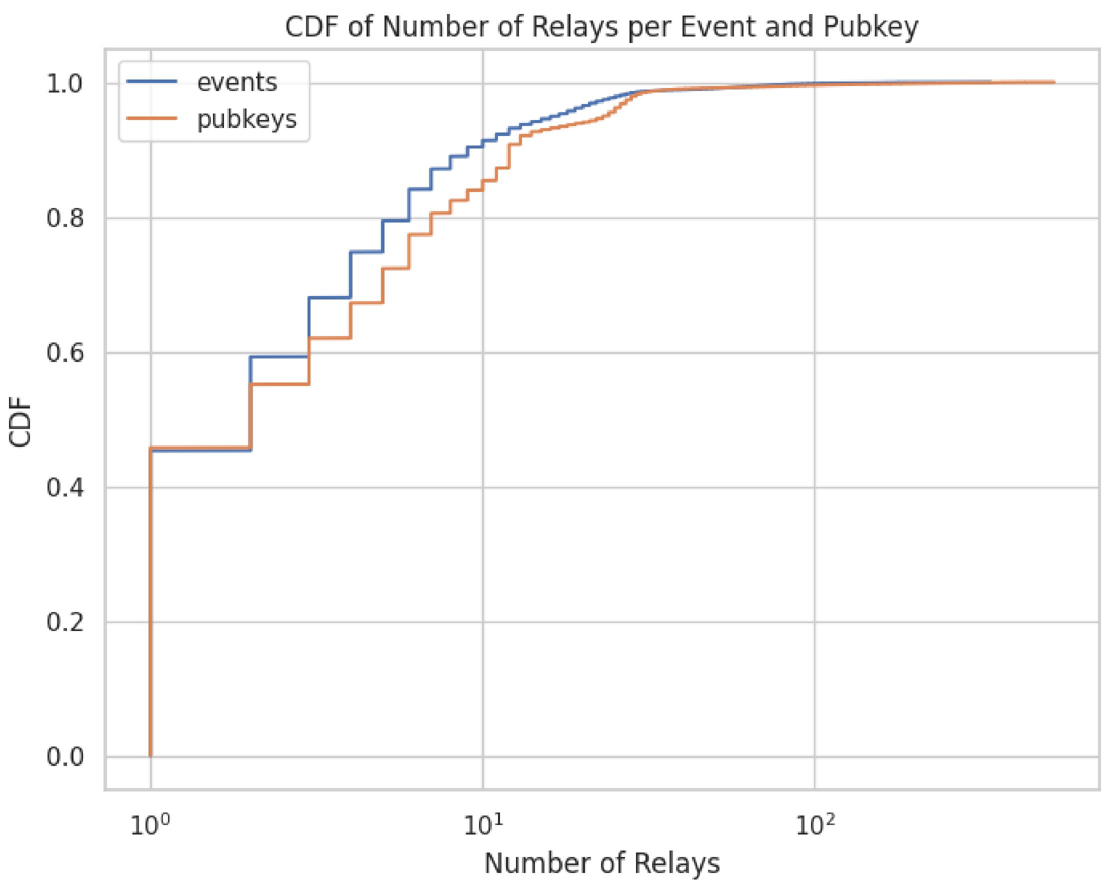
```
File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/pandas/
↪core/interchange/from_dataframe.py:127, in protocol_df_chunk_to_pandas(df)
    126 elif dtype == DtypeKind.STRING:
--> 127     columns[name], buf = string_column_to_ndarray(col)
    128 elif dtype == DtypeKind.DATETIME:

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/pandas/
↪core/interchange/from_dataframe.py:251, in string_column_to_ndarray(col)
    250 assert protocol_data_dtype[1] == 8
--> 251 assert protocol_data_dtype[2] in (
    252     ArrowCTypes.STRING,
    253     ArrowCTypes.LARGE_STRING,
    254 )  # format_str == utf-8
    255 # Convert the buffers to NumPy arrays. In order to go from STRING to
    256 # an equivalent ndarray, we claim that the buffer is uint8 (i.e., a byt␣
↪array)

AssertionError:

The above exception was the direct cause of the following exception:

RuntimeError                                  Traceback (most recent call last)
Cell In[19], line 3
      1 # --- Boxplot: Distribution of Events for each Lifespan Group (Ordered)
      2 plt.figure(figsize=(12, 6))
----> 3␣
↪sns.boxplot(data=pubkey_stats, x='lifespan_group', y='event_count', palette="Set2", order=
      4 plt.title('Boxplot of Event Count per Lifespan Group')
      5 plt.yscale('log')

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
↪categorical.py:1597, in boxplot(data, x, y, hue, order, hue_order, orient,␣
↪color, palette, saturation, fill, dodge, width, gap, whis, linecolor,␣
↪linewidth, fliersize, hue_norm, native_scale, log_scale, formatter, legend,␣
↪ax, **kwargs)
   1589 def boxplot(
   1590     data=None, *, x=None, y=None, hue=None, order=None, hue_order=None,
   1591     orient=None, color=None, palette=None, saturation=.75, fill=True,
   (…)
   1594     legend="auto", ax=None, **kwargs
   1595 ):
-> 1597     p = _CategoricalPlotter(
   1598         data=data,
   1599         variables=dict(x=x, y=y, hue=hue),
   1600         order=order,
   1601         orient=orient,
   1602         color=color,
   1603         legend=legend,
   1604     )
   1606     if ax is None:
   1607         ax = plt.gca()

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪categorical.py:67, in _CategoricalPlotter.__init__(self, data, variables,␣
 ↪order, orient, require_numeric, color, legend)
     56 def __init__(
     57     self,
     58     data=None,
   (…)
     64     legend="auto",
     65 ):
---> 67     super().__init__(data=data, variables=variables)
     69     # This method takes care of some bookkeeping that is necessary␣
 ↪because the
     70     # original categorical plots (prior to the 2021 refactor) had some␣
 ↪rules that
     71     # don't fit exactly into VectorPlotter logic. It may be wise to hav␣
 ↪a second
   (…)
     76     # default VectorPlotter rules. If we do decide to make orient part␣
 ↪of the
     77     # _base variable assignment, we'll want to figure out how to expres␣
 ↪that.
     78     if self.input_format == "wide" and orient in ["h", "y"]:

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪_base.py:634, in VectorPlotter.__init__(self, data, variables)
    629 # var_ordered is relevant only for categorical axis variables, and may
    630 # be better handled by an internal axis information object that tracks
    631 # such information and is set up by the scale_* methods. The analogous
    632 # information for numeric axes would be information about log scales.
    633 self._var_ordered = {"x": False, "y": False}  # alt., used DefaultDict
--> 634 self.assign_variables(data, variables)
    636 # TODO Lots of tests assume that these are called to initialize the
    637 # mappings to default values on class initialization. I'd prefer to
    638 # move away from that and only have a mapping when explicitly called.
    639 for var in ["hue", "size", "style"]:

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪_base.py:679, in VectorPlotter.assign_variables(self, data, variables)
    674 else:
    675     # When dealing with long-form input, use the newer PlotData
    676     # object (internal but introduced for the objects interface)
    677     # to centralize / standardize data consumption logic.
    678     self.input_format = "long"
```

```
--> 679     plot_data = PlotData(data, variables)
    680     frame = plot_data.frame
    681     names = plot_data.names

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪_core/data.py:57, in PlotData.__init__(self, data, variables)
     51 def __init__(
     52     self,
     53     data: DataSource,
     54     variables: dict[str, VariableSpec],
     55 ):
---> 57     data = handle_data_source(data)
     58     frame, names, ids = self._assign_variables(data, variables)
     60     self.frame = frame

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪_core/data.py:275, in handle_data_source(data)
    271 """Convert the data source object to a common union representation."""
    272 if isinstance(data, pd.DataFrame) or hasattr(data, "__dataframe__"):
    273     # Check for pd.DataFrame inheritance could be removed once
    274     # minimal pandas version supports dataframe interchange (1.5.0).
--> 275     data = convert_dataframe_to_pandas(data)
    276 elif data is not None and not isinstance(data, Mapping):
    277     err = f"Data source must be a DataFrame or Mapping, not {type(data)
 ↪r}."

File /eth/vincenzo/bigbrotr-analysis/venv/lib/python3.8/site-packages/seaborn/
 ↪_core/data.py:319, in convert_dataframe_to_pandas(data)
    314 except Exception as err:
    315     msg = (
    316         "Encountered an exception when converting data source "
    317         "to a pandas DataFrame. See traceback above for details."
    318     )
--> 319     raise RuntimeError(msg) from err

RuntimeError: Encountered an exception when converting data source to a pandas␣
 ↪DataFrame. See traceback above for details.
```

```
<Figure size 1200x600 with 0 Axes>
```

```
Top 20 relays: 108527490 unique events (63.01%) and 9386055 unique pubkeys
(47.91%)
```


Distribuzione eventi (bar1) e pubkeys (bar2) per relay
Conteggio assoluto
0
1
2
3
4
5
bar1: kind
0
1
1000
10002
3
4
42
5
5300
6
7
9735
other
bar2: lifespan_group
long
mid
short
wss://relay.nostr.band
wss://a.nos.lol
wss://dave.st.germa.in/nostr
wss://nos.lol
wss://nostr-01.yakihonne.com
wss://nostr.data.haus
wss://nostr.liberty.fans
wss://soloco.nl
wss://relay.nostromo.social
wss://nostr.bitcoiner.social
wss://relay.satlantis.io
wss://kitchen.zap.cooking
wss://relay.chakany.systems
wss://nostr.mom
wss://bitcoiner.social
wss://nostrue.com
wss://directory.yabu.me
wss://no.str.cr
wss://relay.mostr.pub
wss://oxtrdevav64z64yb7x6rjg4ntzqjhedm5b5zjqulugknhzr46ny2qbad.onion


CDF of Number of Relays per Event and Pubkey
events
pubkeys
1.0
0.8
0.6
0.4
0.2
0.0
CDF
10^0
10^1
10^2
Number of Relays