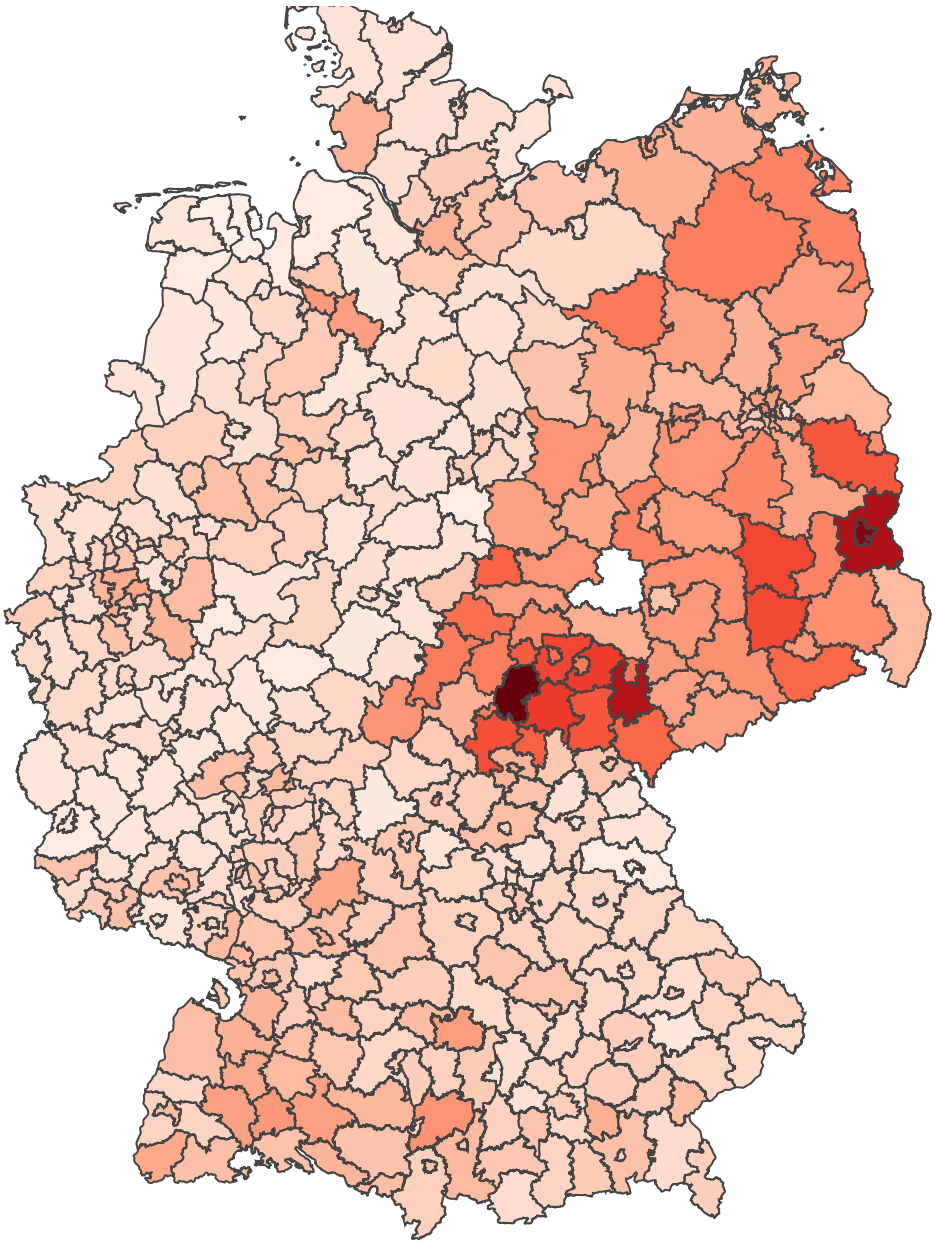
# Spatial Visualizations in quick 15 liners

## 1. Seven-day incidence per county as per Robert-Koch-Institute (Plotly Express interactive figure)

In [38]:

```
import plotly.express as px; import requests; import plotly.io as pio; pio.renderers

apiurl = 'https://opendata.arcgis.com/datasets/917fc37a709542548cc3be077a786c17_0.ge
resp = requests.get(apiurl)

lk_geo = []
for lk in resp.json()["features"]:
    AdmUnitId = lk["properties"]["AdmUnitId"]
    geometry = lk["geometry"]
    defects = [15088, 15002, 8216] # filtering incompatible county polygons to avoid
                                   # todo replace these coordinates with simpler mod
    if AdmUnitId not in defects:
        lk_geo.append({
            "type": "Feature",
            "geometry": geometry,
            "id": AdmUnitId
        })

geo = {'type': 'FeatureCollection', 'features': lk_geo}

df = pd.json_normalize(resp.json()["features"])
df = df[(df['properties.AdmUnitId'] > 16)] # only counties starting from index 17

fig = px.choropleth(df, # create map
                    geojson=geo,
                    scope="europe",
                    locations="properties.AdmUnitId",
                    color="properties.cases7_per_100k",
                    template="simple_white",
                    hover_name="properties.GEN",
                    hover_data=("properties.cases7_per_100k", "properties.cases7_lk"
                    labels={"properties.AdmUnitId": "Landkreis",
                            "properties.cases7_per_100k": "7-Tage Inzidenz",
                            "properties.death7_lk": "7-Tage Todesfälle",
                            "properties.cases7_lk": "7-Tage Neuinfektionen"},
                    color_continuous_scale=px.colors.sequential.Reds, width=900, hei
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r": 0, "t": 25, "l": 0, "b": 0})
fig.update_layout(title_text="Seven-day incidence per county as per Robert-Koch-Inst
fig.show()
```

Seven-day incidence per county as per Robert-Koch-Institute - 29



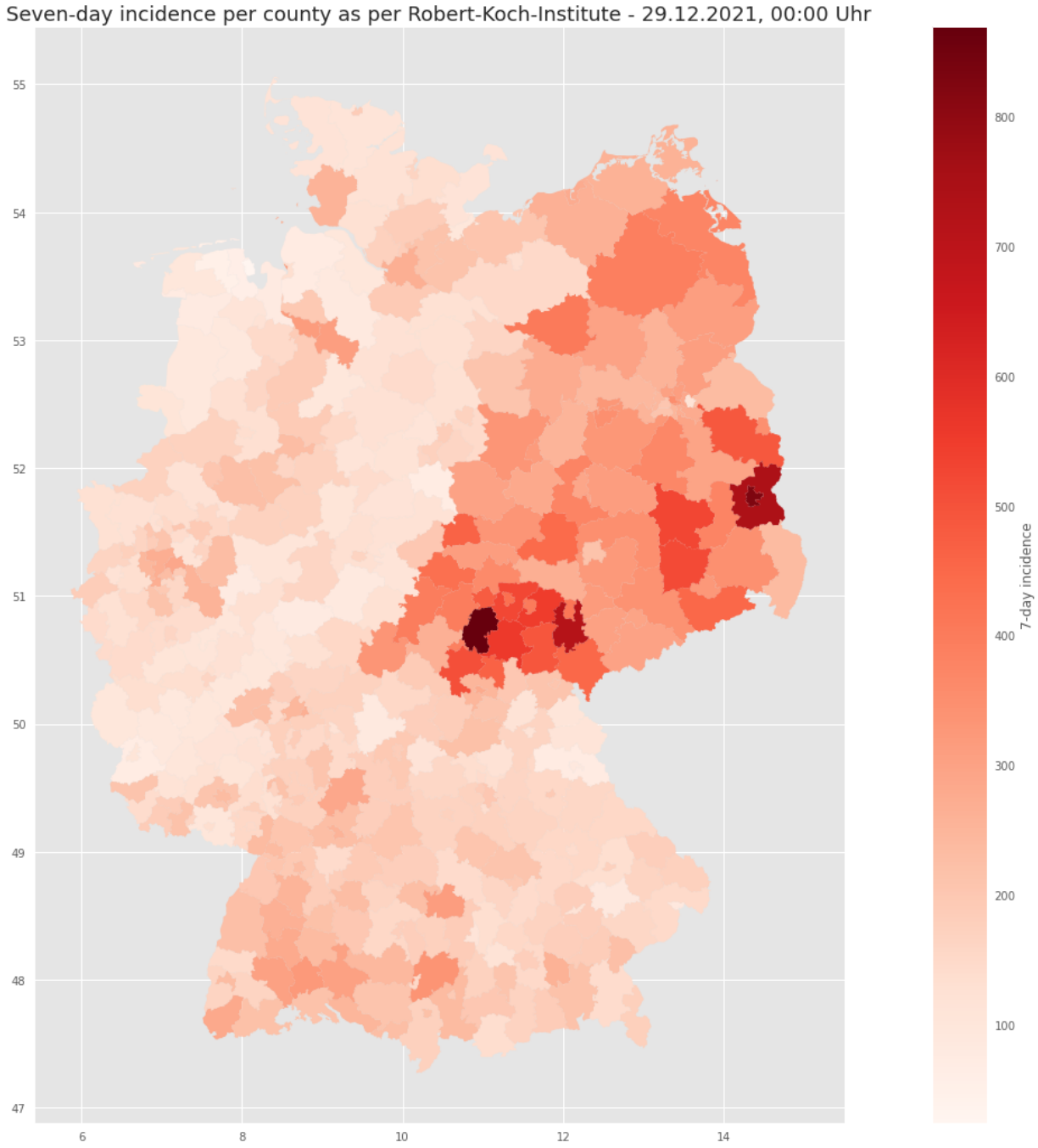
Unfortunately 3 countys as per RKI polygons do not work properly with Plotly Express at least.

Therefore we opt to re-plot the same incidence data as a geopandas image plot.

# 2. Seven-day incidence per county as per Robert-Koch-Institute (Geopandas image plot)

In [25]:
```
import geopandas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
apiurl = 'https://opendata.arcgis.com/datasets/917fc37a709542548cc3be077a786c17_0.ge
df2 = geopandas.read_file(apiurl)
with plt.style.context(("seaborn", "ggplot")):
    df2.plot("cases7_per_100k",
             figsize=(28,18),
             legend=True,
             cmap=plt.cm.Reds,
             legend_kwds={"label":"7-day incidence"}
            )
plt.title("Seven-day incidence per county as per Robert-Koch-Institute - " + str(df2
```




# 3. Yearly average earth surface temperature mapped by country through time

In [39]:

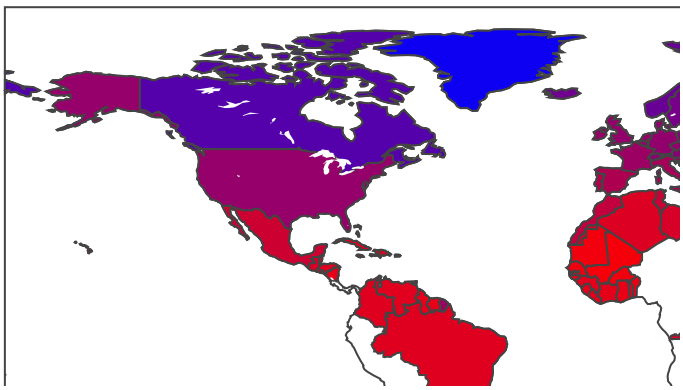
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px; import plotly.io as pio; pio.renderers.default='noteboo
```

```
global_temp_country = pd.read_csv('https://raw.githubusercontent.com/gindeleo/climat
global_temp_country.dropna(axis='index',how='any', subset=['AverageTemperature'],inp
global_temp_country.isna().sum()
global_temp_country['dt'] = pd.to_datetime(global_temp_country['dt'])
global_temp_country['Year'] = global_temp_country['dt'].dt.year

grouped_country_year = global_temp_country[global_temp_country.dt > pd.to_datetime('
```

In [40]:

```
px.defaults.color_continuous_scale = px.colors.sequential.Bluered

# Choropleth Map of the World
fig_cities = px.choropleth(grouped_country_year,locations='Country',
                           locationmode='country names',
                           color='AverageTemperature',
                           animation_frame='Year',
                           range_color = [np.min(grouped_country_year.AverageTempera
                           )
fig_cities.update_layout(title='AverageTemperature ',template="plotly_white")
fig_cities.show()
```

AverageTemperature



# 4. Berlin Xmas market map 2021

In [33]:

```
import folium; from folium import plugins; import geopandas as gpd

berlin_boroughs = 'https://raw.githubusercontent.com/m-hoerz/berlin-shapes/master/be
```

```
berlin_xmas_market_api = "https://www.berlin.de/sen/web/service/maerkte-feste/weihna
gdf = gpd.read_file(berlin_xmas_market_api); gdf['lng'] = gdf.geometry.x; gdf['lat']
```

In [37]:

```
m = folium.Map(location=[52.5, 13.4], tiles='openstreetmap', zoom_start=12, height=9
folium.GeoJson(berlin_boroughs).add_to(m) # boroughs

for id, row in gdf.iterrows(): # Add point markers to the map
    desc = ''
    for key, value in row['data'].items():
        if not key.startswith("rss_"):
            desc += str(key) +  ': ' + str(value) + '</br>'
    folium.Marker(location=[row['lat'],row['lng']], popup=desc).add_to(m)

minimap = folium.plugins.MiniMap(); m.add_child(minimap) # minimap
m.save('my_map.html') # save the map
m # Display the map
```

Out[37]: