Assessing the Impact of Weather on Accidents in German Regions

Introduction

Road safety is an important concern in Germany. Accidents are one of the main indicators of how safe a road is. Weather conditions, particularly the influence of precipitation play a crucial role in these accidents. This project focuses on analyzing the relation between weather factors and road accidents in various German regions.

Accidents involving road vehicles are a significant problem due to their impact on human lives, infrastructure, and the economy. Weather conditions can increase the risk of these happening, but their exact influence needs to be studied.

Using advanced data engineering and analytical methods, in this project will examine historical accident data from 10 Bundesländer/Federal States and detailed weather/precipitation data. This analysis aims to reveal patterns, correlations, and potential causal relationships between weather conditions and accidents.

Main Question

How does weather, particularly seasonal variations, impact the rate and nature of accidents in different regions of Germany?"

Other sub questions/information we can observe

1) How does weather, affect certain road types accidents, eg. in inner town or highways?"

2) How does weather, particularly seasonal variations, impact the rate and nature of accidents in different regions of Germany?"

3) How does weather, particularly seasonal variations, impact the rate and nature of accidents in different regions of Germany?"

Methods

Data Sources

Datasource1: Accidents: Federal states, months, location, severity of injury

This dataset includes road traffic accident statistics for 10 federal states in Germany throughout the timeline [**2022**, **2023 (not complete)**]. It has information for each state divided on the following sub-categories:

- Place/Ortslage
  - inner town/innerorts
  - out of town (without motorways)/außerorts (ohne Autobahnen)
  - on highways/auf Autobahnen
- Severity of injury/Schwere der Verletzung
  - Killed/Getötete
  - Slightly injured/Leichtverletzte
  - Seriously injured/Schwerverletzte
  - In total/Insgesamt

For each place we have all 4 severities of injury and a total for that place. As well as an overall total for the number of accidents for that land.

Datasource2: Regional Averages GE - Monthly - Precipitation

Each dataset includes average precipitation for a given month of all the federal states in Germany from the year **1881** till **2023**.

Both datasets are open source and include all the relevant information to answer the questions being asked.

Work process

The main datapipeline is composed of 2 steps:

- Building a traffic accident dataframe saved into a pickle format
- Building a weather dataframe saved into a pickle format

For the traffic dataset some cleaning needed to be done since the csv includes also some other metadata apart from the main data table, furthermore the empty cells were written as '-' which needed to be replaced with 0. A final transformation was translating the places & severity of the accidents type from German to English.

For the weather dataset the process is trickier since the data of the weather is seperated into different files for each month, where each file has the precipitation records from years **1881** till **2023**. We only need the years 2022 & 2023, so for each month file we only get data for these 2 years. We save this data for every place/state. In the end we construct the weather dataframe with columns: [state] and the rest are all the months with their rain amount.

Problems

Below are some of the encountered problems:

- The traffic accidents dataset is updated at least once a year with the prev years data being deleted. This can cause problems long term.
- The data structure of the traffic data set includes metadata so having to clean up, skip rows and focusing only on the main table information involved a lot of small, hardcoded indexes which can be problematic in case the structure changes just a little, e.g a new row of metadata is added.

Results

Install dependencies and read the datasets with which we are going to work. (The datasets are the output of the pipeline)

```
!pip install pandas
!pip install plotly
!pip install nbformat
!pip install matplotlib
```

```
Requirement already satisfied: pandas in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (1.5.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from pandas) (2022.7.1)
Requirement already satisfied: numpy>=1.21.0 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from pandas) (1.24.2)
Requirement already satisfied: six>=1.5 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from python-dateutil>=2.8.1->pandas) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: plotly in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (5.18.0)
Requirement already satisfied: tenacity>=6.2.0 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from plotly) (23.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: nbformat in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (5.9.2)
Requirement already satisfied: fastjsonschema in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from nbformat) (2.19.1)
Requirement already satisfied: jsonschema>=2.6 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from nbformat) (4.20.0)
```

```
Requirement already satisfied: jupyter-core in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from nbformat) (5.7.1)
Requirement already satisfied: traitlets>=5.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from nbformat) (5.14.1)
Requirement already satisfied: attrs>=22.2.0 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from jsonschema>=2.6->nbformat) (22.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from jsonschema>=2.6->nbformat) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from jsonschema>=2.6->nbformat) (0.32.1)
Requirement already satisfied: rpds-py>=0.7.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from jsonschema>=2.6->nbformat) (0.16.2)
Requirement already satisfied: platformdirs>=2.5 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from jupyter-core->nbformat) (4.1.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: matplotlib in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (3.8.2)
Requirement already satisfied: contourpy>=1.0.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (4.47.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy<2,>=1.21 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (1.24.2)
Requirement already satisfied: packaging>=20.0 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=8 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
```

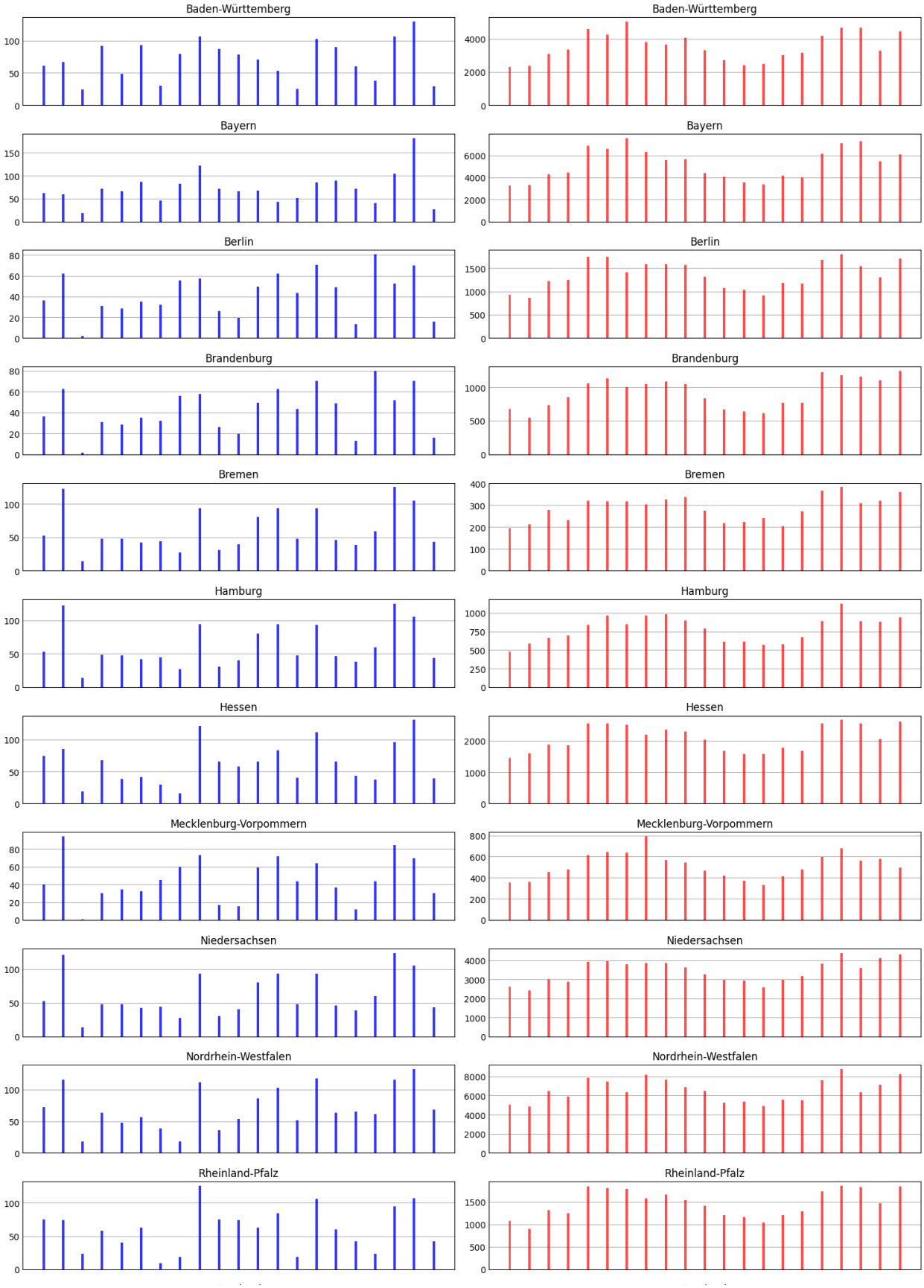
```
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in
/home/kristi/PycharmProjects/made-template/venv/lib/python3.10/site-
packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
```

```
import pandas as pd

rain = pd.read_pickle("../data/precipitation.pkl")
traffic = pd.read_pickle("../data/traffic_accident.pkl")
```

For our main question we can use the below graphs, where side by side we have the rains prec and the total number of accidents in each state.

```
from datetime import datetime

import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')
states = traffic["state"].unique().tolist()
states.sort()

# plot
fig, ax = plt.subplots(len(states), 2, figsize=(15, 30))

years_labels = list(traffic.columns[3:-2])
month_dict = {
    1: 'Jan',
    2: 'Feb',
    3: 'Mar',
    4: 'Apr',
    5: 'May',
    6: 'Jun',
    7: 'Jul',
    8: 'Aug',
    9: 'Sep',
    10: 'Oct',
    11: 'Nov',
    12: 'Dec'
}
years_labels = [
    f"{month_dict[datetime.strptime(date, '%d-%m-%Y').month]}-{date[-
2:]}"
```

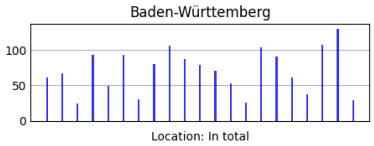
```
    for date in years_labels
]

for state, index in zip(states, range(16)):
    state_rain = rain[rain["state"] == state].iloc[0, 1:-3].values
    state_total_accidents = traffic[
                                (traffic["state"] == state) &
(traffic["place"] == "In total") & (
                                        traffic["severity"] == "In
total")
                                ].iloc[0, 3:-2].values

    ax[index][0].bar(years_labels, state_rain, width=np.pi / 25,
alpha=0.8, color="blue",
                     label='Rain')
    ax[index][0].set_xticks([])
    ax[index][0].set_xticklabels([])

    ax[index][1].bar(years_labels, state_total_accidents, width=np.pi
/ 25, alpha=0.7, color='red',
                     label='Accident')
    ax[index][1].set_xticks([])
    ax[index][1].set_xticklabels([])

    ax[index][0].set_title(state)
    ax[index][1].set_title(state)

plt.tight_layout()
plt.show()
```

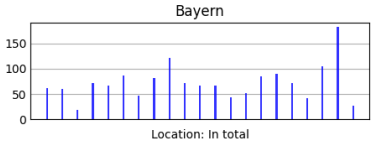

Baden-Württemberg
Baden-Württemberg
Bayern
Bayern
Berlin
Berlin
Brandenburg
Brandenburg
Bremen
Bremen
Hamburg
Hamburg
Hessen
Hessen
Mecklenburg-Vorpommern
Mecklenburg-Vorpommern
Niedersachsen
Niedersachsen
Nordrhein-Westfalen
Nordrhein-Westfalen
Rheinland-Pfalz
Rheinland-Pfalz

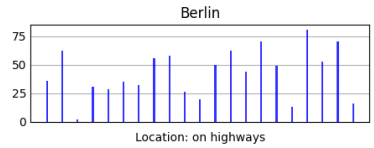
We can see that even in months when the rain precipitation is low the accidents rate is higher.

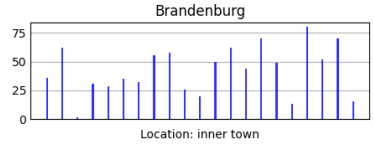
Let us check if it will affect in different location types (to answer our second question)

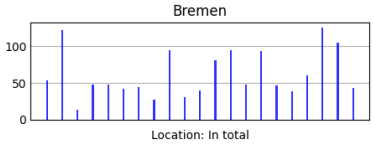
```
import random
places = traffic["place"].unique().tolist()
fig, ax = plt.subplots(len(states), 2, figsize=(15, 30))


for state, index in zip(states, range(16)):
    picked_location = places[random.randint(0, len(places)-1)]
    state_rain = rain[rain["state"] == state].iloc[0, 1:-3].values
    state_total_accidents = traffic[
                                (traffic["state"] == state) &
(traffic["place"] == picked_location) & (
                                    traffic["severity"] == "In
total")
                                ].iloc[0, 3:-2].values

    ax[index][0].bar(years_labels, state_rain, width=np.pi / 25,
alpha=0.8, color="blue",
                     label='Rain')
    ax[index][0].set_xticks([])
    ax[index][0].set_xticklabels([])

    ax[index][1].bar(years_labels, state_total_accidents, width=np.pi
/ 25, alpha=0.7, color='red',
                     label='Accident')
    ax[index][1].set_xticks([])
    ax[index][1].set_xticklabels([])
    # Set custom text
    ax[index, 1].text(0.5, -0.2, f'Location: {picked_location}',
ha='center', transform=ax[index, 0].transAxes)



    ax[index][0].set_title(state)
    ax[index][1].set_title(state)

plt.tight_layout()
plt.show()
```

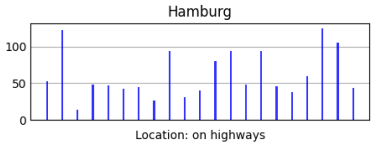

Baden-Württemberg
100
50
0
Location: In total

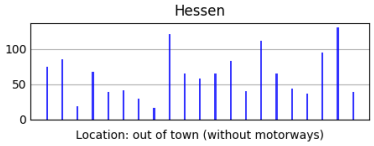

Bayern
150
100
50
0
Location: In total

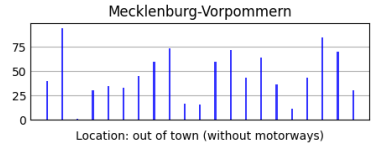

Berlin
75
50
25
0
Location: on highways

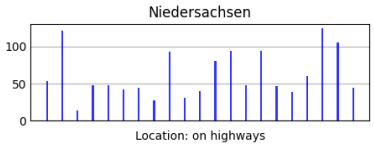

Brandenburg
75
50
25
0
Location: inner town

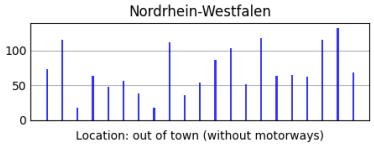

Bremen
100
50
0
Location: In total

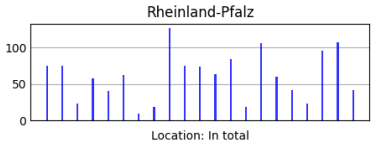

Hamburg
100
50
0
Location: on highways

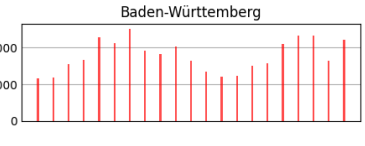

Hessen
100
50
0
Location: out of town (without motorways)

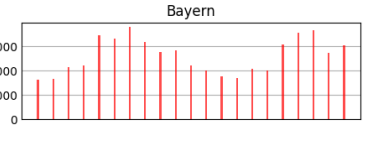

Mecklenburg-Vorpommern
75
50
25
0
Location: out of town (without motorways)

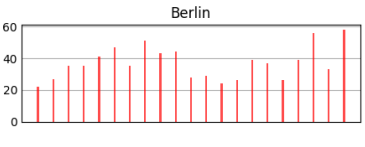

Niedersachsen
100
50
0
Location: on highways

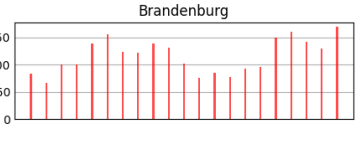

Nordrhein-Westfalen
100
50
0
Location: out of town (without motorways)

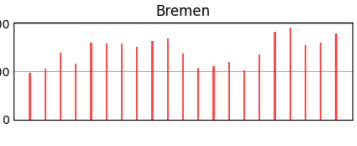

Rheinland-Pfalz
100
50
0
Location: In total

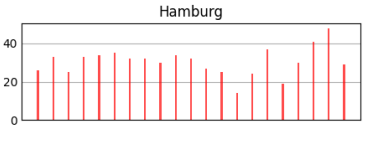

Baden-Württemberg
4000
2000
0

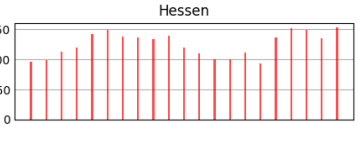

Bayern
6000
4000
2000
0

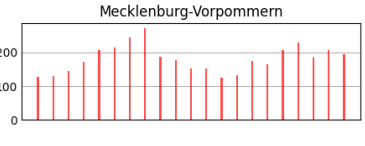

Berlin
60
40
20
0

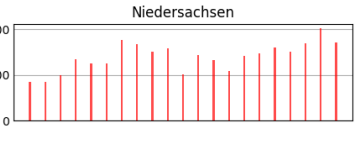

Brandenburg
750
500
250
0

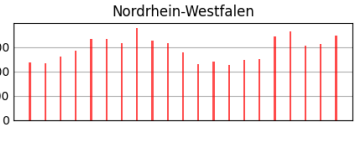

Bremen
400
200
0

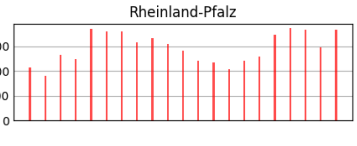

Hamburg
40
20
0


Hessen
750
500
250
0


Mecklenburg-Vorpommern
200
100
0


Niedersachsen
400
200
0


Nordrhein-Westfalen
1500
1000
500
0


Rheinland-Pfalz
1500
1000
500
0

We can arrive at the same conclusion that even in different specific locations rain is not a big factor in the accidents number.

Discussion/Conclusions

We can see that rain does not necessarily affect the accidents rate in the state of Germany. Other factors such as:

- the road conditions,
- driving under the influence,
- car conditions
- traffic numbers (cars crossing at the time)

might play a bigger role in accidents number. Also rain does not necessarily mean that weather does not play a role, maybe snow or foggy weather has a bigger affect.

Another thing to account which is a limitation in this project as well is the amount of data for the accidents, which is expended in a year and a half period. I believe that taking all the others factor as well as having accident data throughout all the year can give a more solid answer weather or not rain affects the accident rates.