Exploratory Data Analysis: 23th March, 2024

In [1]:
```
# importing libraries

import pandas as pd
import numpy as np

import matplotlib

import matplotlib.pyplot as plt
import seaborn as sns

from matplotlib.ticker import StrMethodFormatter
```

```
C:\ProgramData\Anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumP
y version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected versio
n 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
# Loading the final cleaned dataset with extended variables

df = pd.read_csv('data_cleaned_descriptive_analysis_final.csv', header = 0)

pd.set_option('display.max_columns', None)
df.head(9)
```

Out[2]:

	Unnamed: 0	no_cars	gender	age	income	political_party	education	postal_code	EUROS
0	25	1.0	Weiblich	65	3000.0	CDU/CSU	(Fach-) Hochschulabschluss (Bachelor, Master, ...	66440	
1	26	2.0	Weiblich	59	800.0	Keine Angabe	Allgemeine oder fachgebundene Hochschulreife/A...	65933	
2	27	0.0	Weiblich	60	1750.0	Keine Angabe	Berufsausbildung, Lehre oder Ausbildung an ein...	95028	
3	28	1.0	Männlich	73	2500.0	SPD	Realschulabschluss (Mittlere Reife) oder gleic...	63741	
4	30	0.0	Männlich	43	2500.0	Einer anderen Partei	Berufsausbildung, Lehre oder Ausbildung an ein...	13059	
5	31	1.0	Weiblich	49	2300.0	CDU/CSU	Berufsausbildung, Lehre oder Ausbildung an ein...	39112	
6	32	1.0	Weiblich	57	600.0	CDU/CSU	Realschulabschluss (Mittlere Reife) oder gleic...	78244	
7	33	2.0	Männlich	39	5000.0	SPD	(Fach-) Hochschulabschluss (Bachelor, Master, ...	10115	
8	34	2.0	Männlich	62	0.0	Keine Angabe	(Fach-) Hochschulabschluss (Bachelor, Master, ...	46149	

In [3]:
```
# Names of the columns in this dataframe
df.columns
```

Out[3]:
```
Index(['Unnamed: 0', 'no_cars', 'gender', 'age', 'income', 'political_party',
       'education', 'postal_code', 'EUROSTAT', 'RLK2022', 'KTU2022',
       'federal_state', 'NUTS2_NAME', 'NUTS3_NAME', 'CO2_housing',
       'CO2_electricity', 'CO2_housing_electricity', 'CO2_cruise',
       'CO2_flight', 'CO2_public_transport', 'CO2_car1', 'CO2_car2',
       'CO2_car3', 'CO2_car4', 'CO2_car5', 'CO2_car_total', 'CO2_mobility',
       'CO2_food', 'CO2_other_consumption', 'public_emission', 'CO2_total',
       'belief_housing_electricity', 'belief_mobility', 'belief_food',
       'belief_other_consumption', 'belief_total',
       'actual_rank_CO2_housing_electricity1', 'actual_rank_CO2_mobility1',
       'actual_rank_CO2_food1', 'actual_rank_CO2_other_consumption1',
       'actual_rank_CO2_total1', 'actual_rank_CO2_housing_electricity2',
       'actual_rank_CO2_mobility2', 'actual_rank_CO2_food2',
       'actual_rank_CO2_other_consumption2', 'actual_rank_CO2_total2',
       'belief_diff_housing_electricity', 'belief_diff_mobility',
       'belief_diff_food', 'belief_diff_other_consumption',
       'belief_diff_total'],
      dtype='object')
```

In [4]:
```
df['urban_rural_class'] = df['RLK2022']
```

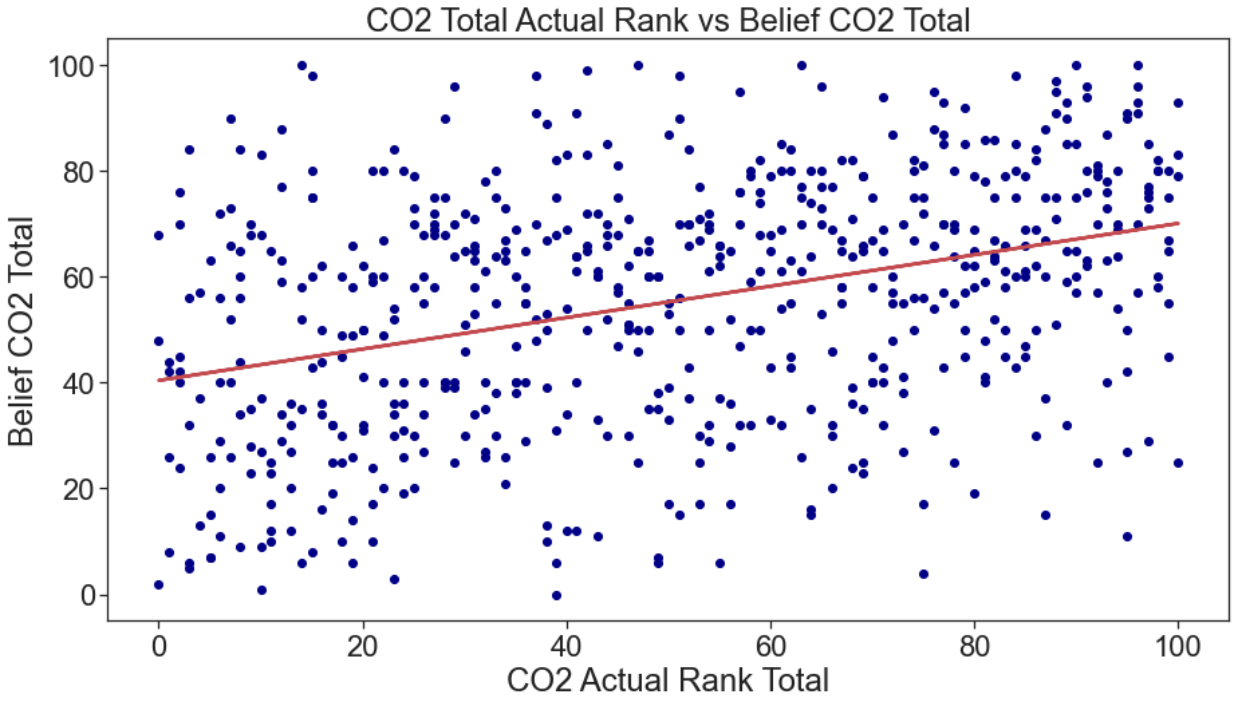
In [5]:
```
# There are total of 424 data points collected

len(df)
```

Out[5]:
```
588
```

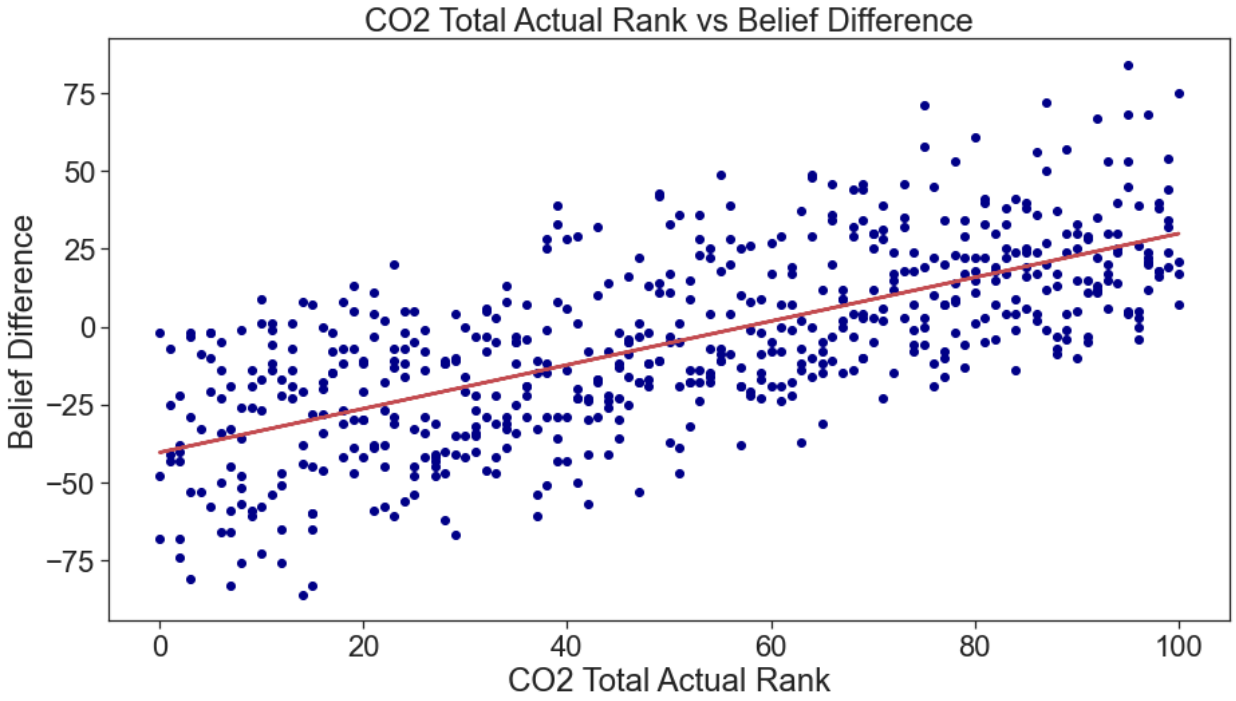
In [6]:
```
# Setting the graph style
sns.set(font_scale = 2)
sns.set_style("ticks")
```

1. Scatterplots for Actual vs Belief

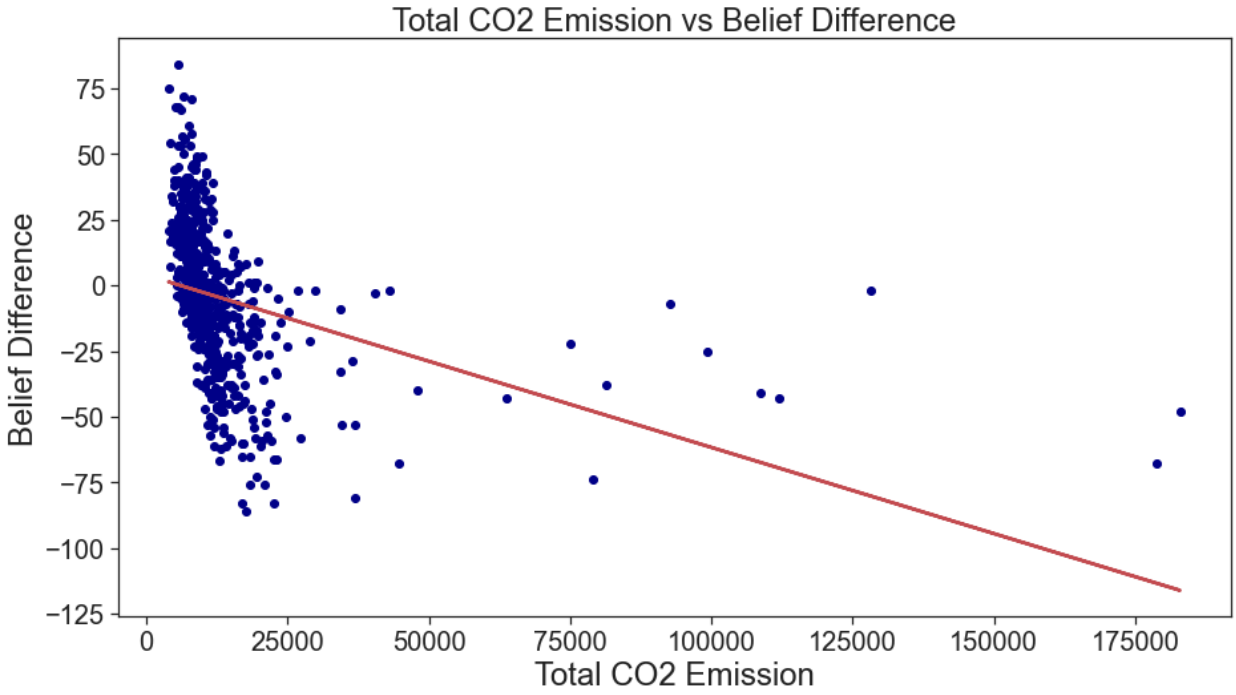
In [7]:
```
# Scatterplot: CO2 Total Actual Rank vs Belief CO2 Total

sns.set_style("ticks")

x = df['actual_rank_CO2_total2'] # ending with 2 is scaled rank in a group of 100 peop
y = df['belief_total'] # answer from the respondent, how many people have higher than

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Actual Rank vs Belief CO2 Total')
plt.xlabel('CO2 Actual Rank Total')
plt.ylabel('Belief CO2 Total')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



In [8]:
```
# Scatterplot: CO2 Total vs Belief Difference

sns.set_style("ticks")

x = df['actual_rank_CO2_total2'] # ending with 2 is scaled rank in a group of 100 peop
y = df['belief_diff_total'] # answer from the respondent

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Actual Rank vs Belief Difference')
plt.xlabel('CO2 Total Actual Rank')
plt.ylabel('Belief Difference')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



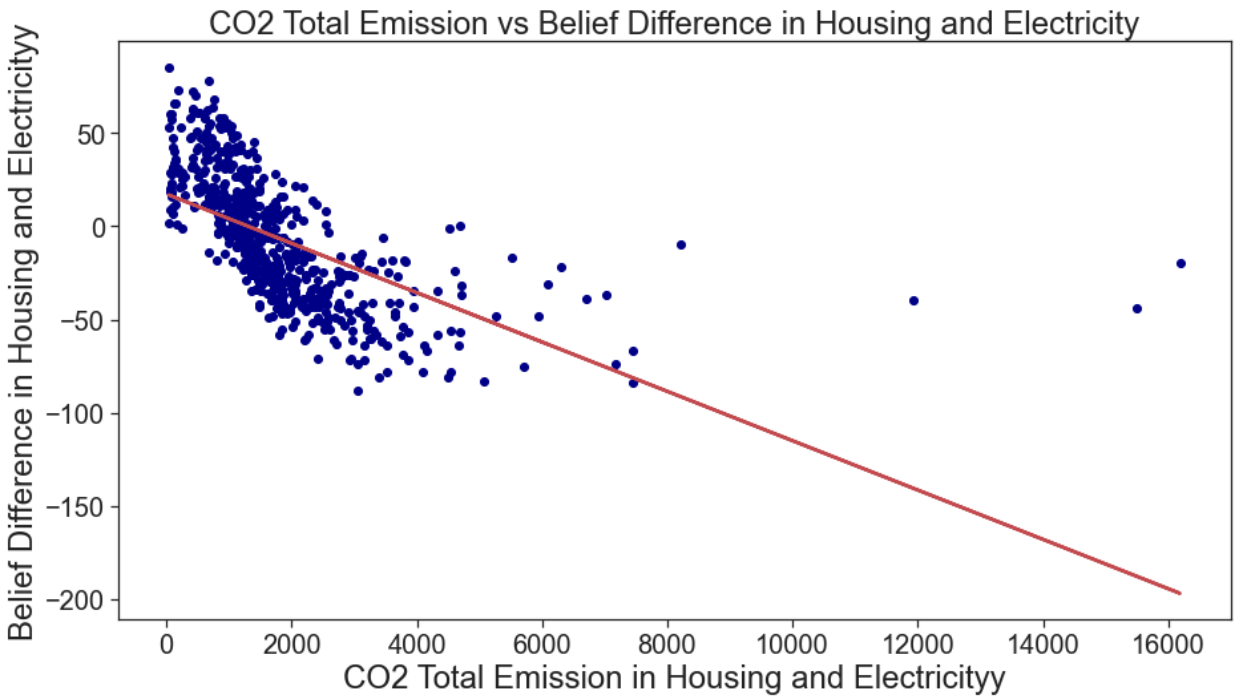
In [9]:
```
# Scatterplot: CO2 Total Emission vs Belief difference

sns.set_style("ticks")

x = df['CO2_total']
y = df['belief_diff_total']

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('Total CO2 Emission vs Belief Difference')
plt.xlabel('Total CO2 Emission')
plt.ylabel('Belief Difference')

plt.tick_params(labelsize=20)


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



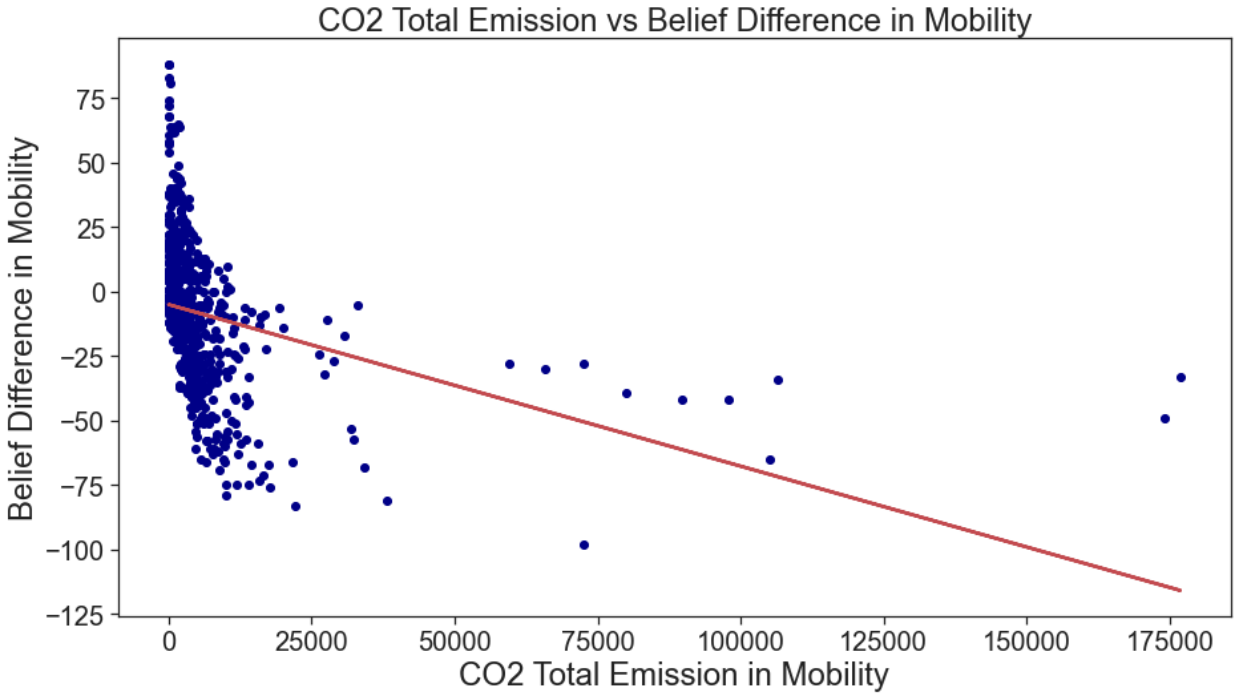
2. Deepdive for each categories in CO2 Emission

CO2 Emission vs Belief difference

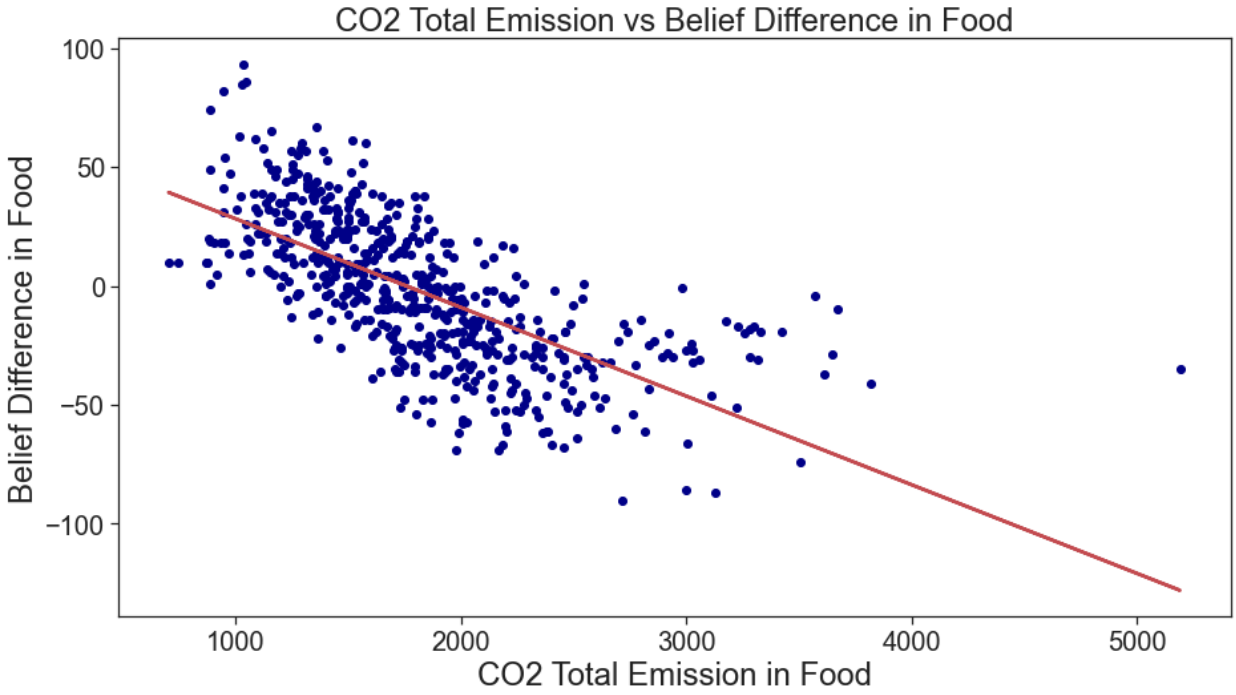
In [10]:
```
# Scatterplot: CO2 Total Emission vs Belief difference

sns.set_style("ticks")

x = df['CO2_housing_electricity']
y = df['belief_diff_housing_electricity']

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Emission vs Belief Difference in Housing and Electricity')
plt.xlabel('CO2 Total Emission in Housing and Electricityy')
plt.ylabel('Belief Difference in Housing and Electricityy')

plt.tick_params(labelsize=20)


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



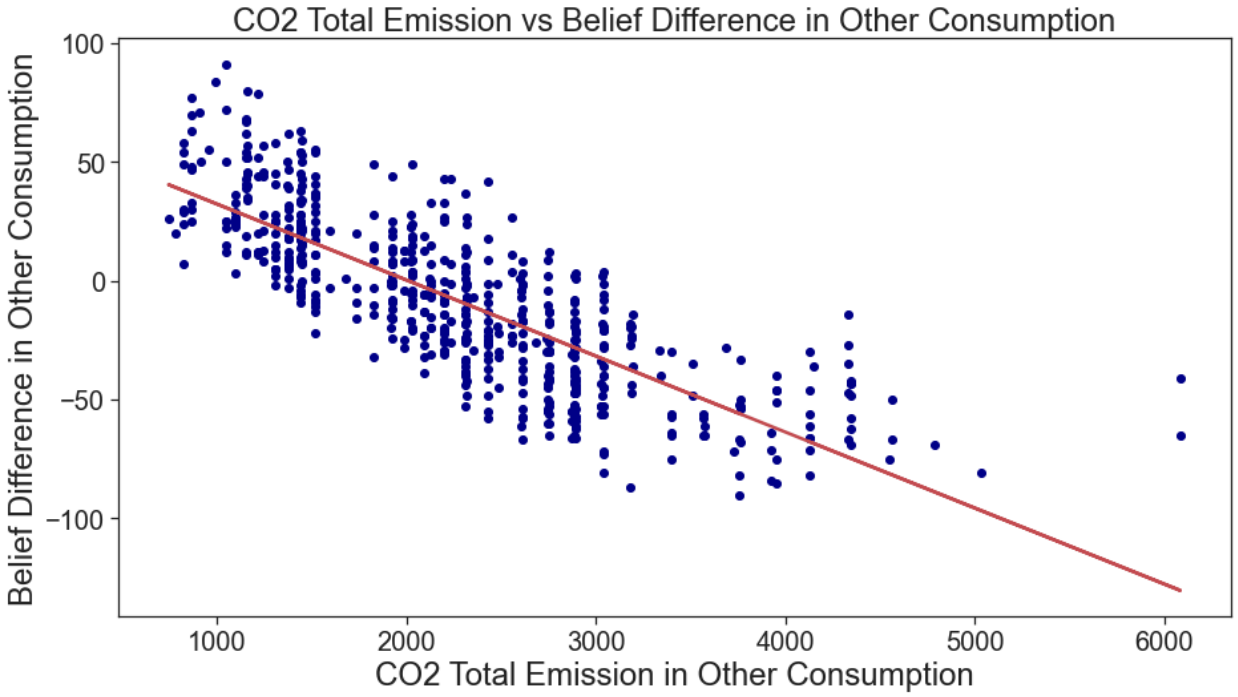
In [11]:
```
# Scatterplot: CO2 Total Emission vs Belief difference

sns.set_style("ticks")

x = df['CO2_mobility']
y = df['belief_diff_mobility']

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Emission vs Belief Difference in Mobility')
plt.xlabel('CO2 Total Emission in Mobility')
plt.ylabel('Belief Difference in Mobility')

plt.tick_params(labelsize=20)


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



In [12]:
```
# Scatterplot: CO2 Total Emission vs Belief difference

sns.set_style("ticks")

x = df['CO2_food']
y = df['belief_diff_food']

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Emission vs Belief Difference in Food')
plt.xlabel('CO2 Total Emission in Food')
plt.ylabel('Belief Difference in Food')

plt.tick_params(labelsize=20)


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```

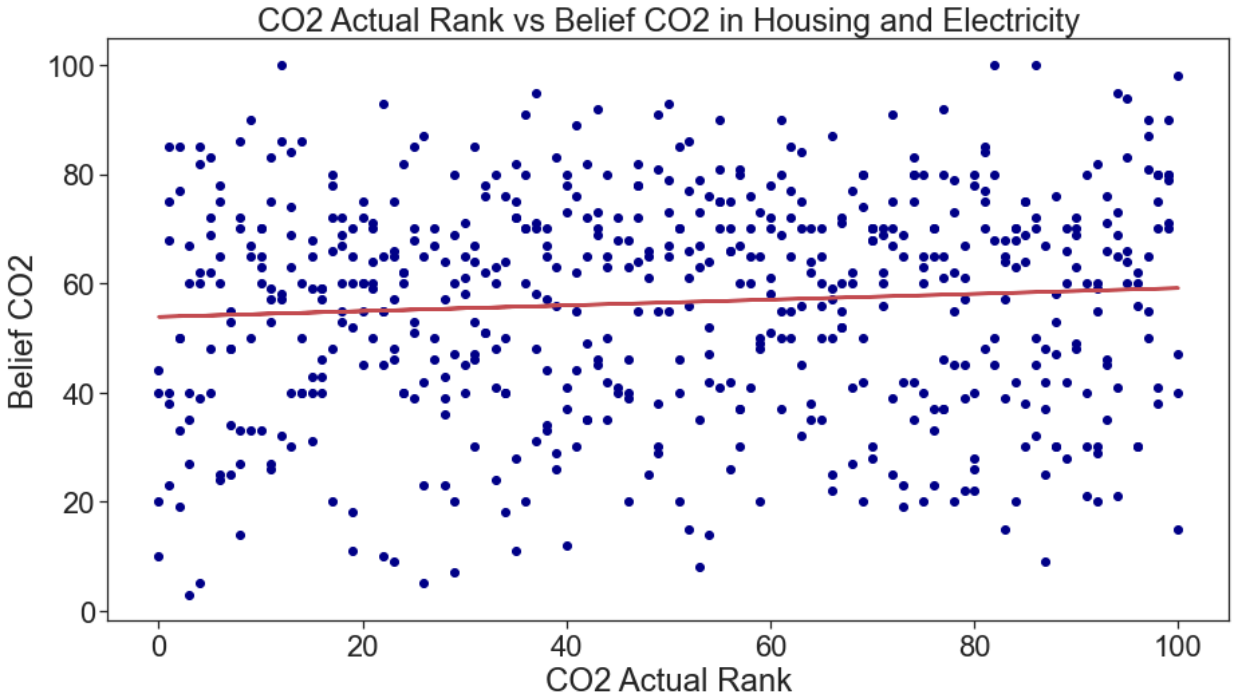


In [13]:
```
# Scatterplot: CO2 Total Emission vs Belief difference

sns.set_style("ticks")

x = df['CO2_other_consumption']
y = df['belief_diff_other_consumption']

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Total Emission vs Belief Difference in Other Consumption')
plt.xlabel('CO2 Total Emission in Other Consumption')
plt.ylabel('Belief Difference in Other Consumption')

plt.tick_params(labelsize=20)


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```

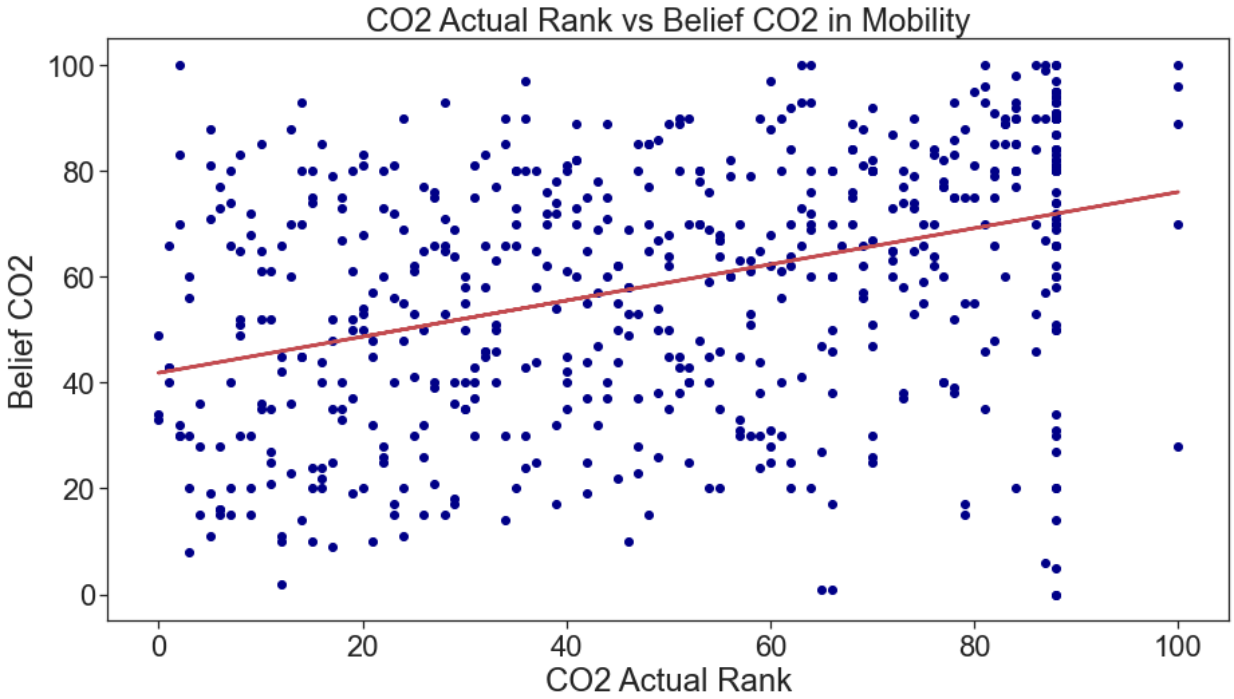


CO2 Rank vs Belief Answer

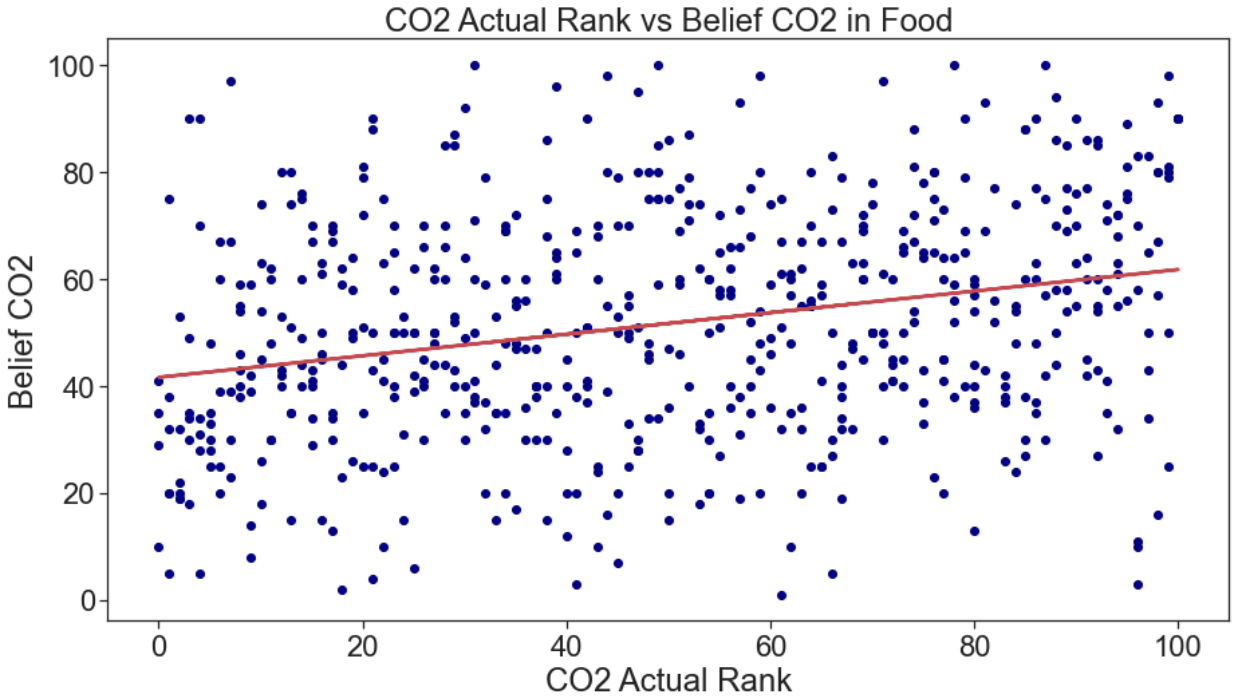
In [14]:
```
# Scatterplot: CO2 Actual Rank vs Belief CO2 in Housing and Electricity

sns.set_style("ticks")

x = df['actual_rank_CO2_housing_electricity2'] # ending with 2 is scaled rank in a gro
y = df['belief_housing_electricity'] # answer from the respondent

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Actual Rank vs Belief CO2 in Housing and Electricity')
plt.xlabel('CO2 Actual Rank')
plt.ylabel('Belief CO2')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



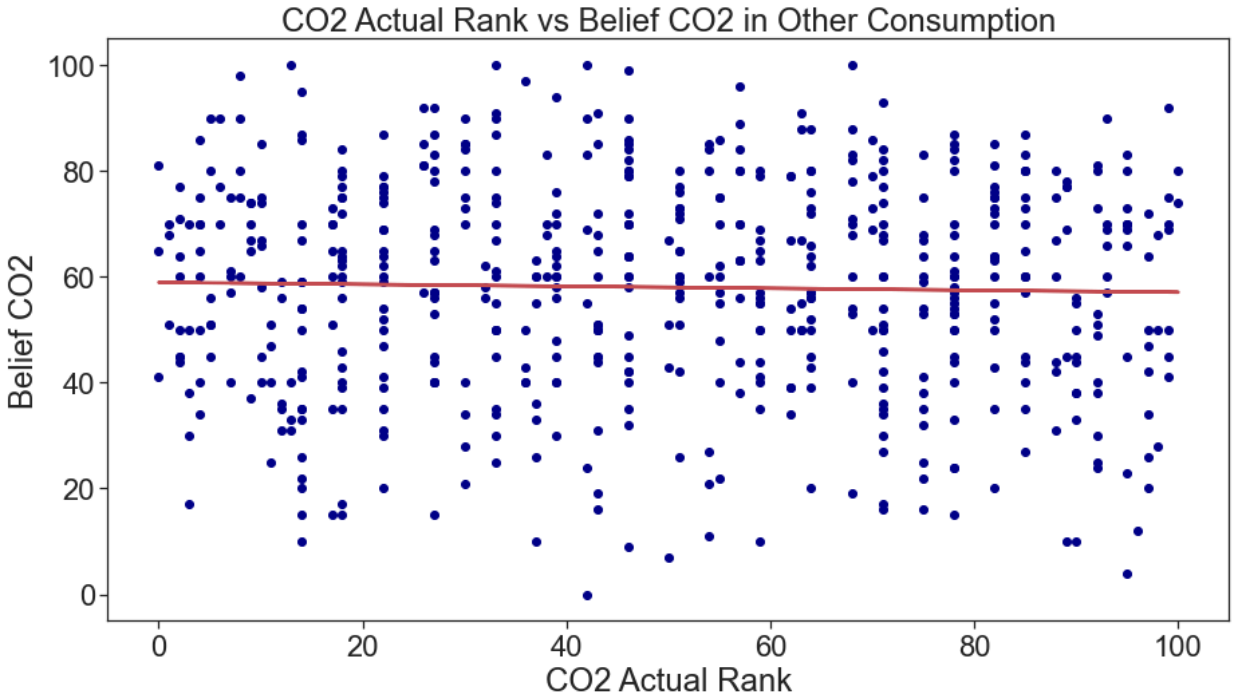
In [15]:
```
# Scatterplot: CO2 Actual Rank vs Belief CO2 in Mobility

sns.set_style("ticks")

x = df['actual_rank_CO2_mobility2'] # the smaller the higher CO2 footprint
y = df['belief_mobility'] # the smaller underestimate

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Actual Rank vs Belief CO2 in Mobility')
plt.xlabel('CO2 Actual Rank')
plt.ylabel('Belief CO2')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



In [16]:
```
# Scatterplot: CO2 Actual Rank vs Belief CO2 in Food

sns.set_style("ticks")

x = df['actual_rank_CO2_food2'] # ending with 2 is scaled rank in a group of 100 peopl
y = df['belief_food'] # answer from the respondent

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Actual Rank vs Belief CO2 in Food')
plt.xlabel('CO2 Actual Rank')
plt.ylabel('Belief CO2')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



In [17]:
```
# Scatterplot: CO2 Actual Rank vs Belief CO2 in Other Consumption

sns.set_style("ticks")

x = df['actual_rank_CO2_other_consumption2'] # ending with 2 is scaled rank in a group
y = df['belief_other_consumption'] # answer from the respondent

fig, ax = plt.subplots(figsize = (15, 8))

ax.scatter(x, y, c='DarkBlue')
plt.title('CO2 Actual Rank vs Belief CO2 in Other Consumption')
plt.xlabel('CO2 Actual Rank')
plt.ylabel('Belief CO2')


# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (a) and intercept (b)
# deg=1 means linear fit (i.e. polynomial of degree 1)
a, b = np.polyfit(x, y, deg=1)

# Plot regression line
ax.plot(x, a*x + b, color="r", lw=2.5);
```



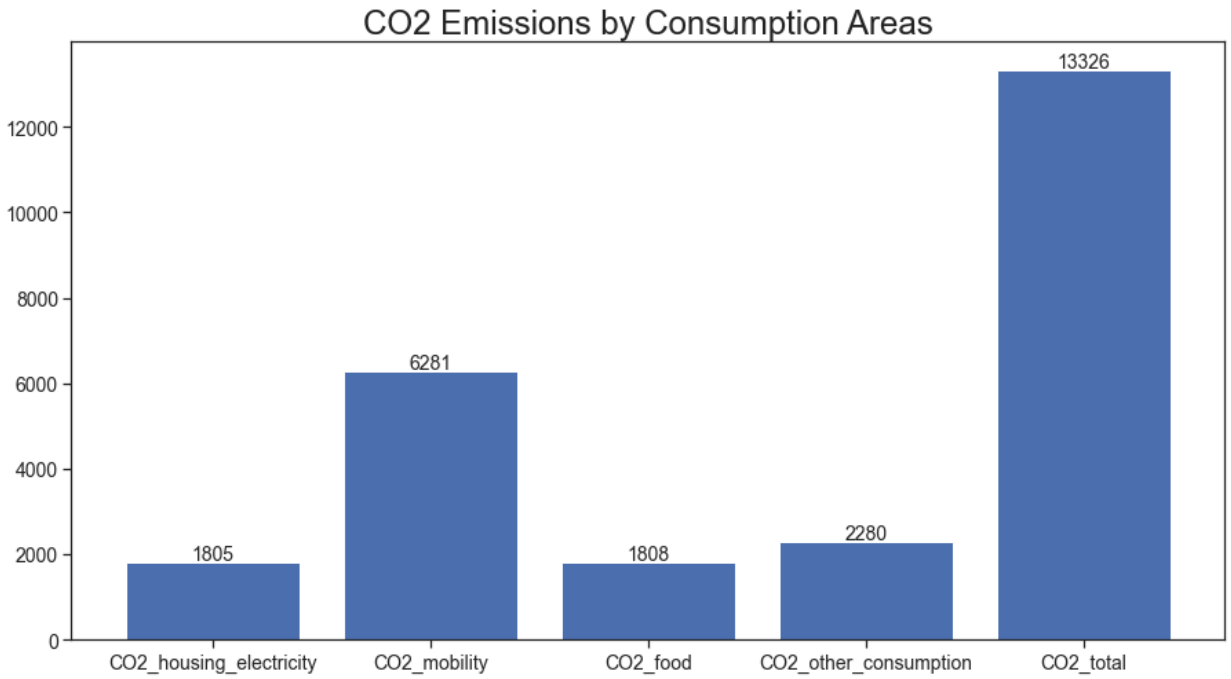
3. Deepdive for Urban and Rural Classification

In [18]:
```
df.columns
```

Out[18]:
```
Index(['Unnamed: 0', 'no_cars', 'gender', 'age', 'income', 'political_party',
       'education', 'postal_code', 'EUROSTAT', 'RLK2022', 'KTU2022',
       'federal_state', 'NUTS2_NAME', 'NUTS3_NAME', 'CO2_housing',
       'CO2_electricity', 'CO2_housing_electricity', 'CO2_cruise',
       'CO2_flight', 'CO2_public_transport', 'CO2_car1', 'CO2_car2',
       'CO2_car3', 'CO2_car4', 'CO2_car5', 'CO2_car_total', 'CO2_mobility',
       'CO2_food', 'CO2_other_consumption', 'public_emission', 'CO2_total',
       'belief_housing_electricity', 'belief_mobility', 'belief_food',
       'belief_other_consumption', 'belief_total',
       'actual_rank_CO2_housing_electricity1', 'actual_rank_CO2_mobility1',
       'actual_rank_CO2_food1', 'actual_rank_CO2_other_consumption1',
       'actual_rank_CO2_total1', 'actual_rank_CO2_housing_electricity2',
       'actual_rank_CO2_mobility2', 'actual_rank_CO2_food2',
       'actual_rank_CO2_other_consumption2', 'actual_rank_CO2_total2',
       'belief_diff_housing_electricity', 'belief_diff_mobility',
       'belief_diff_food', 'belief_diff_other_consumption',
       'belief_diff_total', 'urban_rural_class'],
      dtype='object')
```

In [19]:
```
## breakdown of the CO2 CF sources

sns.set_style("ticks")

ax2 = df[['CO2_housing_electricity','CO2_mobility', 'CO2_food','CO2_other_consumption'
#ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),fontsize=14,
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
                 ha='center',
                 va='center')
```

```
plt.tick_params(labelsize=14)
```



In [20]:
```
df_summary = pd.DataFrame(df[['CO2_housing_electricity','CO2_mobility', 'CO2_food','CO
df_summary
```

Out[20]:

	0
CO2_housing_electricity	1804.841717
CO2_mobility	6281.433010
CO2_food	1807.743657
CO2_other_consumption	2280.010132
CO2_total	13326.028517

In [21]:
```
import plotly.graph_objects as go
```

In [22]:
```
1805+6281+ 1808+ 2280+1152
```

Out[22]:
```
13326
```

In [23]:
```
## breakdown of the CO2 CF sources - waterfall chart

fig = go.Figure(go.Waterfall(
    name = "20", orientation = "v",
    measure = ["relative", "relative","relative", "relative", "relative", "total"],
    x = ['Housing Electricity','Mobility', 'Food','Other Consumption', 'Public Emissio
    textposition = "outside",
    text = ["1805", "6281", "1808", "2280", "1152", "13326"],
    y = [1805, 6281, 1808, 2280, 1152, 13326],
    connector = {"line":{"color":"rgb(63, 63, 63)"}},
))

fig.update_layout(
```

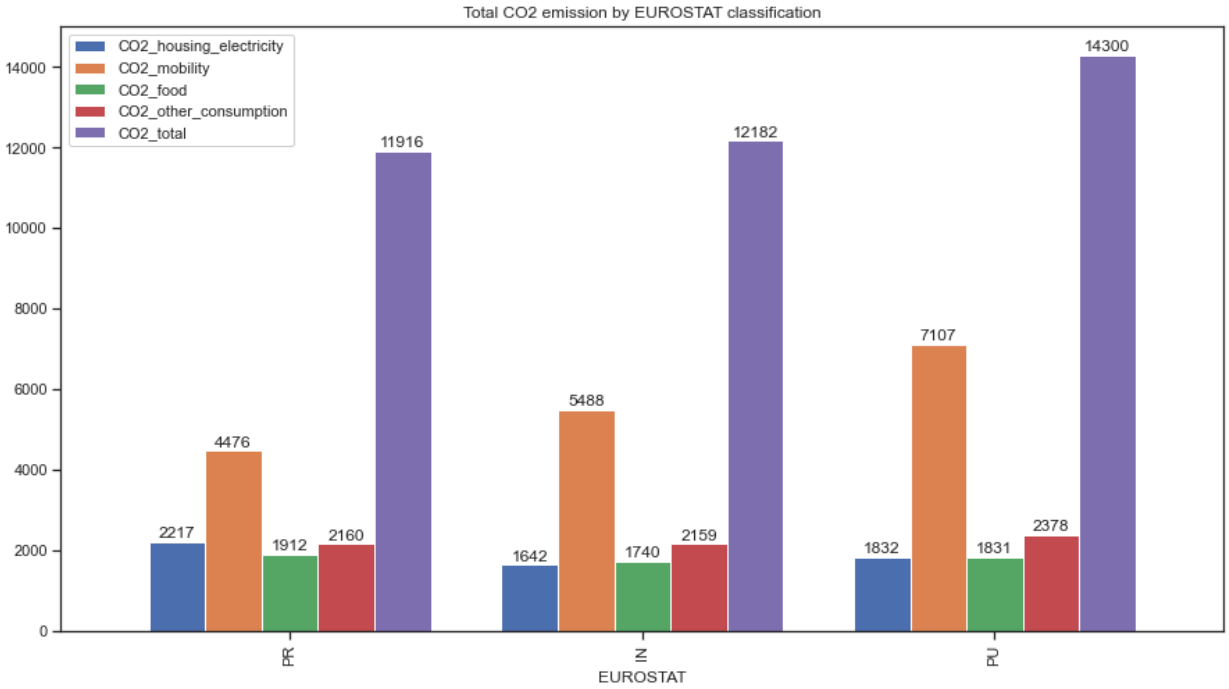
```
        title = "CO2 Emissions by Consumption Areas (unit: CO2 equivalent [kg/year])",
        showlegend = False,
        plot_bgcolor='rgba(0, 0, 0, 0)',
        font_size=14, height = 600, font_family= 'Arial'
)
fig.show()
```

In [24]:
```
# Setting the graph style
sns.set(font_scale = 1)
sns.set_style("ticks")
```

In [25]:
```
### EUROSTAT Classification

sns.set_style("ticks")

ax2 = df.groupby('EUROSTAT')['CO2_housing_electricity','CO2_mobility', 'CO2_food','CO2
ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
```

```
                ha='center',
                va='center')
```

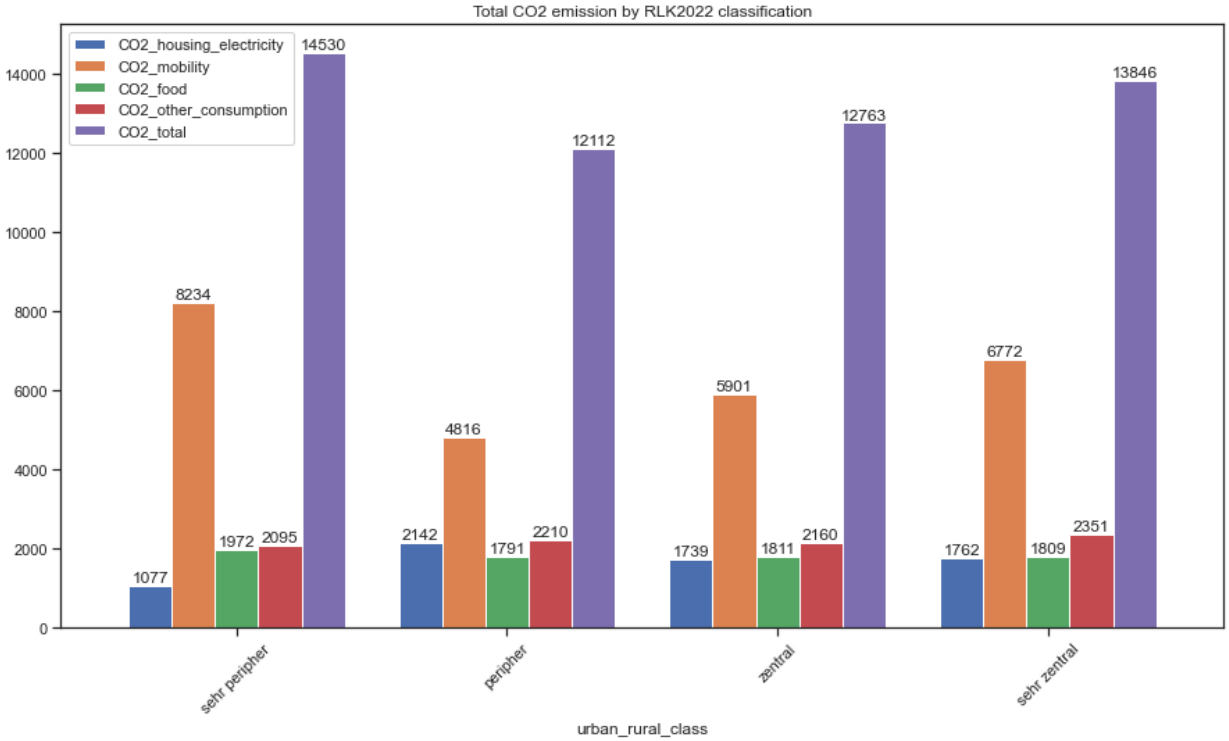
```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\1015744935.py:5: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



In [26]:

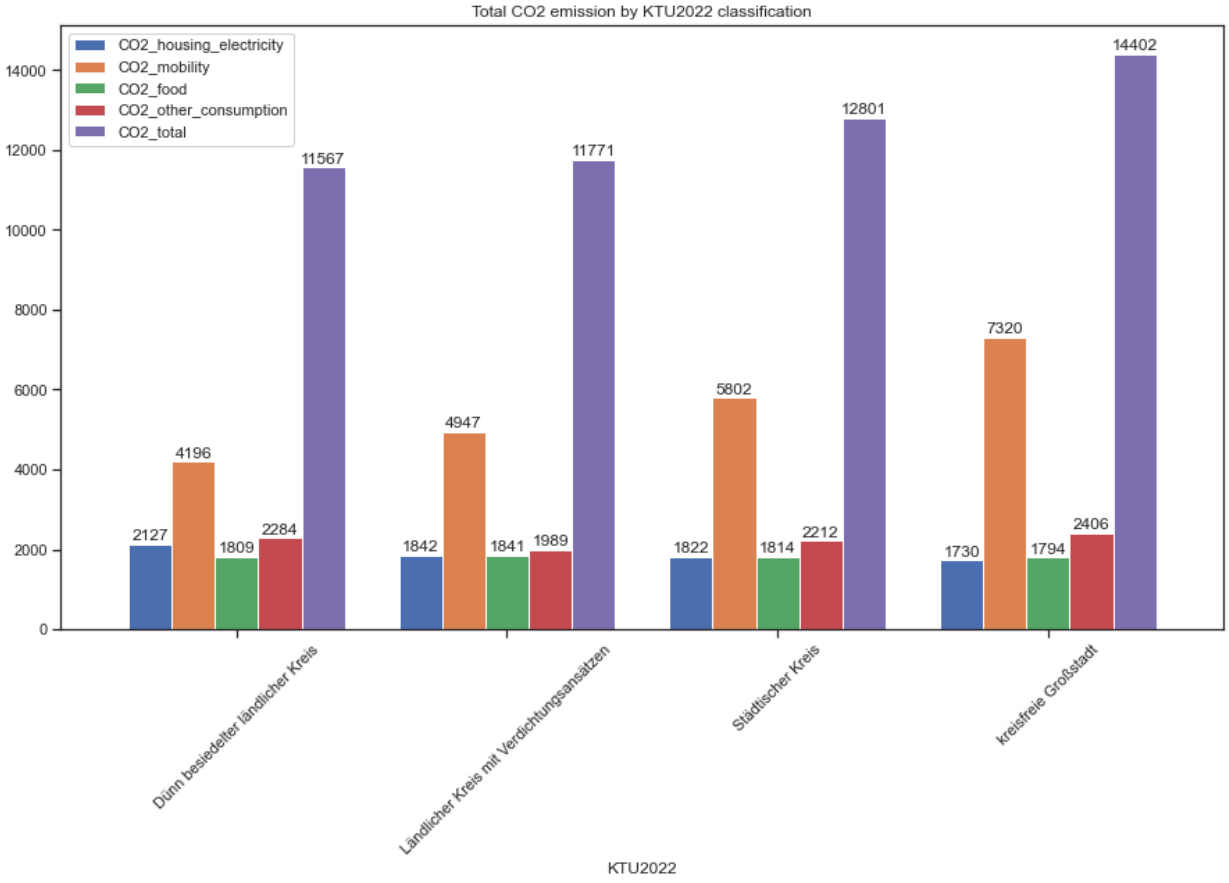
```
### RLK2022 Classification

sns.set_style("ticks")

ax2 = df.groupby('urban_rural_class')['CO2_housing_electricity','CO2_mobility', 'CO2_f
ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),
                xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
                ha='center',
                va='center')
```

```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\4052282977.py:5: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



In [27]:
```
### KTU2022 Classification

sns.set_style("ticks")

ax2 = df.groupby('KTU2022')['CO2_housing_electricity','CO2_mobility', 'CO2_food','CO2_
ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
                 ha='center',
                 va='center')
```

```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\1938877123.py:5: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



4. Deepdive for CF in Mobility

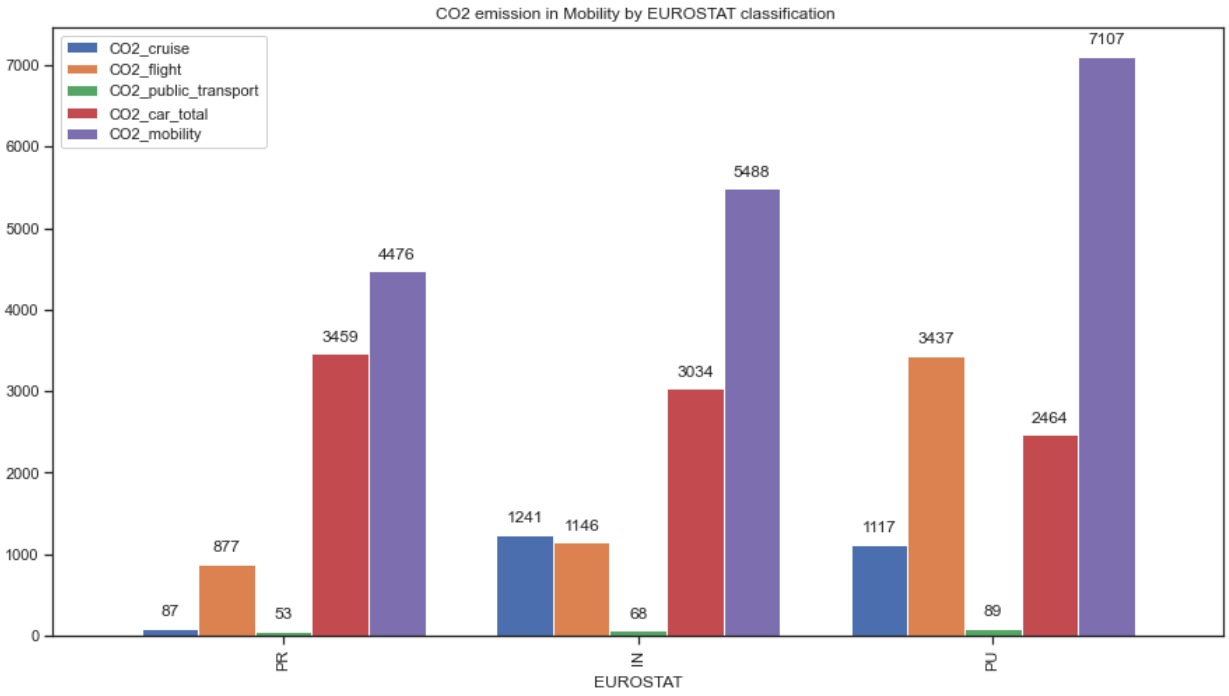
In [28]:
```
df.columns
```

Out[28]:
```
Index(['Unnamed: 0', 'no_cars', 'gender', 'age', 'income', 'political_party',
       'education', 'postal_code', 'EUROSTAT', 'RLK2022', 'KTU2022',
       'federal_state', 'NUTS2_NAME', 'NUTS3_NAME', 'CO2_housing',
       'CO2_electricity', 'CO2_housing_electricity', 'CO2_cruise',
       'CO2_flight', 'CO2_public_transport', 'CO2_car1', 'CO2_car2',
       'CO2_car3', 'CO2_car4', 'CO2_car5', 'CO2_car_total', 'CO2_mobility',
       'CO2_food', 'CO2_other_consumption', 'public_emission', 'CO2_total',
       'belief_housing_electricity', 'belief_mobility', 'belief_food',
       'belief_other_consumption', 'belief_total',
       'actual_rank_CO2_housing_electricity1', 'actual_rank_CO2_mobility1',
       'actual_rank_CO2_food1', 'actual_rank_CO2_other_consumption1',
       'actual_rank_CO2_total1', 'actual_rank_CO2_housing_electricity2',
       'actual_rank_CO2_mobility2', 'actual_rank_CO2_food2',
       'actual_rank_CO2_other_consumption2', 'actual_rank_CO2_total2',
       'belief_diff_housing_electricity', 'belief_diff_mobility',
       'belief_diff_food', 'belief_diff_other_consumption',
       'belief_diff_total', 'urban_rural_class'],
      dtype='object')
```

In [29]:
```
### EUROSTAT Classification

sns.set_style("ticks")

ax2 = df.groupby('EUROSTAT')['CO2_cruise', 'CO2_flight', 'CO2_public_transport','CO2_c
ax2.legend(loc='upper left')
```

```
for p in ax2.patches:
    ax2.annotate(round(p.get_height()),
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
                 ha='center',
                 va='center')
```

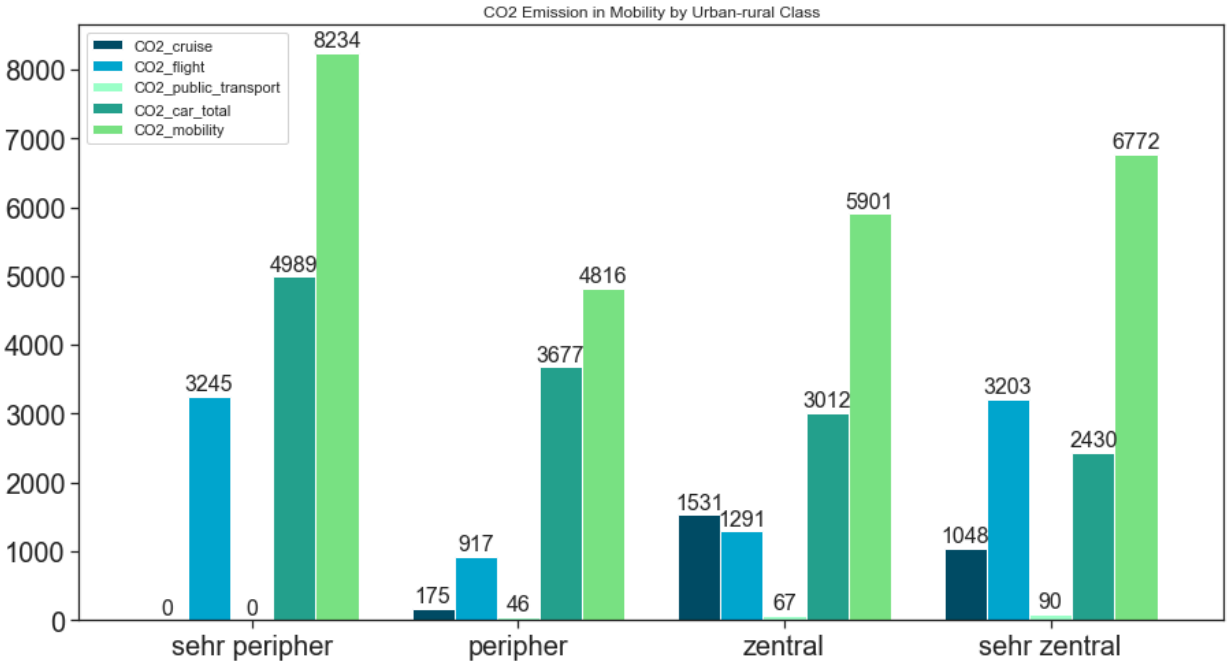
```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\1159766974.py:5: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



In [30]:

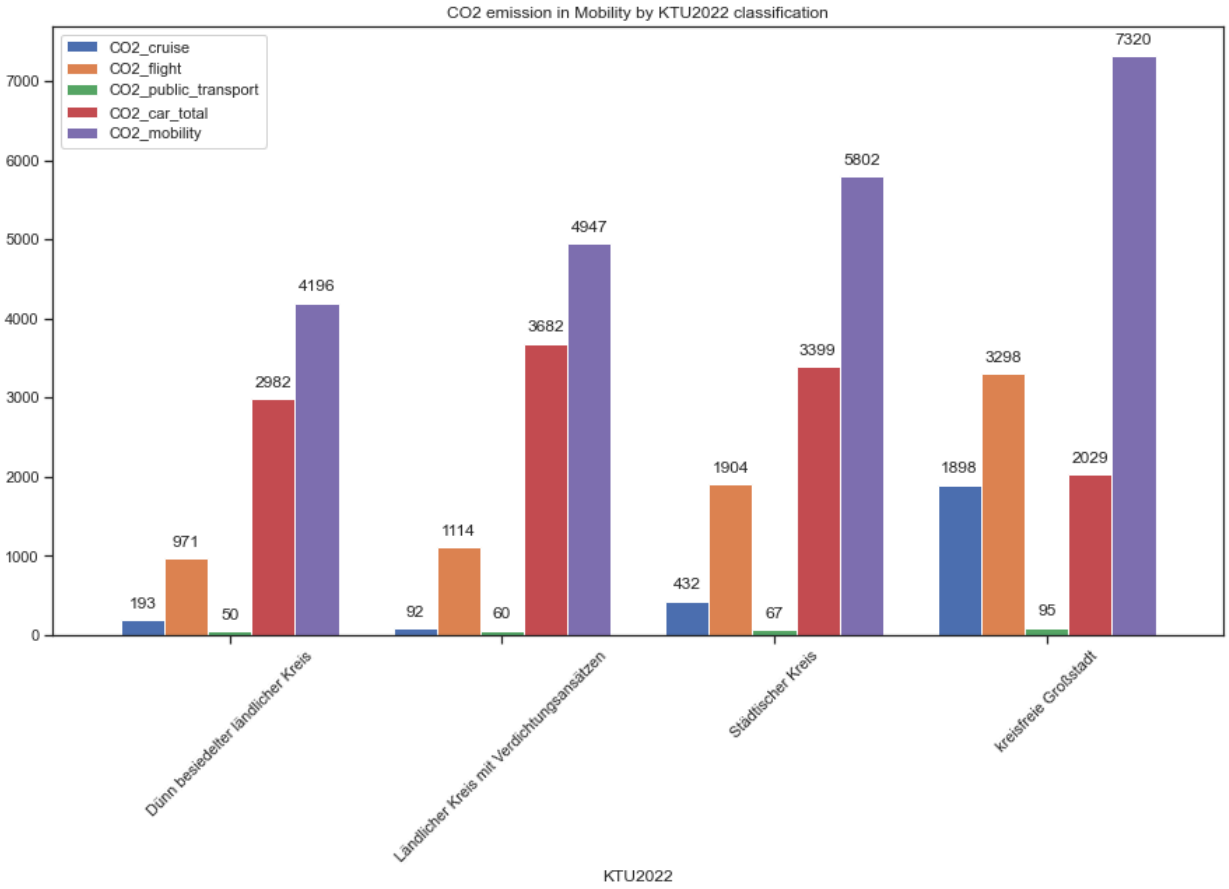
```
### RLK2022 Classification

sns.set_style("ticks")

color = ['#004e64', '#00A5CF', '#9FFFCB', '#25A18E', '#7AE582']


ax2 = df.groupby('urban_rural_class')['CO2_cruise', 'CO2_flight', 'CO2_public_transpor
ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),fontsize=16,
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 180),
                 ha='center',
                 va='center')

plt.tick_params(labelsize=20)

x_axis = ax2.xaxis
x_axis.label.set_visible(False)
```

```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\3616765604.py:8: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



In [31]:
```
### KTU2022 Classification
sns.set_style("ticks")

ax2 = df.groupby('KTU2022')['CO2_cruise', 'CO2_flight', 'CO2_public_transport','CO2_ca
ax2.legend(loc='upper left')

for p in ax2.patches:
    ax2.annotate(round(p.get_height()),
                 xy=(p.get_x()+p.get_width()/2., p.get_height()+ 200),
                 ha='center',
                 va='center')
```

```
C:\Users\leajo\AppData\Local\Temp\ipykernel_22624\3932224182.py:4: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be depreca
ted, use a list instead.
```



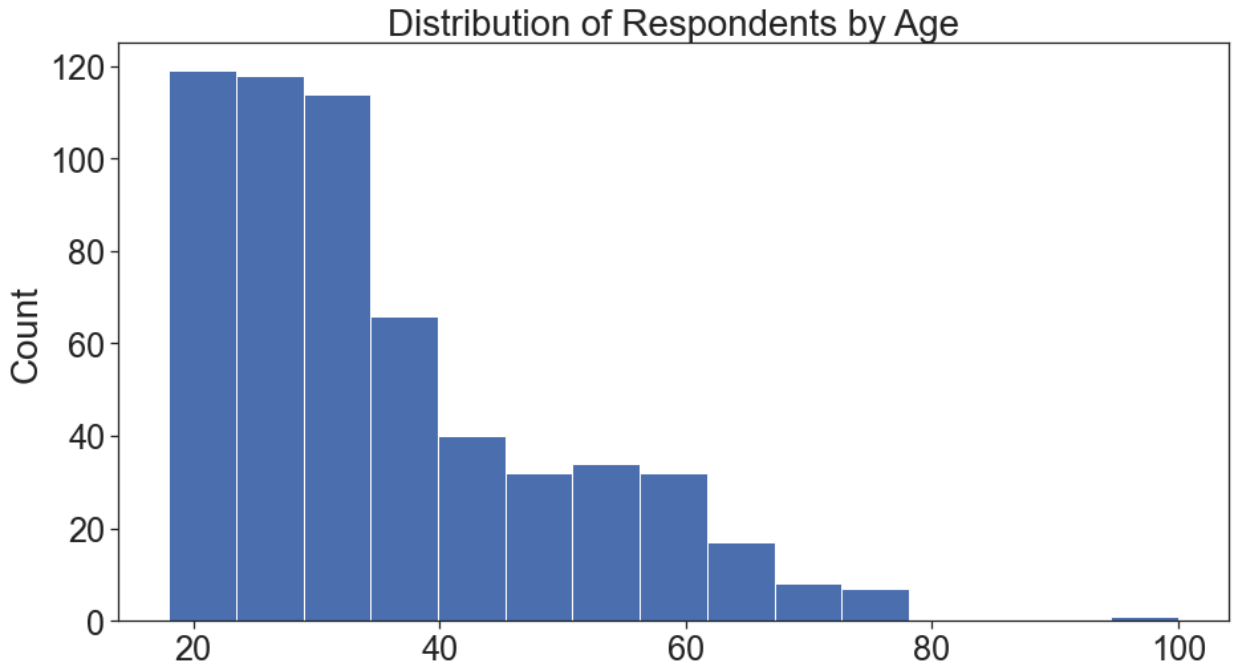
5. Respondents distribution

independent variables: 'age', 'income', 'political_party', 'education' 'RLK2022', 'federal_state'

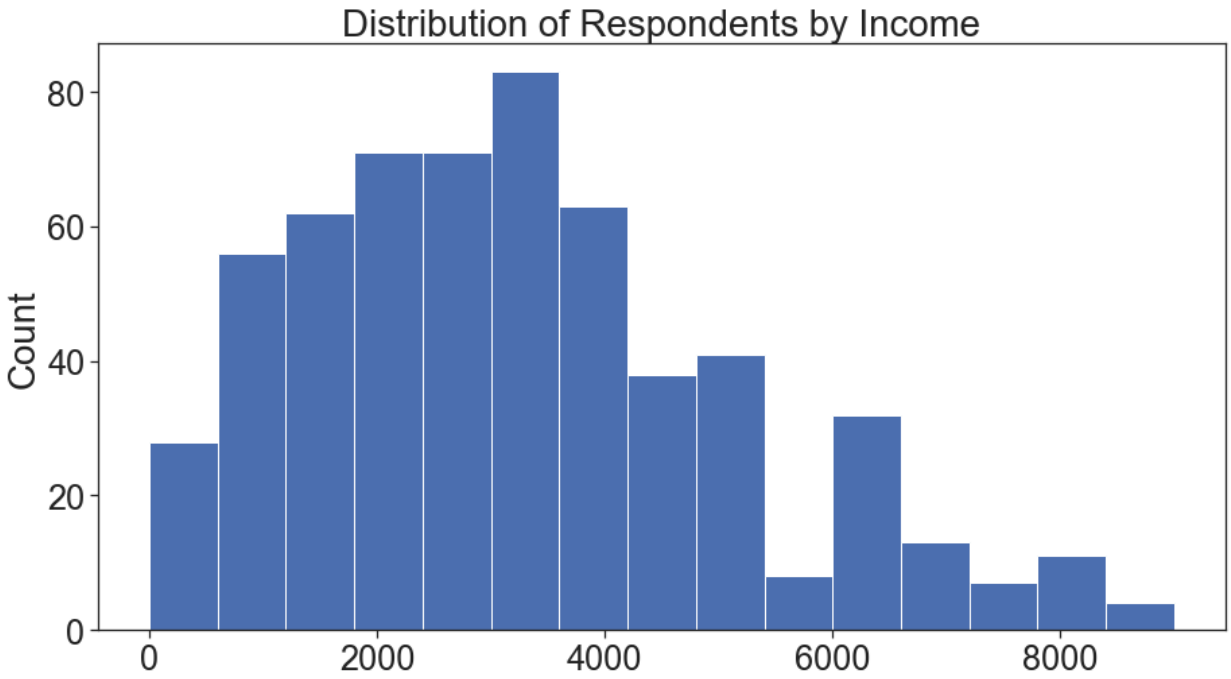
In [31]:
```
### Change column name: 'RLK2022' will be named as urban_rural_class as it is the vari

#df = df.rename(columns={'RLK2022':'urban_rural_class'})
```

In [32]:
```
# Setting the graph style
sns.set(font_scale = 2.3)
sns.set_style("ticks")
```

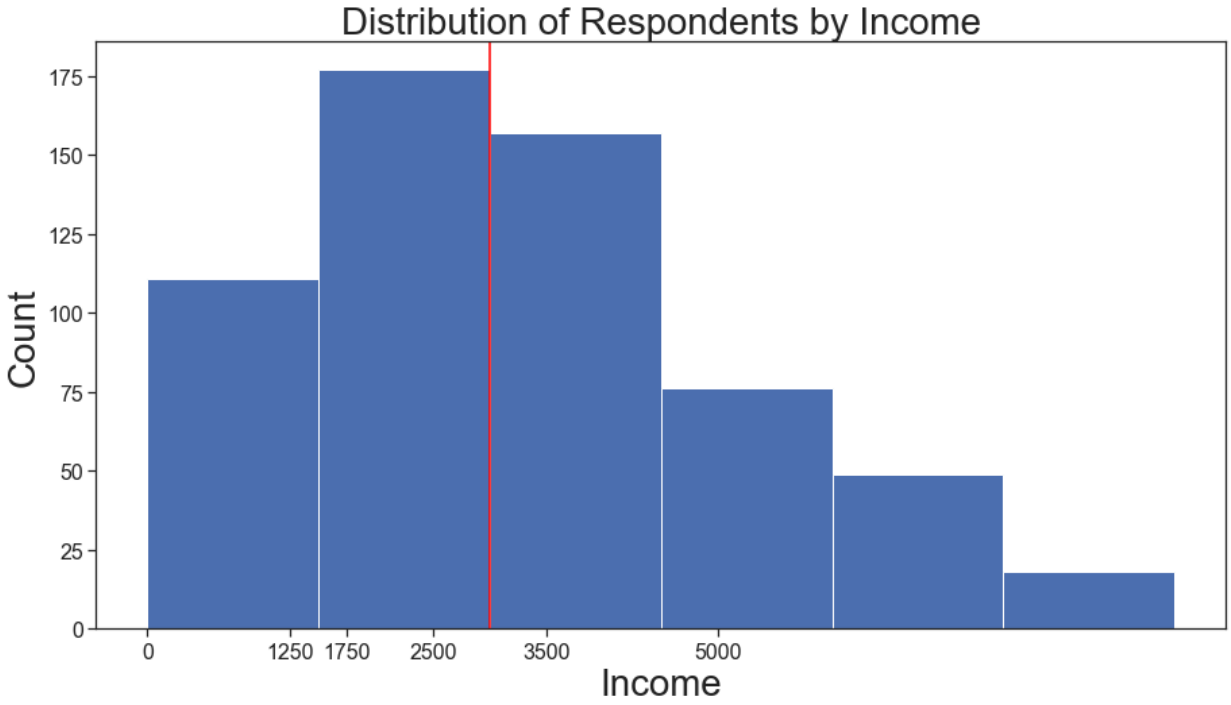
In [33]:
```
# histogram with age distribution

sns.set_style("ticks")

x = df['age']

plt.figure(figsize=(15,8))
plt.hist(x, bins = 15)

#plt.xlabel('Age')
plt.ylabel('Count')

plt.title('Distribution of Respondents by Age')
#plt.tick_params(labelsize=16)

plt.show()
```



In [34]:
```
# histogram with income distribution

sns.set_style("ticks")

x = df['income']

plt.figure(figsize=(15,8))
plt.hist(x, bins = 15)

#plt.xlabel('Income')
plt.ylabel('Count')

plt.title('Distribution of Respondents by Income')
#plt.tick_params(labelsize=16)



plt.show()
```



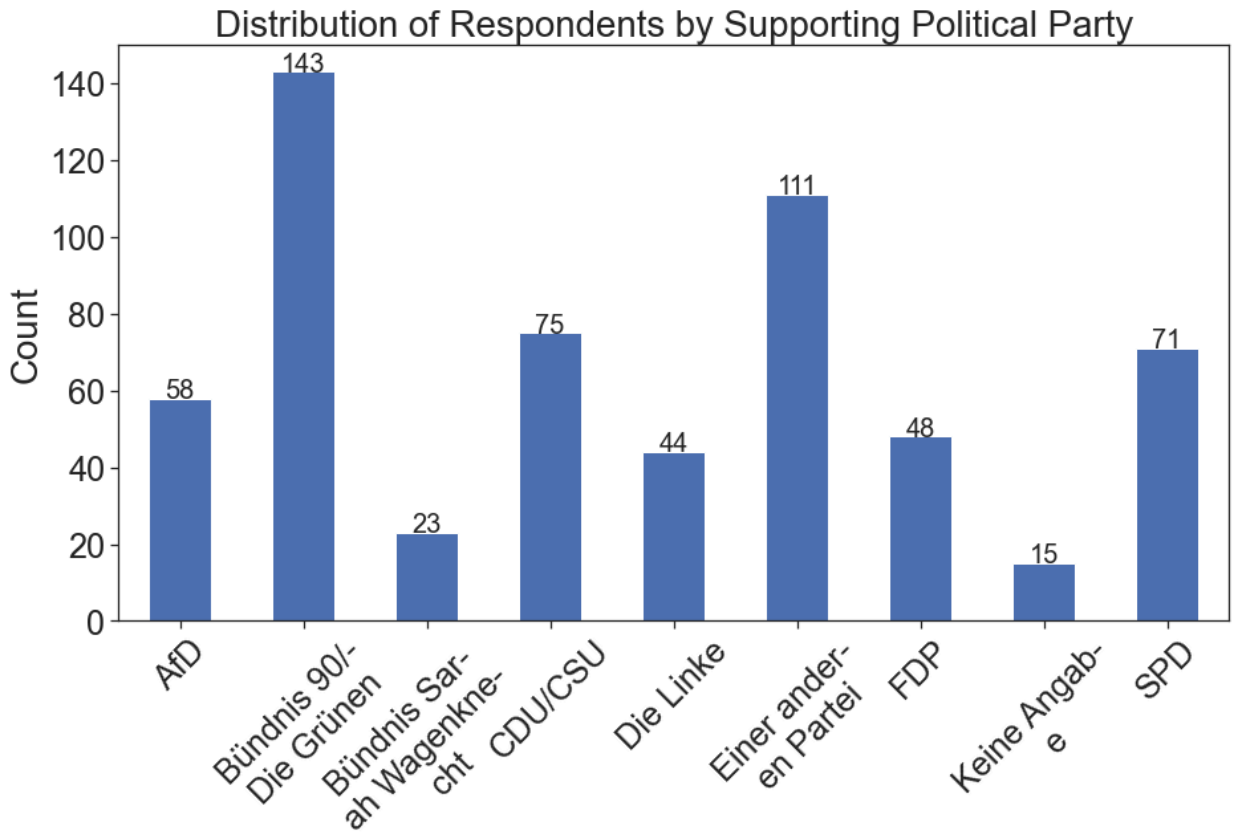
In [35]:
```
# histogram with income distribution - compare with the statistics from Federal Stat.

sns.set_style("ticks")

x = df['income']

plt.figure(figsize=(15,8))
plt.hist(x, bins = 6)

plt.xticks([0, 1250, 1750, 2500, 3500, 5000])

plt.xlabel('Income')
plt.ylabel('Count')

plt.axvline(x.quantile(0.5), color = 'red')

plt.title('Distribution of Respondents by Income')
plt.tick_params(labelsize=16)

plt.show()
```



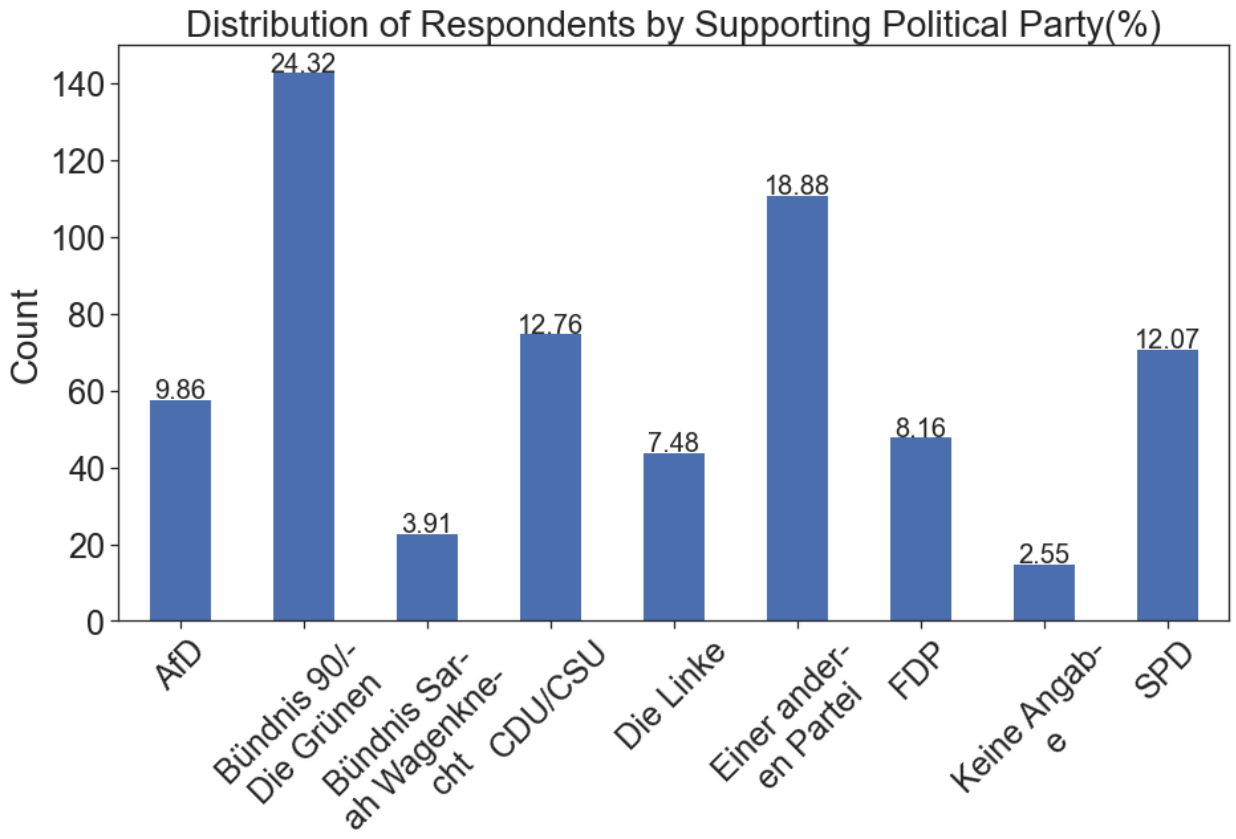
In [56]:
```
df['income'].quantile(0.5)
```

Out[56]:
```
3000.0
```

In [36]:
```
# Bar chart for political party

count = df.groupby(['political_party']).size()
count

ax = count.plot(kind='bar', figsize=(15, 8), ylabel='Count', rot=45, title = 'Distribu

start, end = ax.get_ylim()
ax.yaxis.set_ticks(np.arange(0, round(end + 0.5), 20))

for p in ax.patches:
    ax.annotate(p.get_height(), fontsize =20,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+ 2),
                ha='center',
                va='center')

max_chars = 11

new_labels = ['-\n'.join(label._text[i:i + max_chars]
                         for i in range(0, len(label._text), max_chars ))
              for label in ax.get_xticklabels()]

ax.set_xticklabels(new_labels)
ax.set(xlabel=None);

#plt.tick_params(labelsize=16)
```



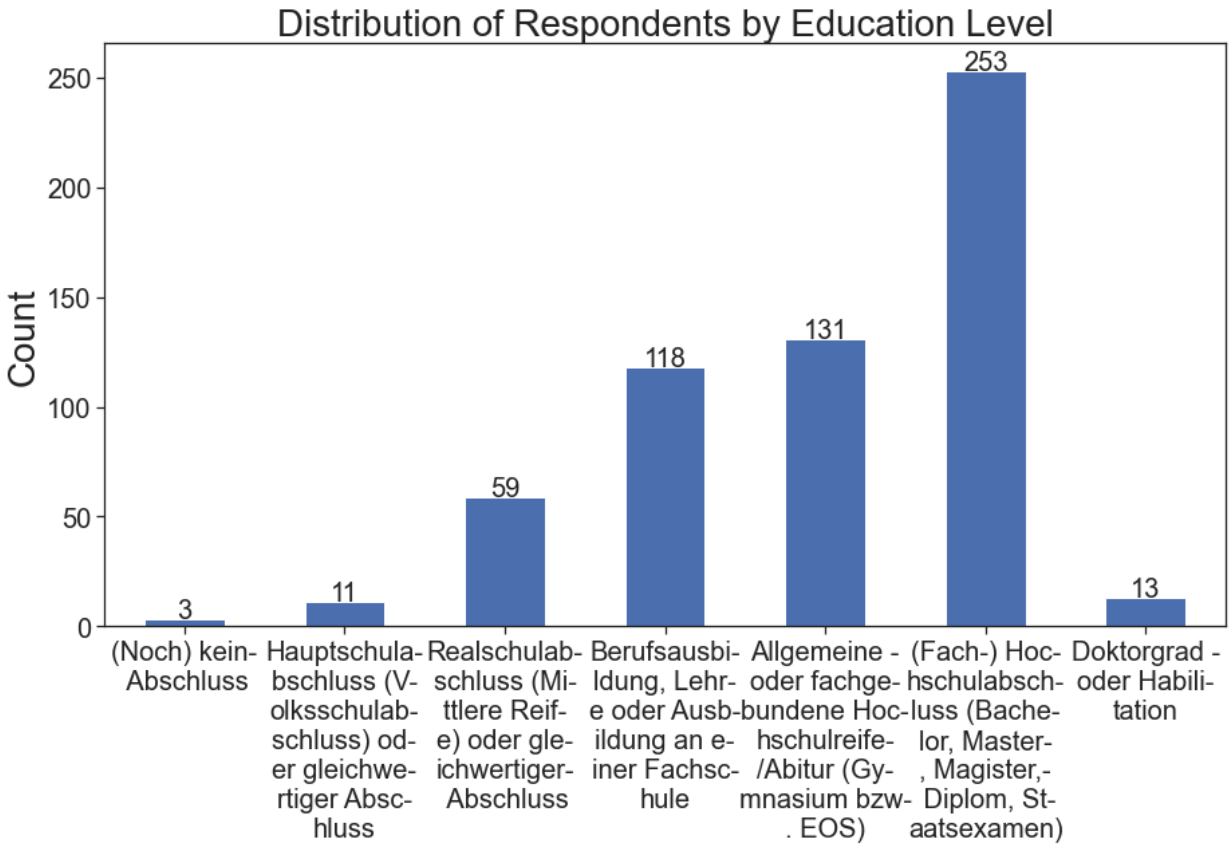
In [37]:
```
# Bar chart for political party

count = df.groupby(['political_party']).size()
count

ax = count.plot(kind='bar', figsize=(15, 8), ylabel='Count', rot=45, title = 'Distribu

start, end = ax.get_ylim()
ax.yaxis.set_ticks(np.arange(0, round(end + 0.5), 20))

for p in ax.patches:
    ax.annotate(round(100*(p.get_height()/588), 2), fontsize =20,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+ 2),
                ha='center',
                va='center')

max_chars = 11

new_labels = ['-\n'.join(label._text[i:i + max_chars]
                         for i in range(0, len(label._text), max_chars ))
              for label in ax.get_xticklabels()]

ax.set_xticklabels(new_labels)
ax.set(xlabel=None);

#plt.tick_params(labelsize=16)
```



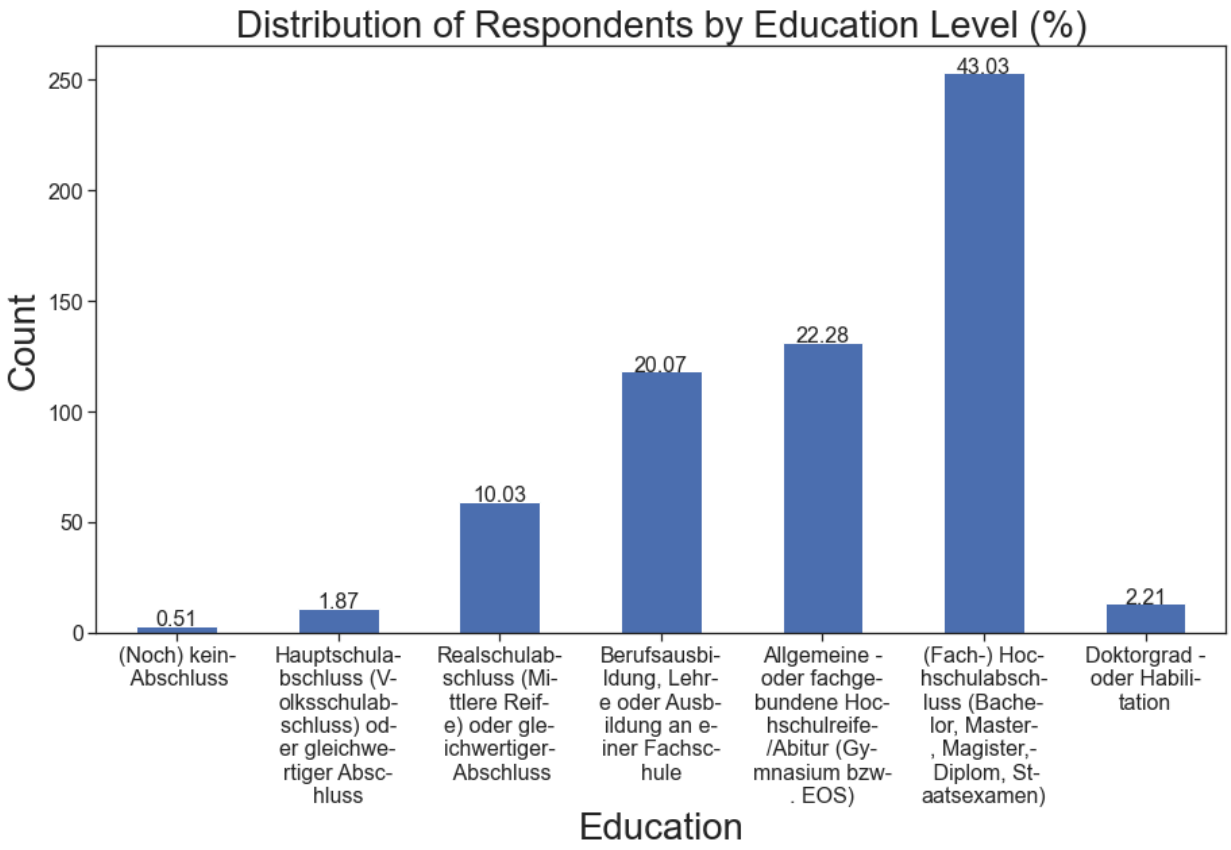
In [60]:
```
print(111/588)
```

```
0.18877551020408162
```

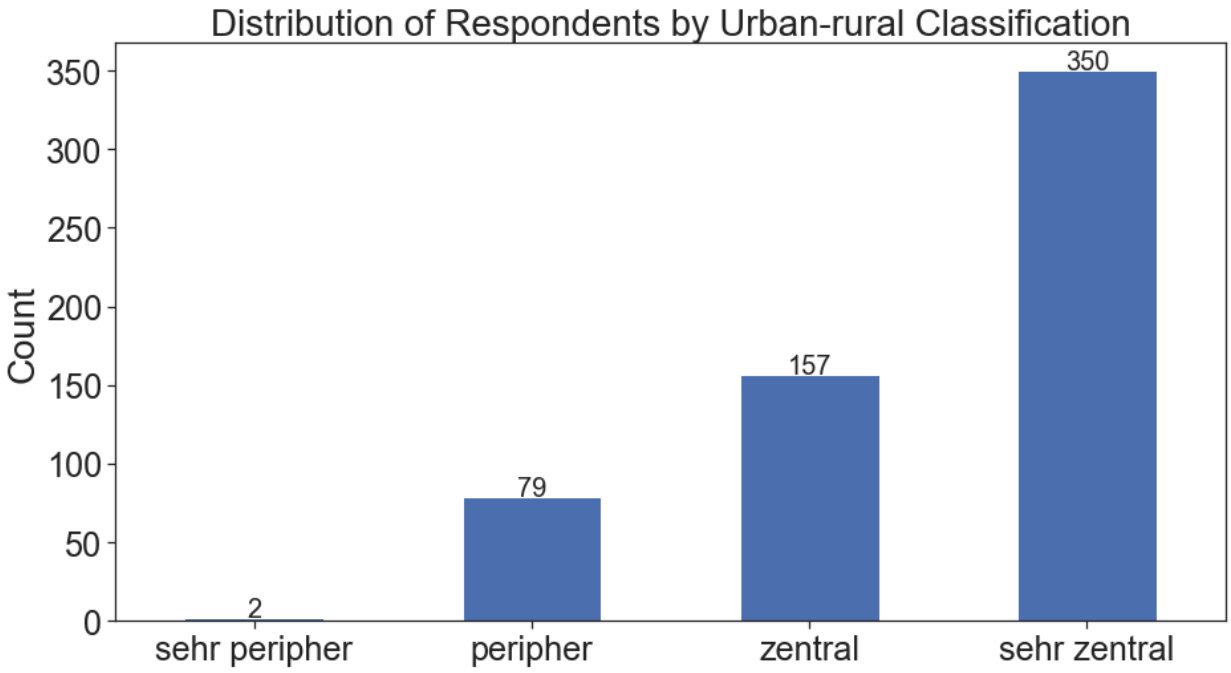
In [97]:
```
# Bar chart for political party

count = df.groupby(['education']).size().to_frame().reset_index()

x_order = ['(Noch) kein Abschluss', 'Hauptschulabschluss (Volksschulabschluss) oder gl
           'Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss',
           'Berufsausbildung, Lehre oder Ausbildung an einer Fachschule',
           'Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. EOS)',
           '(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexame
           'Doktorgrad oder Habilitation']

ax = count.set_index('education').loc[x_order].plot(kind='bar', figsize=(15, 8), ylabe
ax.get_legend().remove()

for p in ax.patches:
    ax.annotate(p.get_height(),fontsize = 20,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+4),
                ha='center',
                va='center')
max_chars = 11

new_labels = ['-\n'.join(label._text[i:i + max_chars]
                         for i in range(0, len(label._text), max_chars ))
              for label in ax.get_xticklabels()]

ax.set_xticklabels(new_labels)
plt.tick_params(labelsize=20)
ax.set(xlabel=None);
```



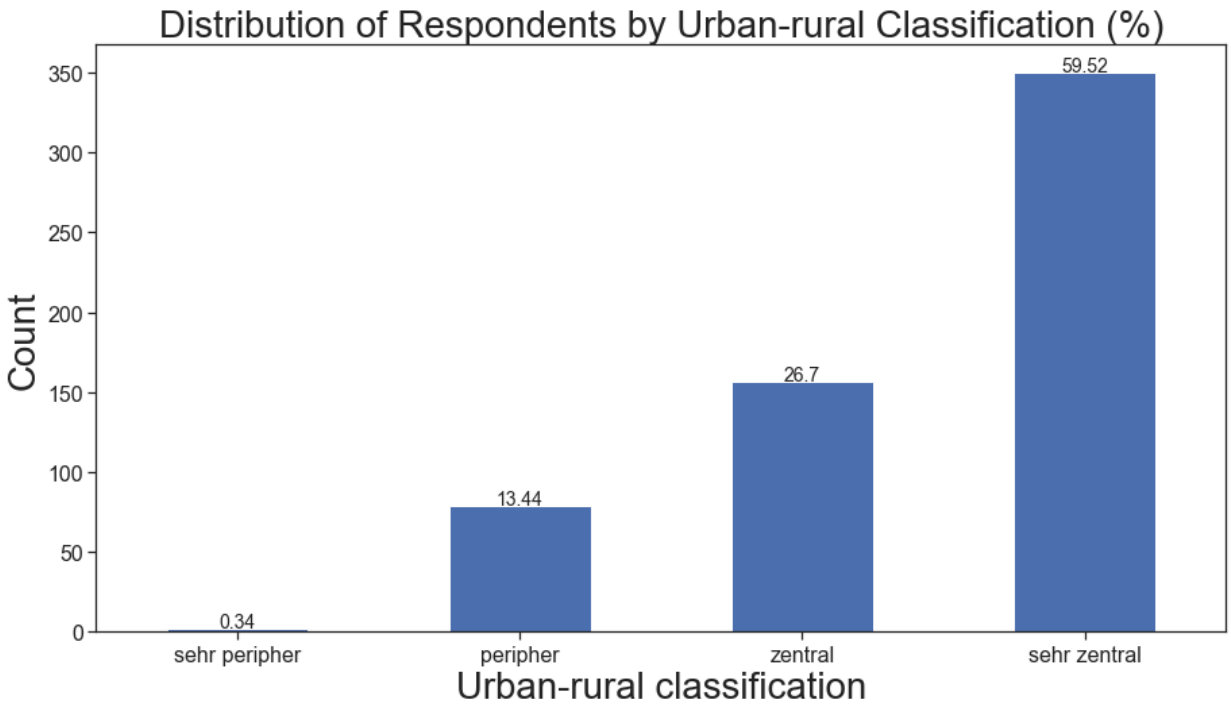
In [82]:
```
# Bar chart for political party

count = df.groupby(['education']).size().to_frame().reset_index()

x_order = ['(Noch) kein Abschluss', 'Hauptschulabschluss (Volksschulabschluss) oder gl
           'Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss',
           'Berufsausbildung, Lehre oder Ausbildung an einer Fachschule',
           'Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. EOS)',
           '(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexame
           'Doktorgrad oder Habilitation']

ax = count.set_index('education').loc[x_order].plot(kind='bar', figsize=(15, 8), xlabe
ax.get_legend().remove()

for p in ax.patches:
    ax.annotate(round(100*(p.get_height()/588), 2),fontsize = 16,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+3),
                ha='center',
                va='center')
max_chars = 11

new_labels = ['-\n'.join(label._text[i:i + max_chars]
                         for i in range(0, len(label._text), max_chars ))
              for label in ax.get_xticklabels()]

ax.set_xticklabels(new_labels)

plt.tick_params(labelsize=16)
```



In [98]:
```
# Bar chart for urban-rural classification

count = df.groupby(['urban_rural_class']).size().to_frame().reset_index()

x_order = ['sehr peripher', 'peripher','zentral','sehr zentral']

ax = count.set_index('urban_rural_class').loc[x_order].plot(kind='bar', figsize=(15, 8

ax.get_legend().remove()

for p in ax.patches:
    ax.annotate(p.get_height(), fontsize = 20,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+5.1),
                ha='center',
                va='center')
ax.set(xlabel=None);
#plt.tick_params(labelsize=16)
```

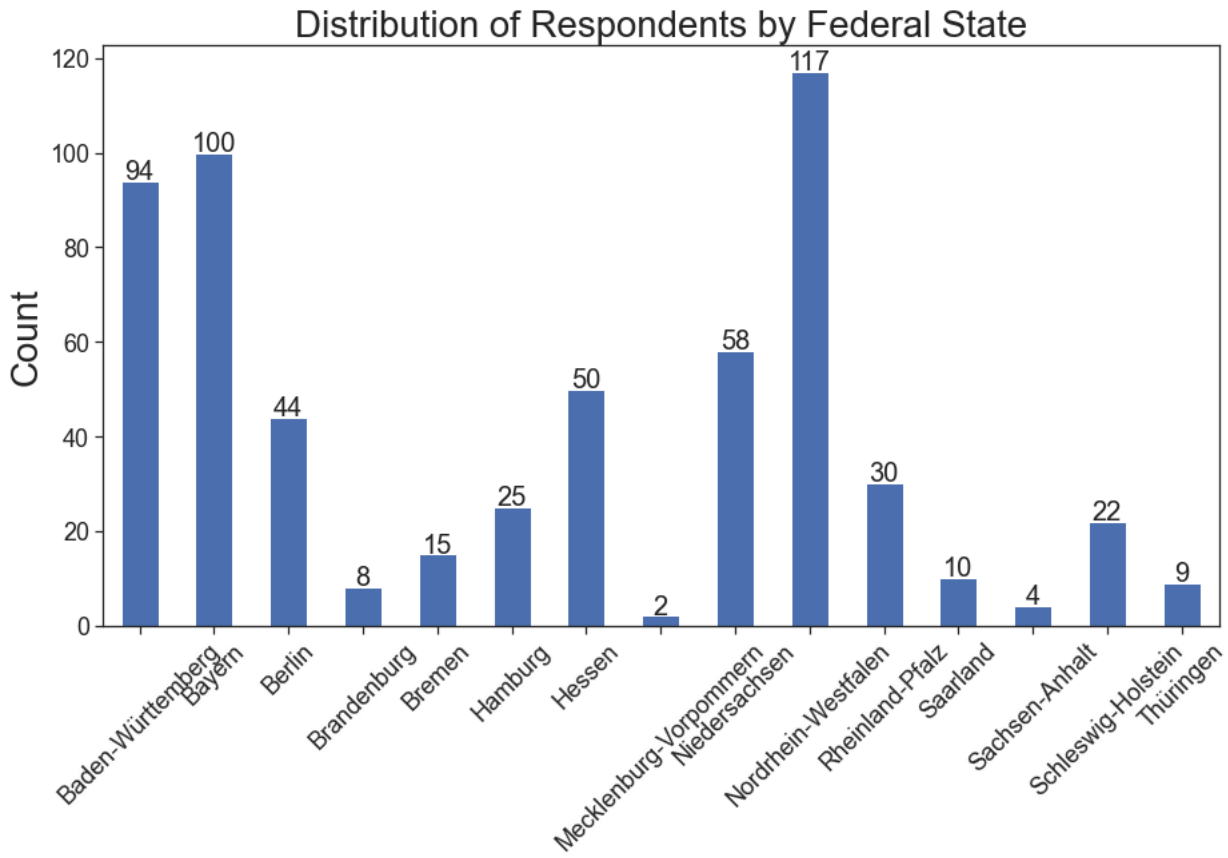


In [84]:
```
# Bar chart for urban-rural classification

count = df.groupby(['urban_rural_class']).size().to_frame().reset_index()

x_order = ['sehr peripher', 'peripher','zentral','sehr zentral']

ax = count.set_index('urban_rural_class').loc[x_order].plot(kind='bar', figsize=(15, 8

ax.get_legend().remove()

for p in ax.patches:
    ax.annotate(round(100*(p.get_height()/588), 2),fontsize=14,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+4),
                ha='center',
                va='center')

plt.tick_params(labelsize=16)
```

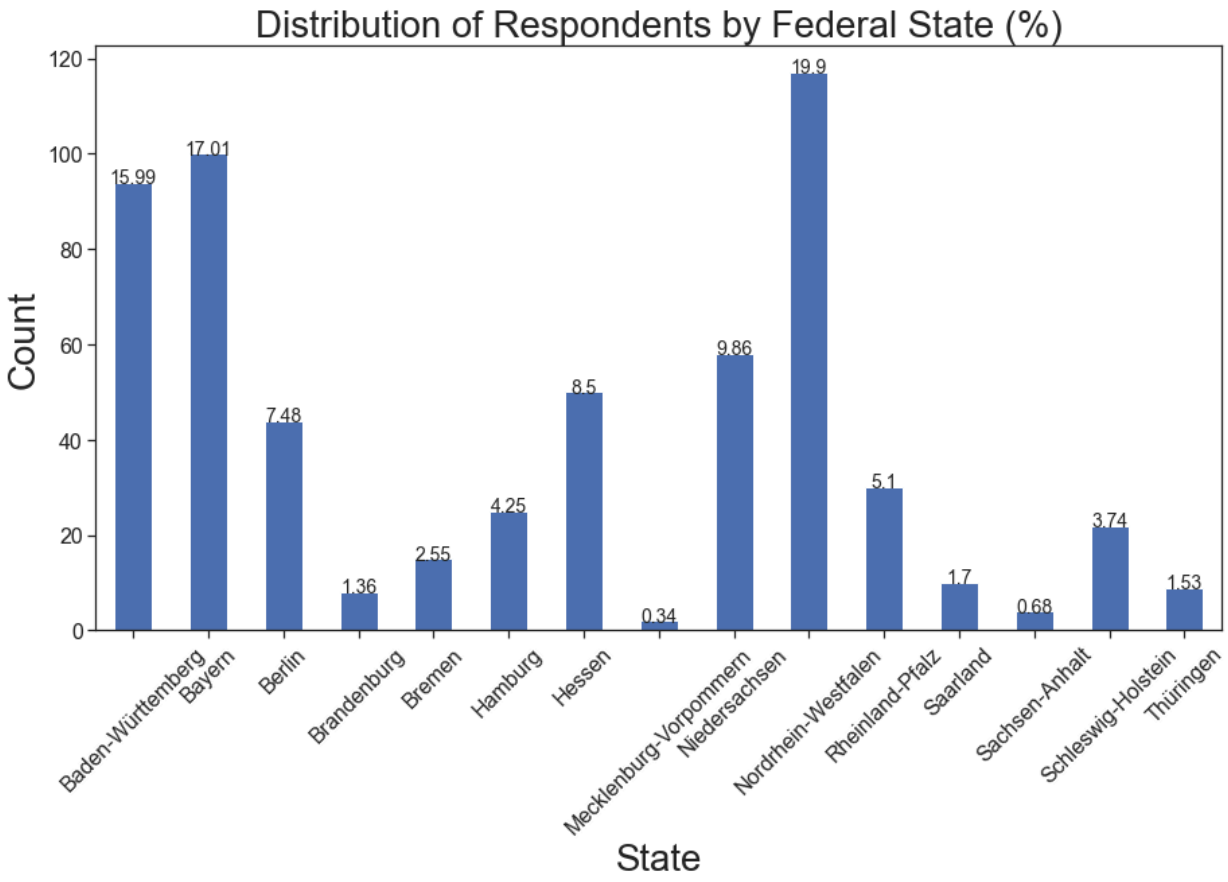


In [101…

```
# Bar chart for federal states

count = df.groupby(['federal_state']).size()
count

ax = count.plot(kind='bar', figsize=(15, 8), xlabel='State', ylabel='Count', rot=45, t

start, end = ax.get_ylim()
ax.yaxis.set_ticks(np.arange(0, round(end + 0.5), 20))

for p in ax.patches:
    ax.annotate(p.get_height(),fontsize=20,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+ 2),
                ha='center',
                va='center')


plt.tick_params(labelsize=18)
ax.set(xlabel=None);
```



In [89]:
```
count = df.groupby(['federal_state']).size()
count

ax = count.plot(kind='bar', figsize=(15, 8), xlabel='State', ylabel='Count', rot=45, t

start, end = ax.get_ylim()
ax.yaxis.set_ticks(np.arange(0, round(end + 0.5), 20))

for p in ax.patches:
    ax.annotate(round(100*(p.get_height()/588), 2),fontsize=14,
                xy=(p.get_x()+p.get_width()/2., p.get_height()+ 1),
                ha='center',
                va='center')


plt.tick_params(labelsize=16)
```



6. Selective variables EDA

- independent variables: 'age', 'income', 'political_party', 'education' 'RLK2022', 'federal_state'
- dependent variable 1: 'CO2_housing_electricity', 'CO2_mobility', 'CO2_food', 'CO2_other_consumption', 'CO2_total'
- dependent variable 2: 'belief_diff_housing_electricity', 'belief_diff_mobility', 'belief_diff_belief_food', 'belief_diff_otherconsumption', 'belief_diff_total'

dependent variable 1: 'CO2_housing_electricity', 'CO2_mobility', 'CO2_food', 'CO2_otherconsumption', 'CO2_total'

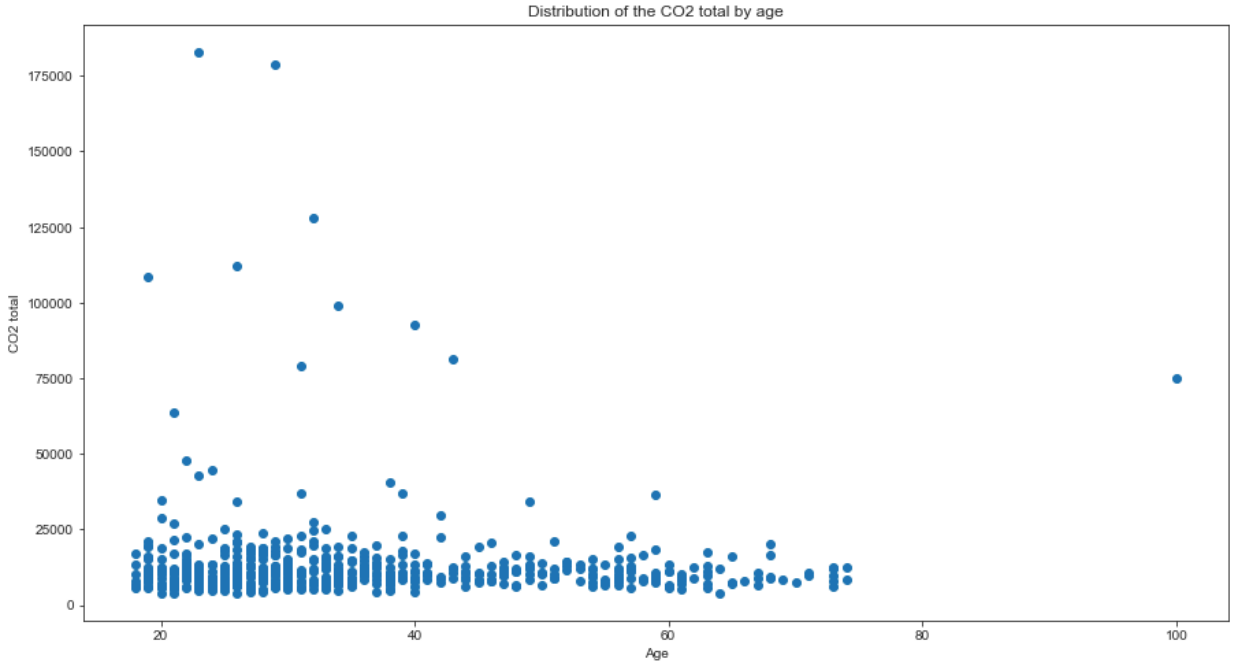
In [32]:
```
## functions to create distribution plots for the chosen dependent variables

def dist_age(dependent_var):
    sns.set_style("ticks")

    x = df['age']
    y = df[dependent_var]

    name_dependent_var = dependent_var.replace("_", " ")

    plt.figure(figsize=(15,8))
    plt.scatter(x, y)

    plt.xlabel('Age')

    plt.ylabel(name_dependent_var)
```

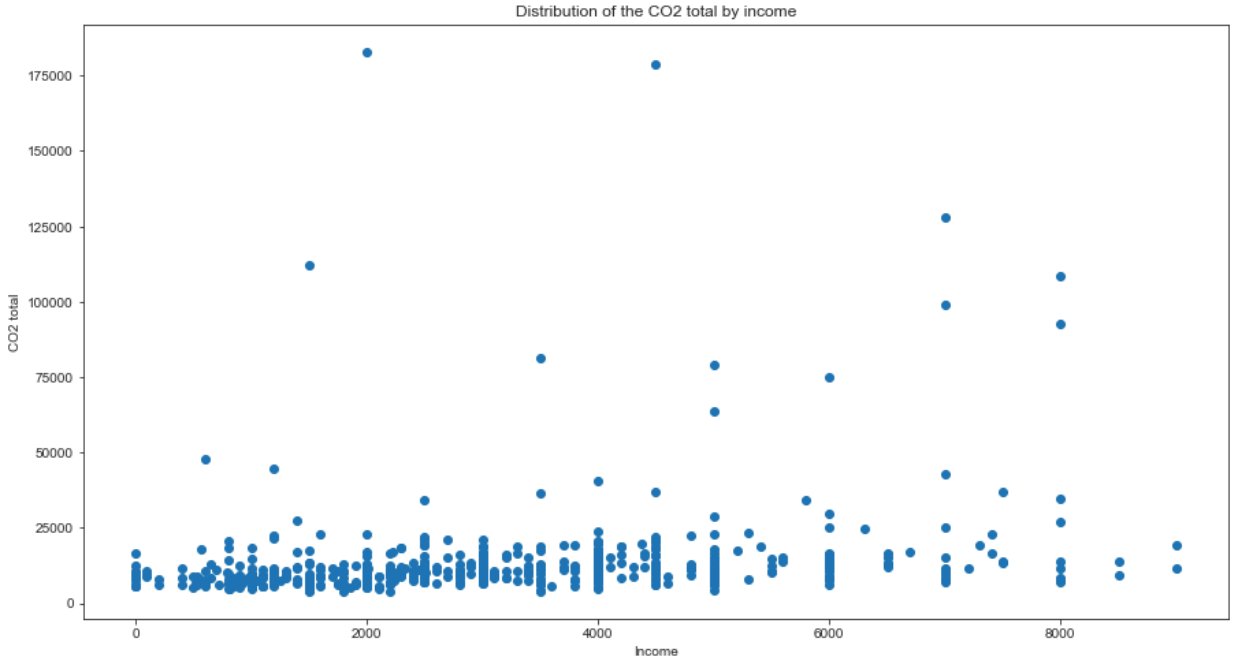
```
    plt.title('Distribution of the ' + name_dependent_var + ' by age')

    plt.show()

def dist_income(dependent_var):
    sns.set_style("ticks")

    x = df['income']
    y = df[dependent_var]

    name_dependent_var = dependent_var.replace("_", " ")

    plt.figure(figsize=(15,8))
    plt.scatter(x, y)

    plt.xlabel('Income')
    plt.ylabel(name_dependent_var)

    plt.title('Distribution of the ' + name_dependent_var +' by income')

    plt.show()

def dist_political(dependent_var):
    sns.set_style("ticks")

    name_dependent_var = dependent_var.replace("_", " ")

    sns.set_style("ticks")

    plt.figure(figsize=(15,8))
    sns.boxplot(x=df['political_party'], y=df[dependent_var])

    plt.xlabel('Political party')
    plt.ylabel(name_dependent_var)

    plt.title('Distribution of the ' + name_dependent_var +' by supporting political p

    plt.show()

def dist_education(dependent_var):
    sns.set_style("ticks")

    x_order = ['(Noch) kein Abschluss', 'Hauptschulabschluss (Volksschulabschluss) ode
               'Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss',
               'Berufsausbildung, Lehre oder Ausbildung an einer Fachschule',
               'Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. EOS
               '(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatse
               'Doktorgrad oder Habilitation']

    name_dependent_var = dependent_var.replace("_", " ")

    fig = plt.figure(figsize=(15, 8))
    ax = fig.add_subplot(111)

    lines  = sns.boxplot(x=df['education'], y=df[dependent_var], order = x_order).set(
    plt.title('Distribution of the ' + name_dependent_var + ' by education level')

    x_labels = ax.get_xticklabels()
```

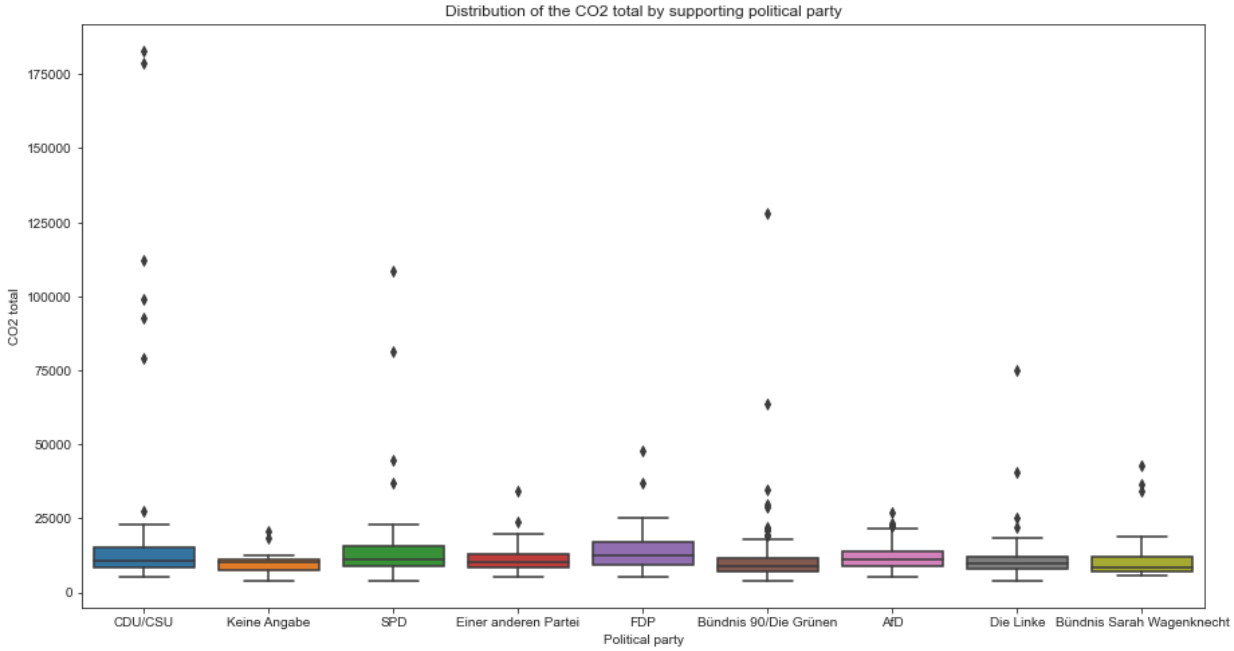
```
    max_chars = 17
    new_labels = ['-\n'.join(label._text[i:i + max_chars] for i in range(0, len(label.
    ax.set_xticklabels(new_labels)

    plt.show()

def dist_urban_rural(dependent_var):

    sns.set_style("ticks")

    x_order = ['sehr peripher', 'peripher','zentral','sehr zentral']

    name_dependent_var = dependent_var.replace("_", " ")

    fig = plt.figure(figsize=(15, 8))
    ax = fig.add_subplot(111)

    lines  = sns.boxplot(x=df['urban_rural_class'], y=df[dependent_var], order = x_orc
    plt.title('Distribution of the '+ name_dependent_var +' by urban-rural classificat

    plt.show()

def dist_federal_state(dependent_var):
    sns.set_style("ticks")

    fig = plt.figure(figsize=(20, 8))
    ax = fig.add_subplot(111)

    name_dependent_var = dependent_var.replace("_", " ")

    lines  = sns.boxplot(x=df['federal_state'], y=df[dependent_var]).set(xlabel = 'Fec
    plt.title('Distribution of the ' + name_dependent_var + ' by federal states')

    x_labels = ax.get_xticklabels()

    max_chars = 10
    new_labels = ['-\n'.join(label._text[i:i + max_chars] for i in range(0, len(label.
    ax.set_xticklabels(new_labels)

    plt.show()


def run_all_var(dependent_var):

    dist_age(dependent_var)
    dist_income(dependent_var)
    dist_political(dependent_var)
    dist_education(dependent_var)
    dist_urban_rural(dependent_var)
    dist_federal_state(dependent_var)
```

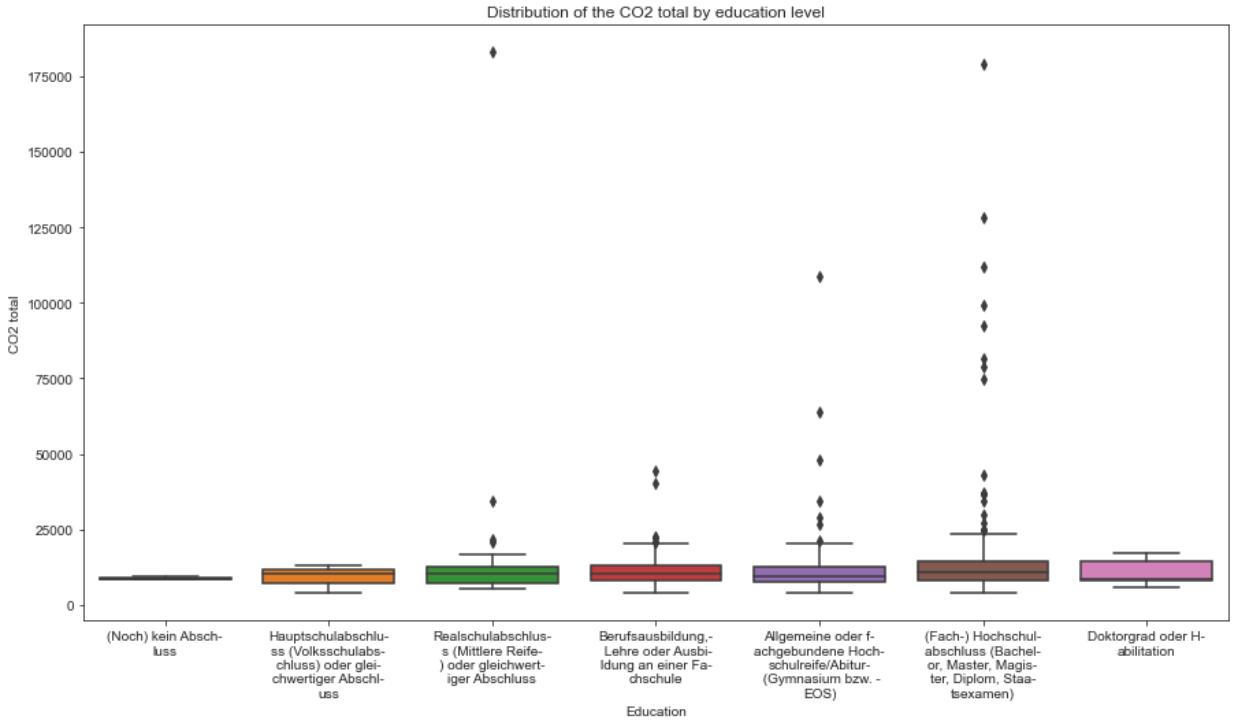
In [33]:

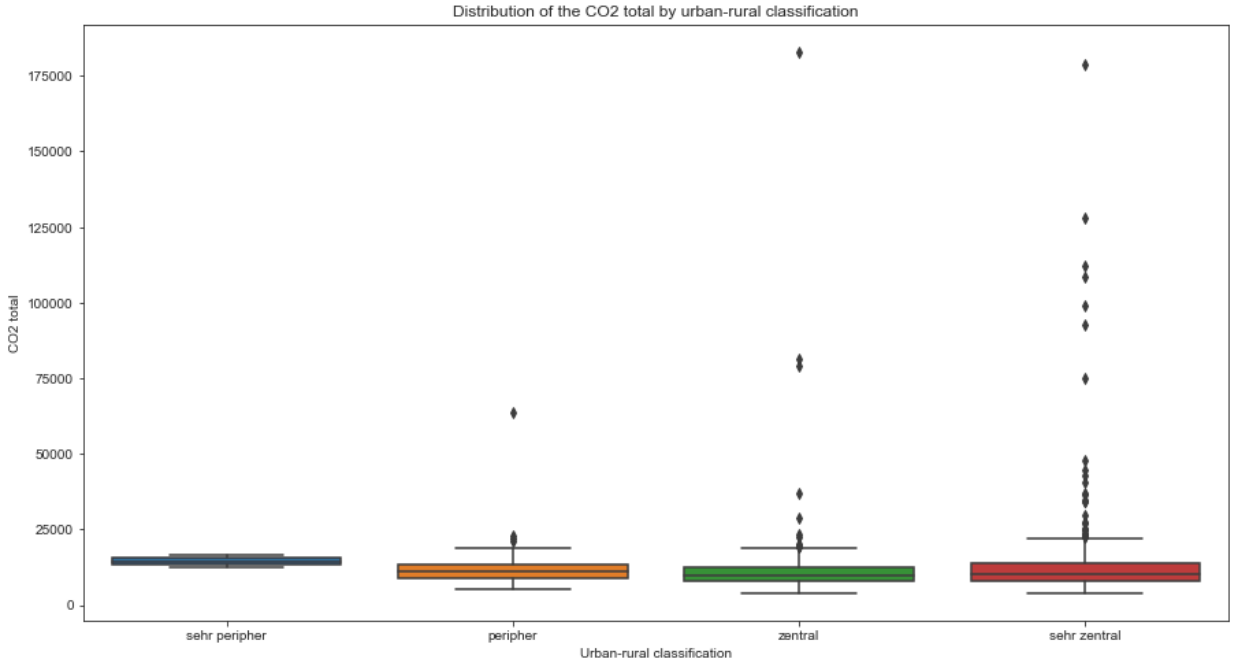
```
### Create the distribution plots for CO2 footprint variables

dependent_vars = ['CO2_total', 'CO2_housing_electricity', 'CO2_mobility', 'CO2_food',
# dependent variable 2: 'belief_diff_housing_electricity', 'belief_diff_mobility', 'be

for dependent_var in dependent_vars:
    run_all_var(dependent_var)
```

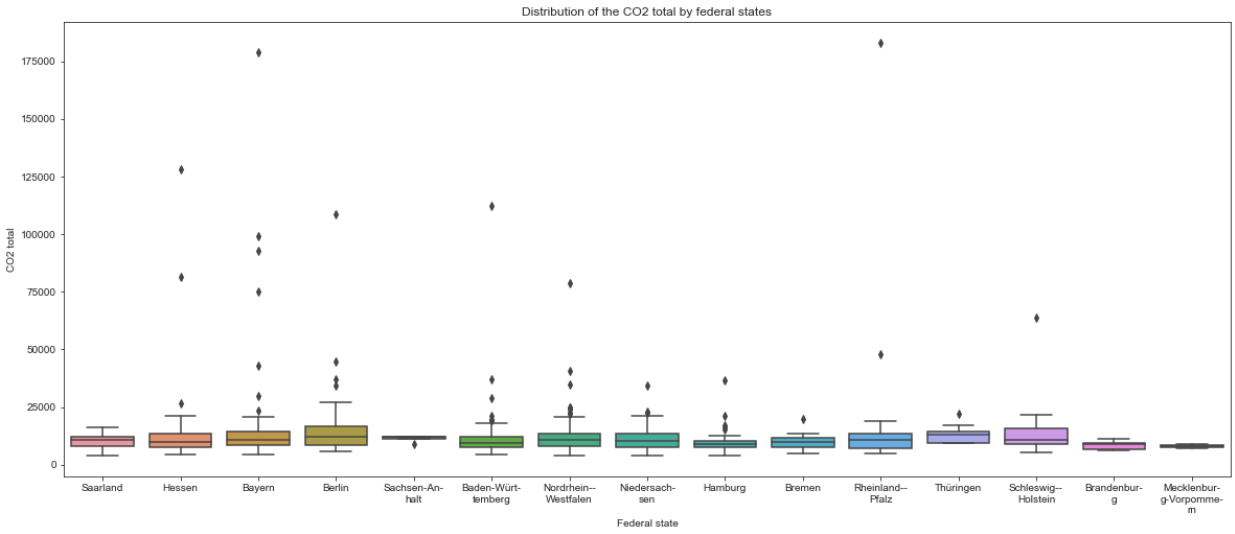

Distribution of the CO2 total by age
CO2 total
Age
0
25000
50000
75000
100000
125000
150000
175000
20
40
60
80
100

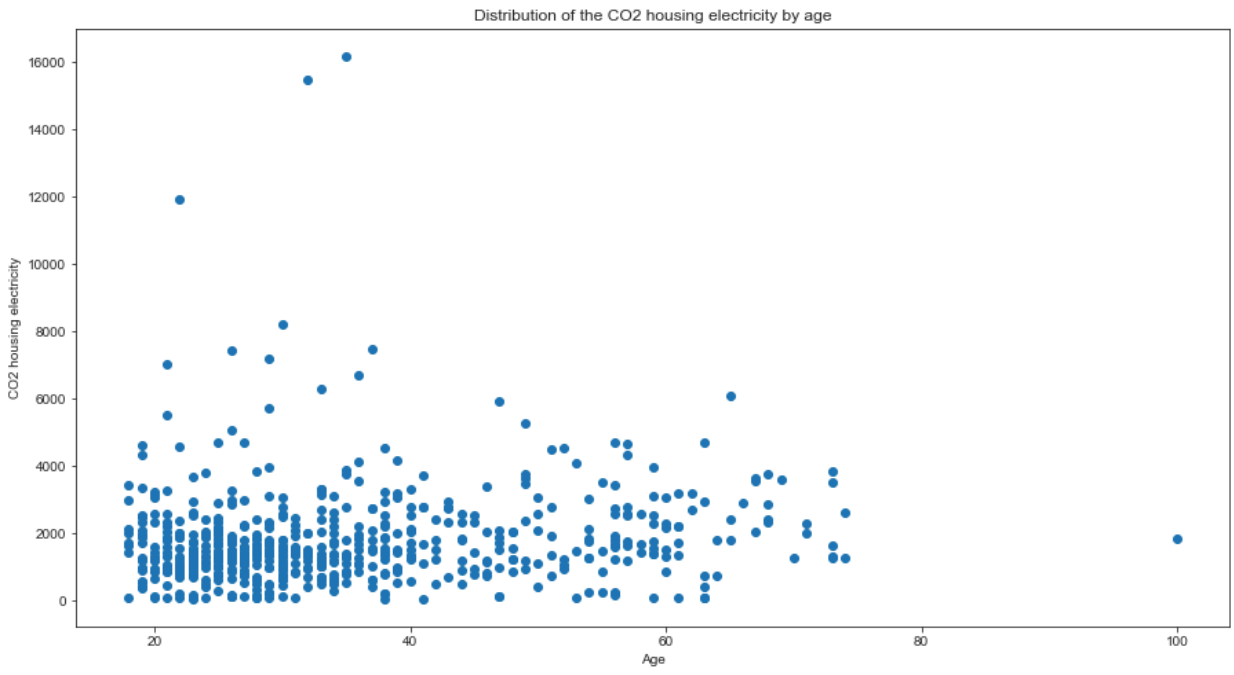

Distribution of the CO2 total by income
CO2 total
Income
0
25000
50000
75000
100000
125000
150000
175000
0
2000
4000
6000
8000

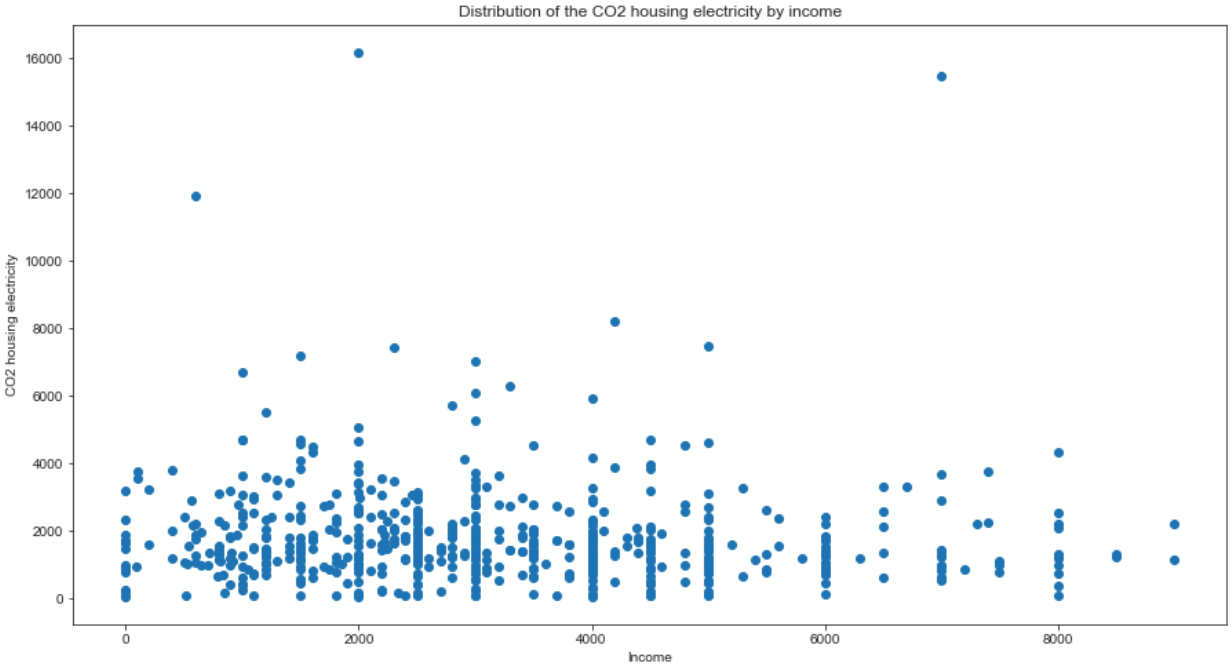

Distribution of the CO2 total by supporting political party
CO2 total
0
25000
50000
75000
100000
125000
150000
175000
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

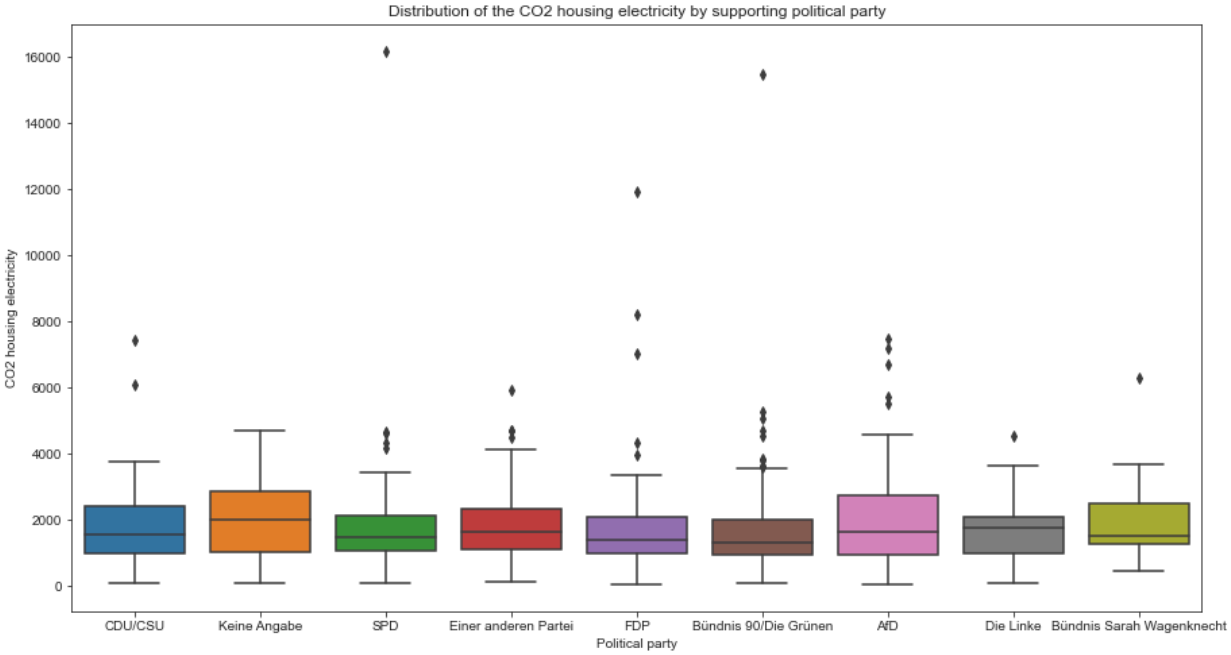

Distribution of the CO2 total by education level
CO2 total
0
25000
50000
75000
100000
125000
150000
175000
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

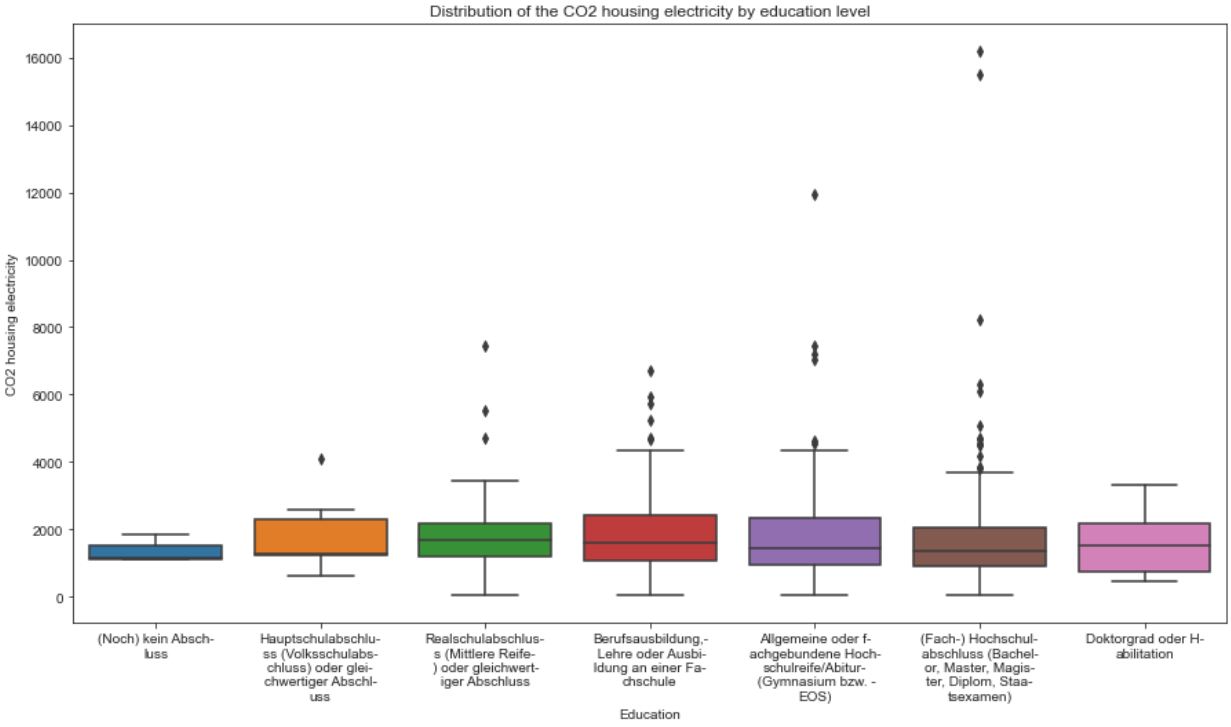

Distribution of the CO2 total by urban-rural classification
CO2 total
175000
150000
125000
100000
75000
50000
25000
0
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

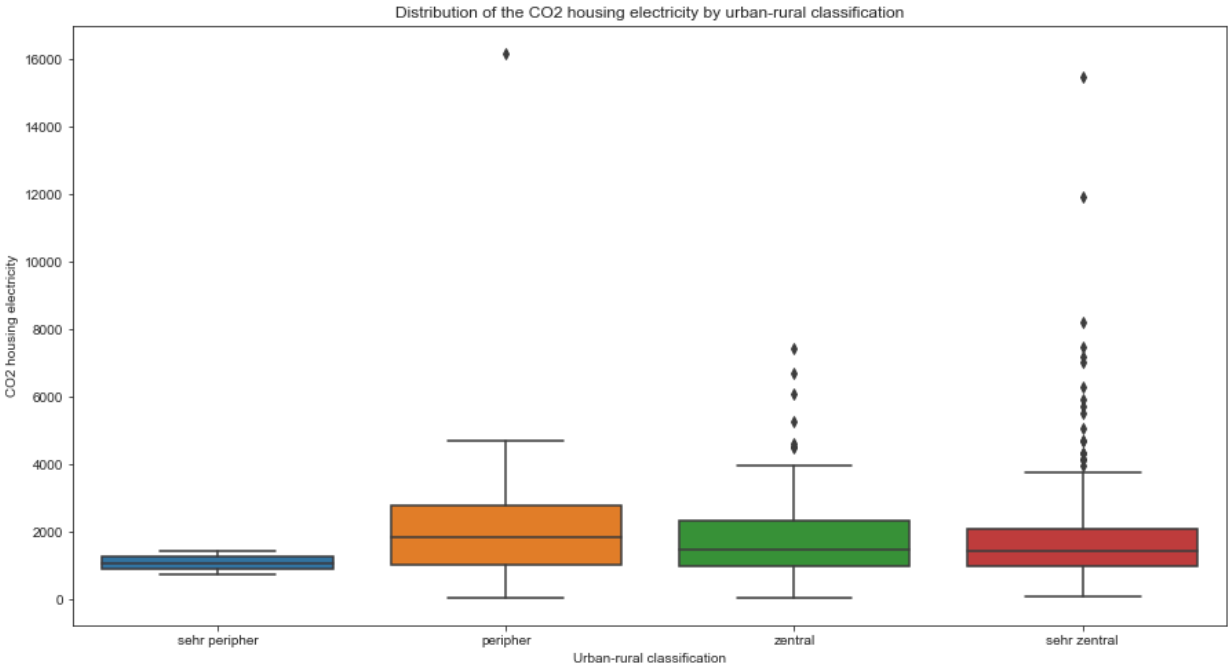

Distribution of the CO2 total by federal states
CO2 total
175000
150000
125000
100000
75000
50000
25000
0
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersach-sen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenbur-g
Mecklenbur-g-Vorpomme-m
Federal state

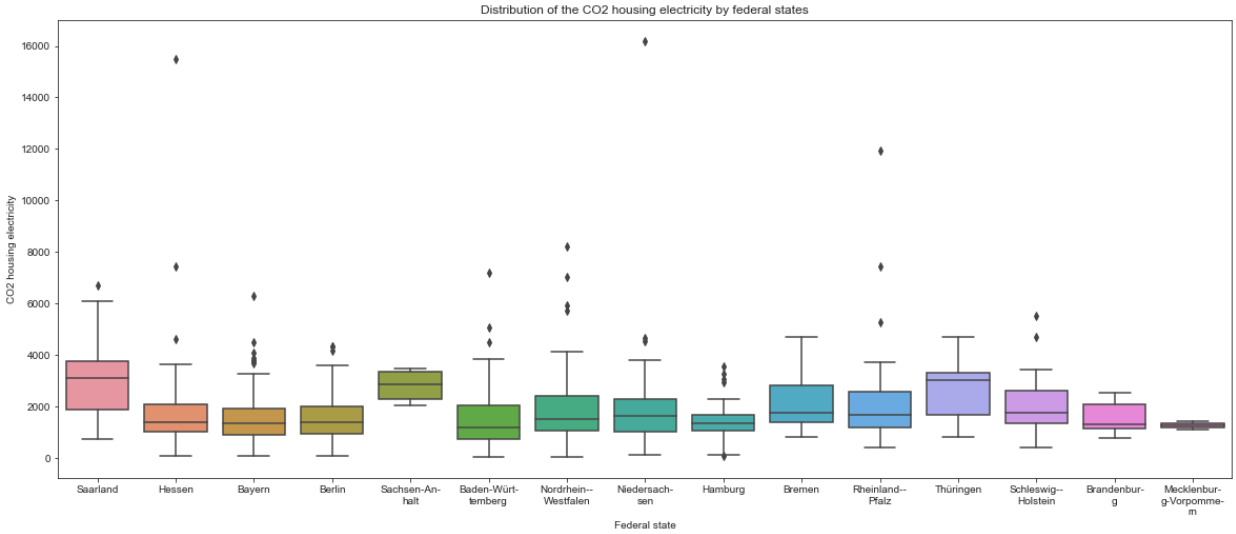

Distribution of the CO2 housing electricity by age
CO2 housing electricity
16000
14000
12000
10000
8000
6000
4000
2000
0
20
40
60
80
100
Age

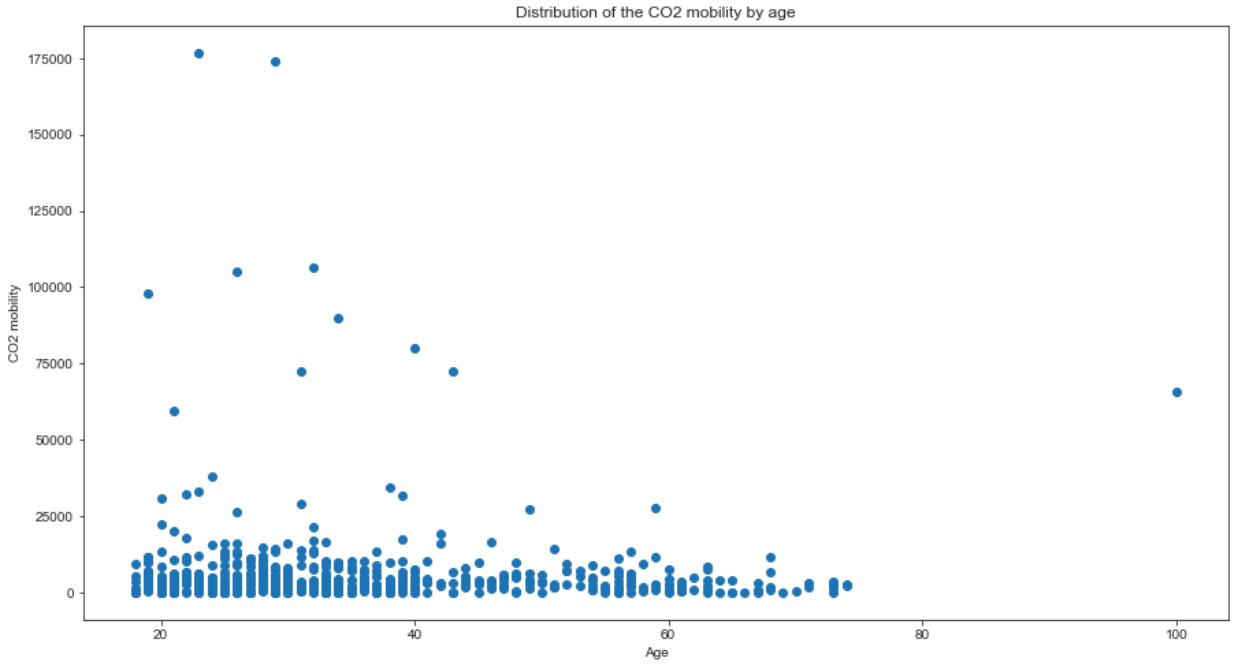

Distribution of the CO2 housing electricity by income
CO2 housing electricity
Income
0
2000
4000
6000
8000
10000
12000
14000
16000

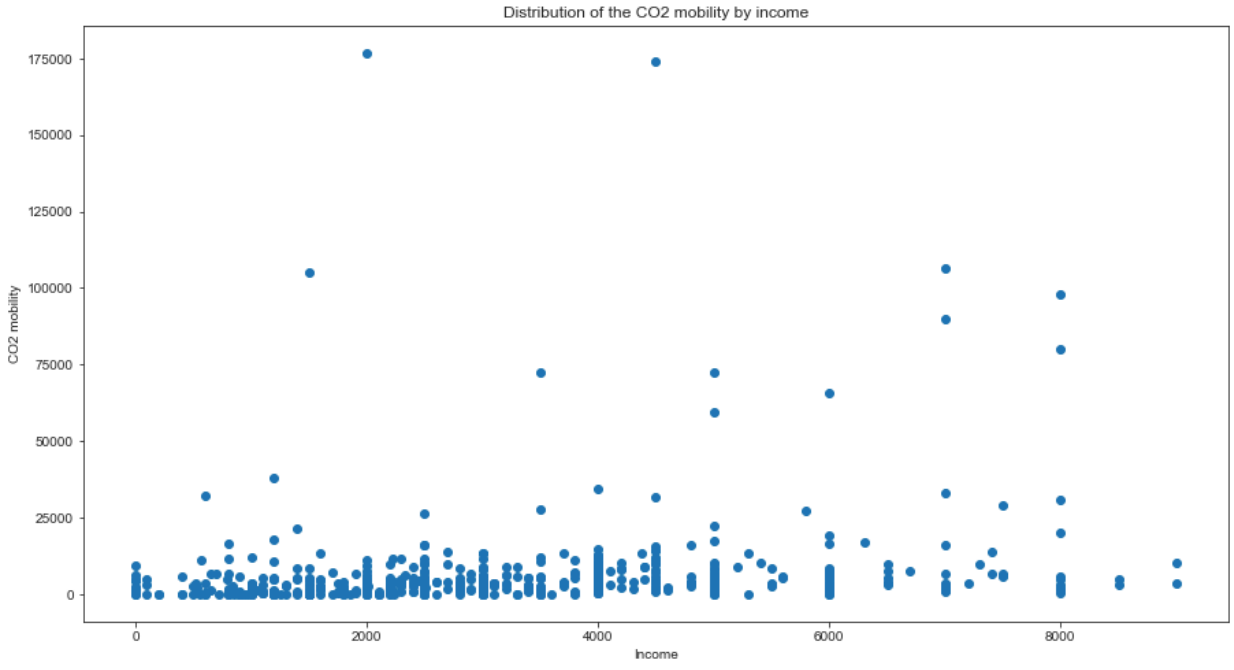

Distribution of the CO2 housing electricity by supporting political party
CO2 housing electricity
Political party
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht

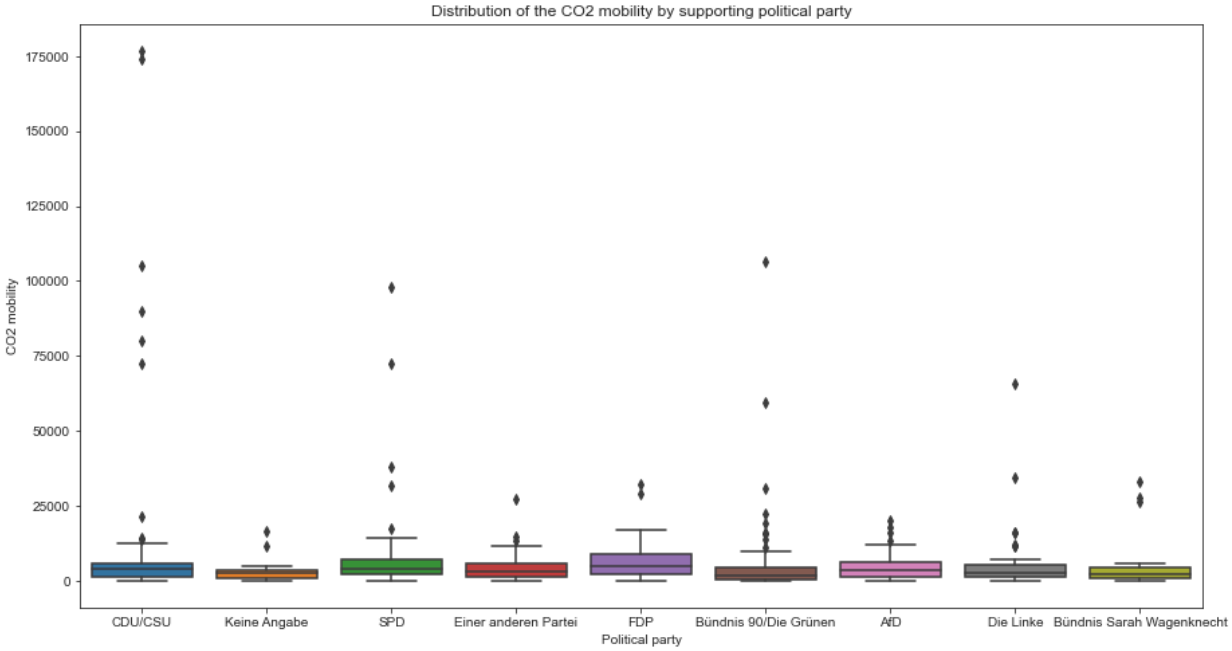

Distribution of the CO2 housing electricity by education level
CO2 housing electricity
0
2000
4000
6000
8000
10000
12000
14000
16000
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

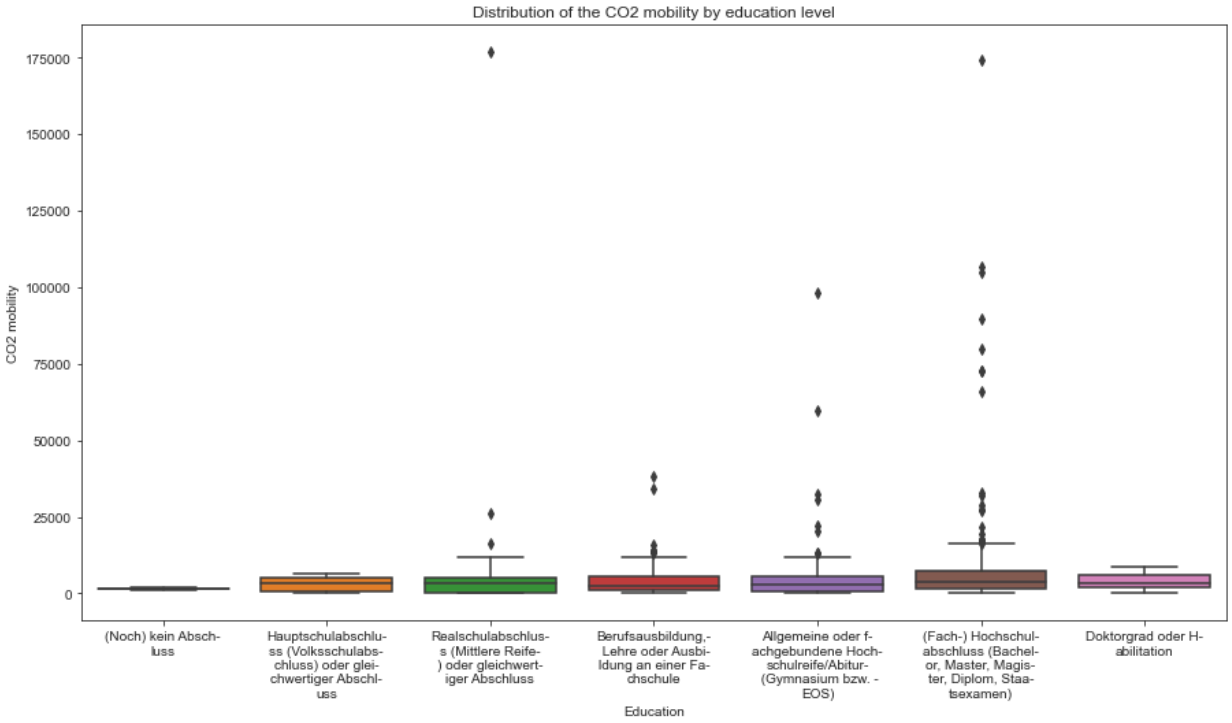

Distribution of the CO2 housing electricity by urban-rural classification
CO2 housing electricity
0
2000
4000
6000
8000
10000
12000
14000
16000
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

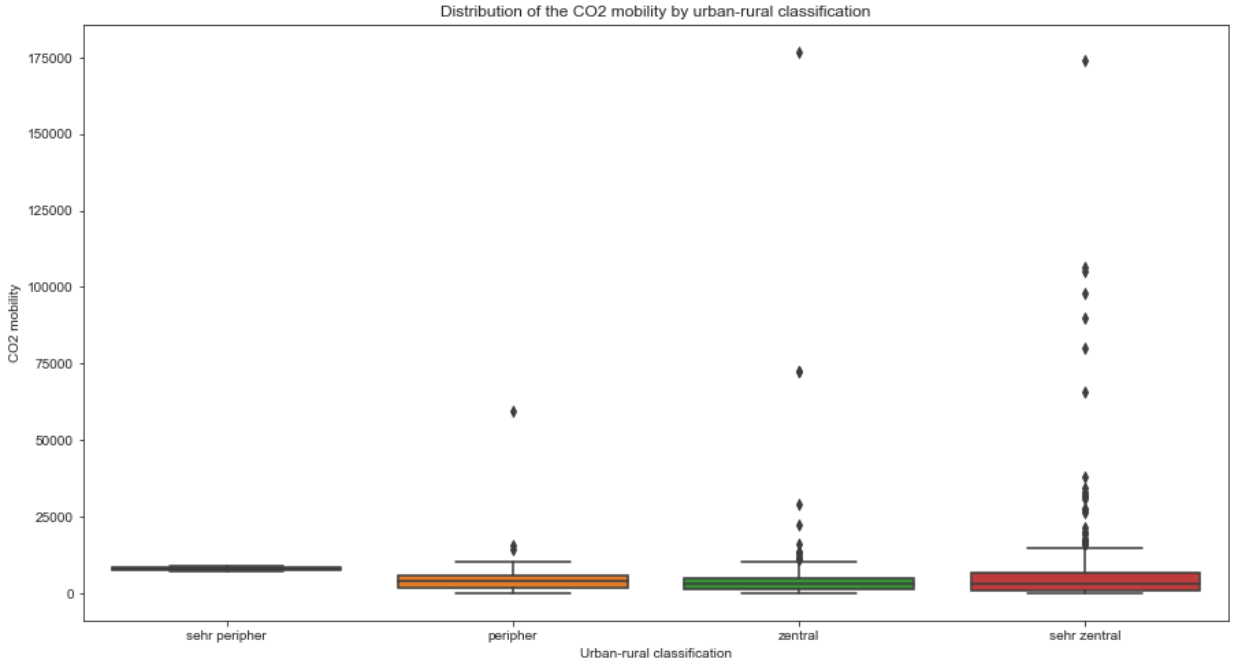

Distribution of the CO2 housing electricity by federal states
CO2 housing electricity
Federal state
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersachsen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenburg
Mecklenburg-Vorpommern

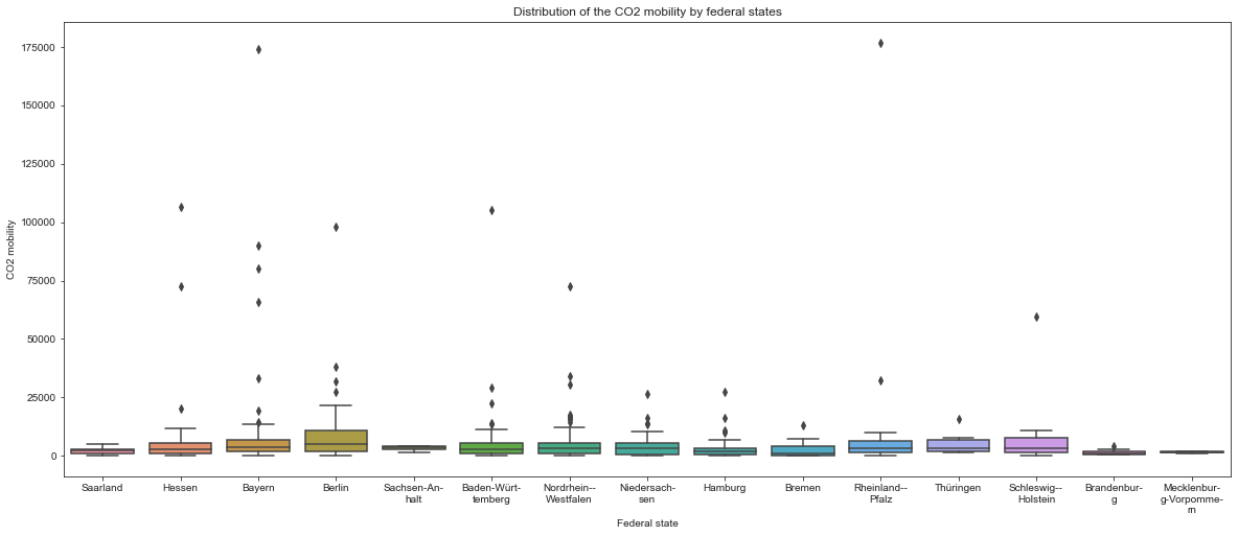

Distribution of the CO2 mobility by age
CO2 mobility
Age

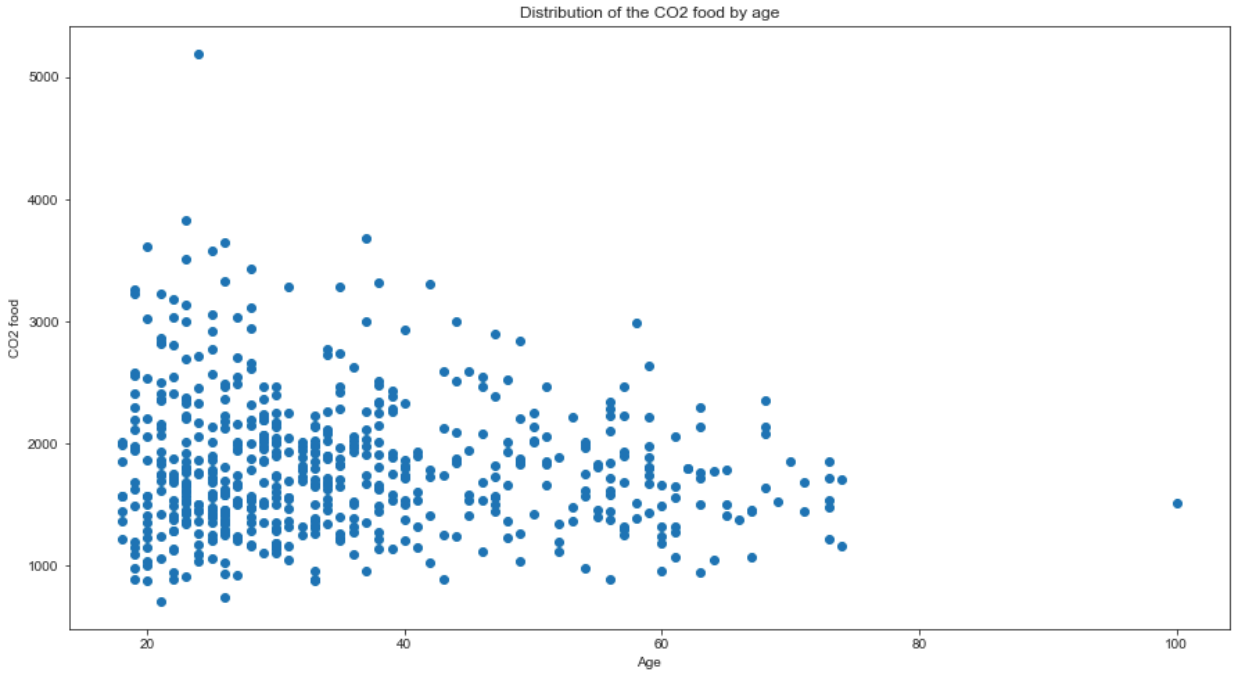

Distribution of the CO2 mobility by income
CO2 mobility
Income

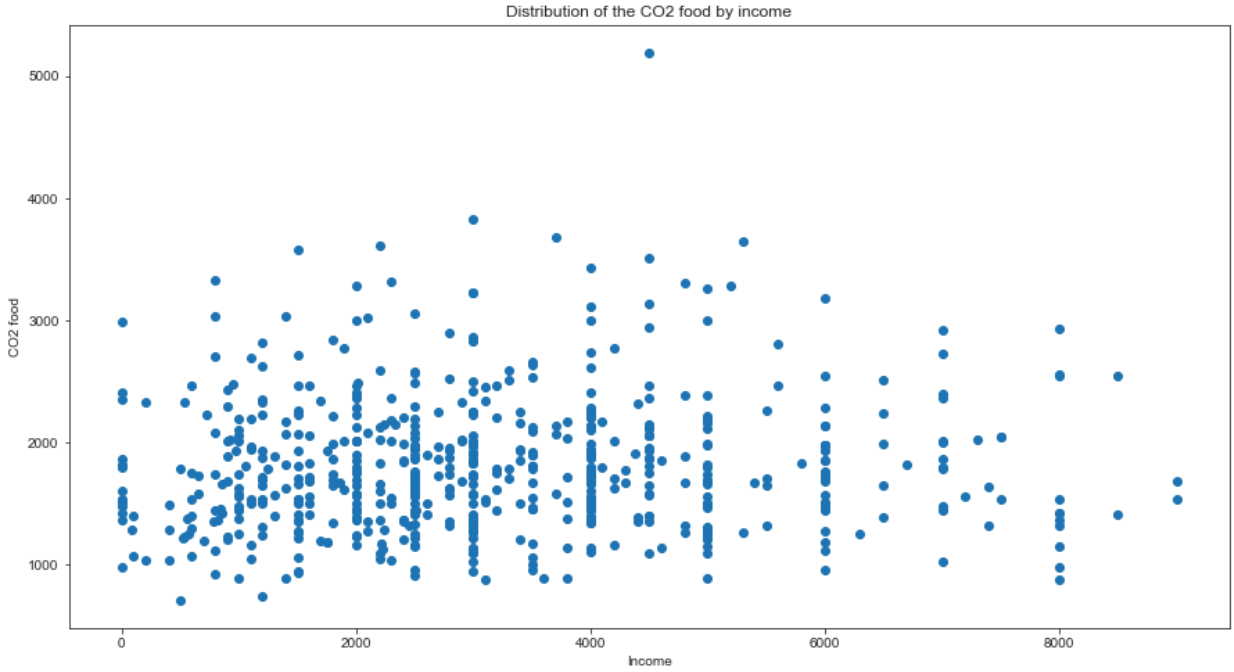

Distribution of the CO2 mobility by supporting political party
CO2 mobility
175000
150000
125000
100000
75000
50000
25000
0
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

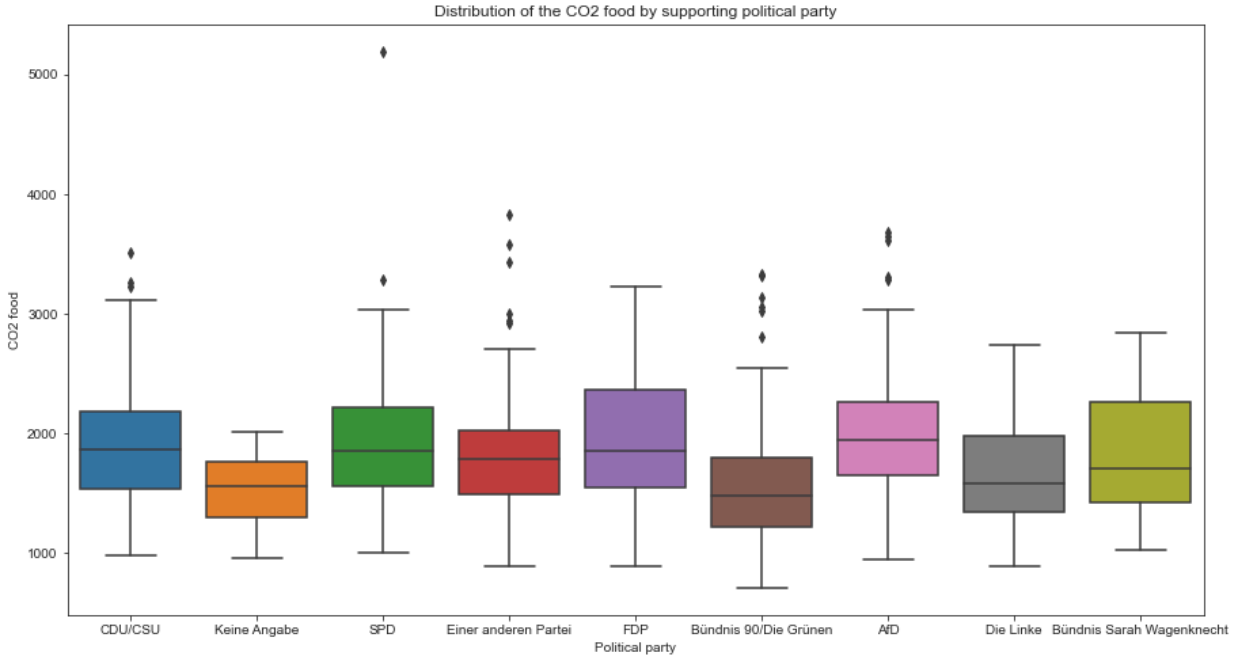

Distribution of the CO2 mobility by education level
CO2 mobility
175000
150000
125000
100000
75000
50000
25000
0
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

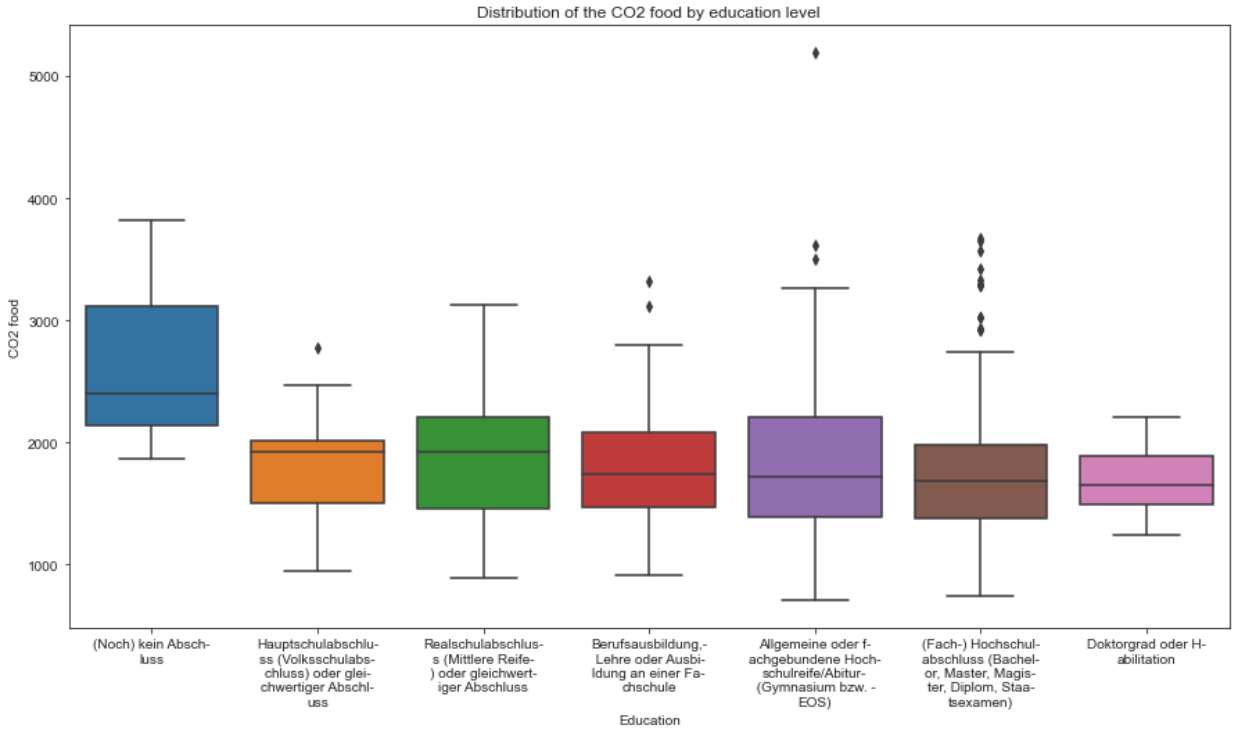

Distribution of the CO2 mobility by urban-rural classification
CO2 mobility
175000
150000
125000
100000
75000
50000
25000
0
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

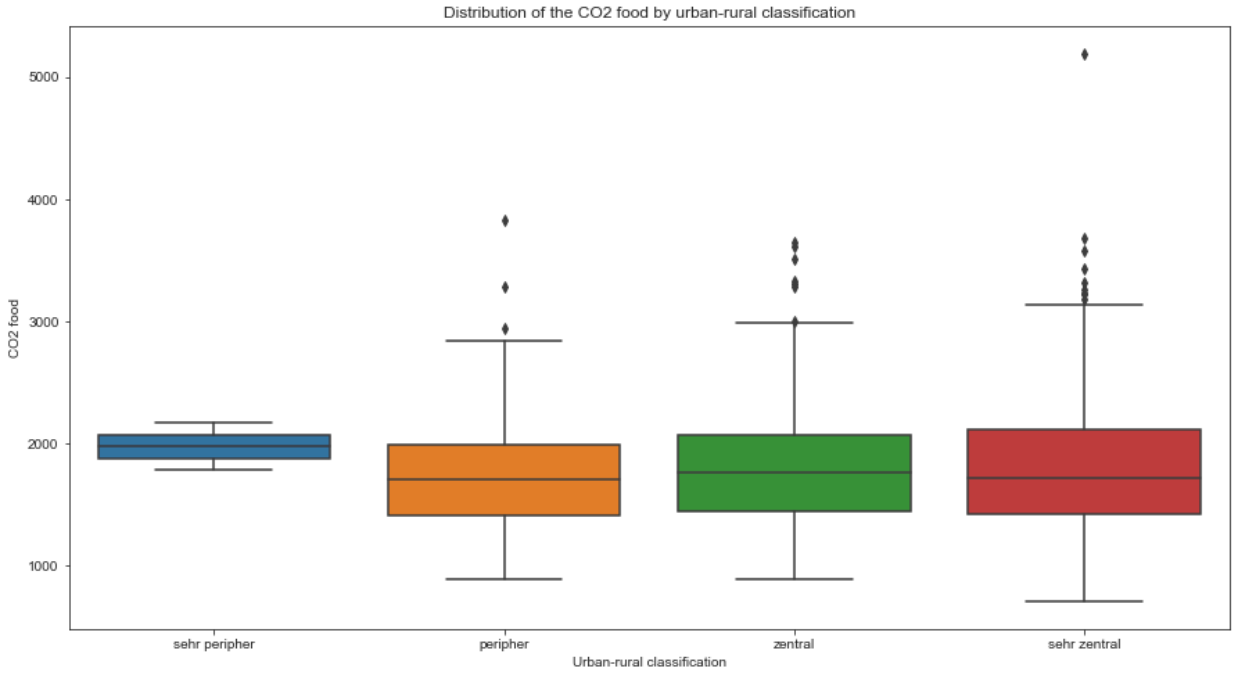

Distribution of the CO2 mobility by federal states
CO2 mobility
175000
150000
125000
100000
75000
50000
25000
0
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersach-sen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenbur-g
Mecklenbur-g-Vorpomme-rn
Federal state

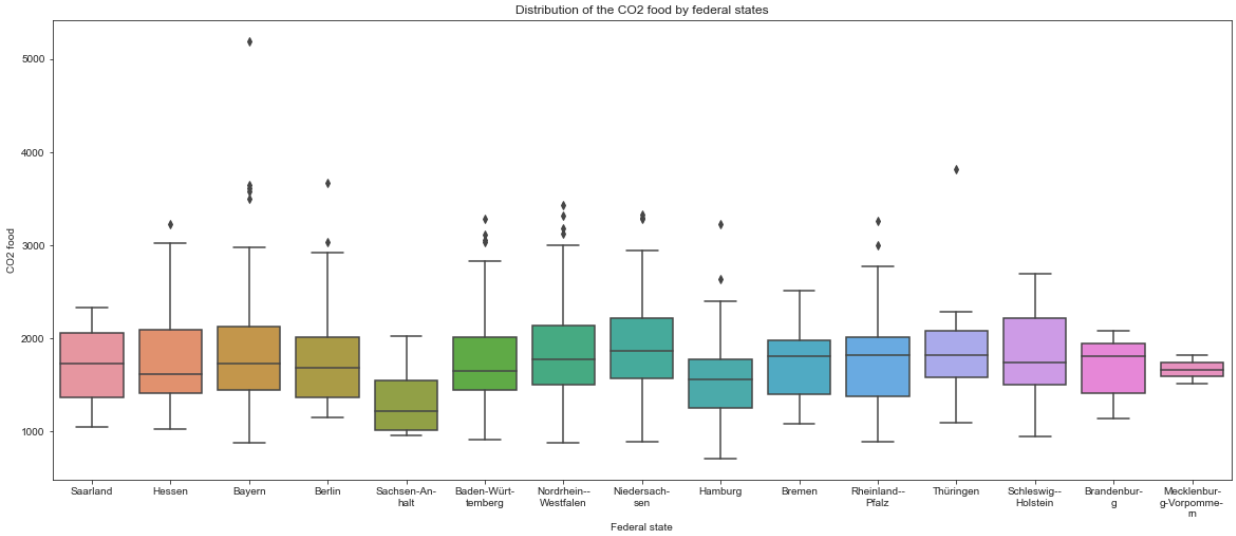

Distribution of the CO2 food by age
CO2 food
5000
4000
3000
2000
1000
20
40
60
80
100
Age

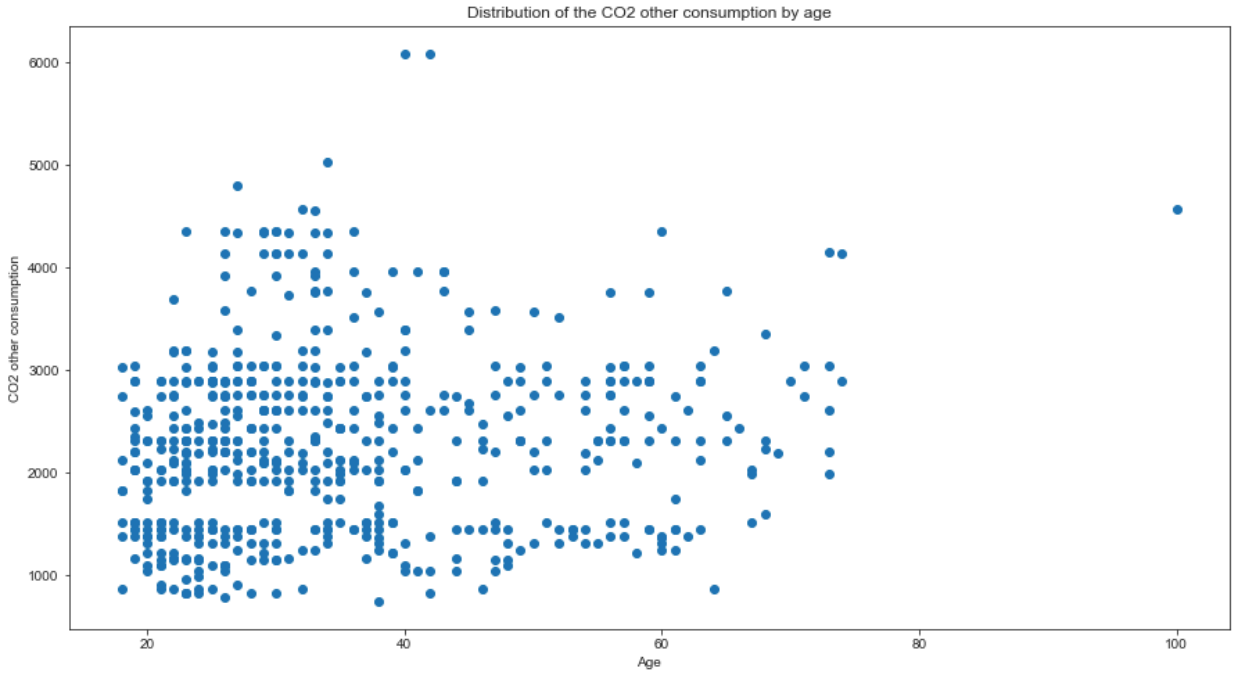

Distribution of the CO2 food by income
CO2 food
Income
5000
4000
3000
2000
1000
0
2000
4000
6000
8000

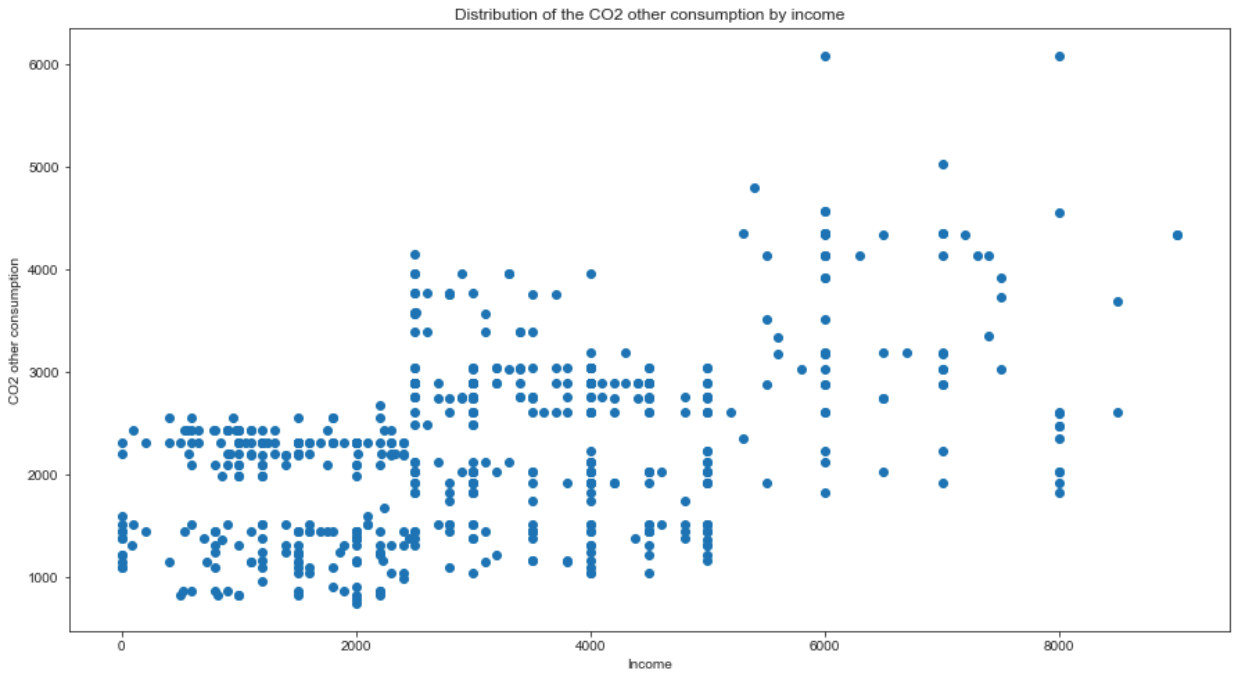

Distribution of the CO2 food by supporting political party
CO2 food
Political party
5000
4000
3000
2000
1000
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht

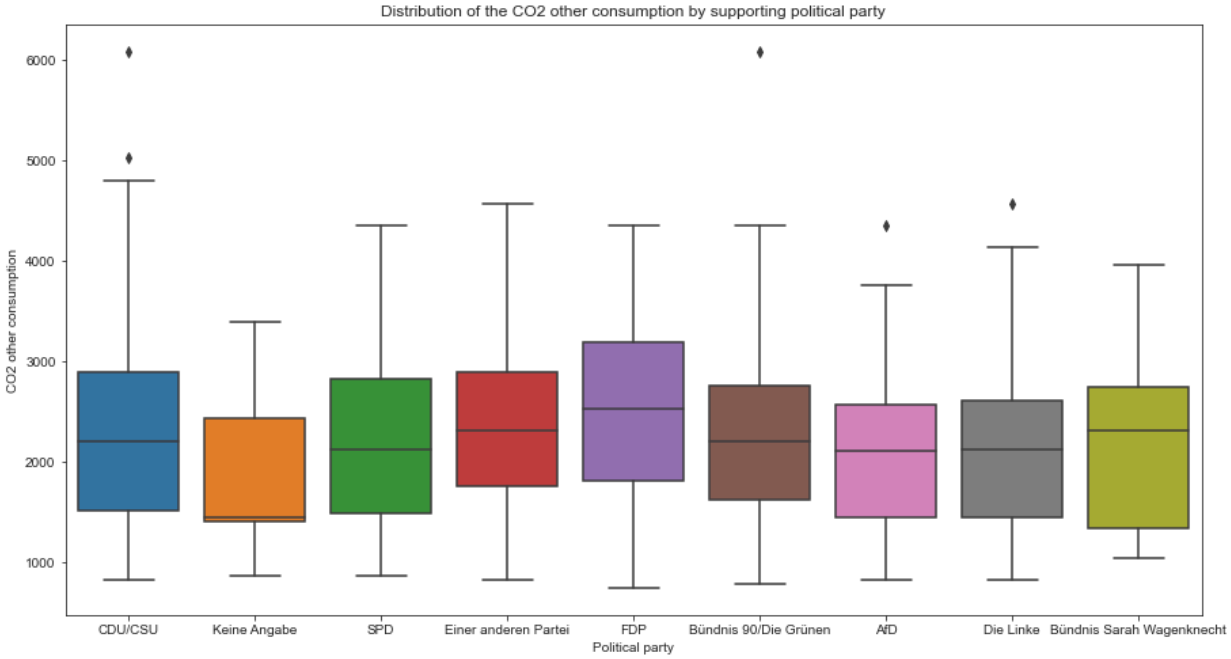

Distribution of the CO2 food by education level
CO2 food
5000
4000
3000
2000
1000
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

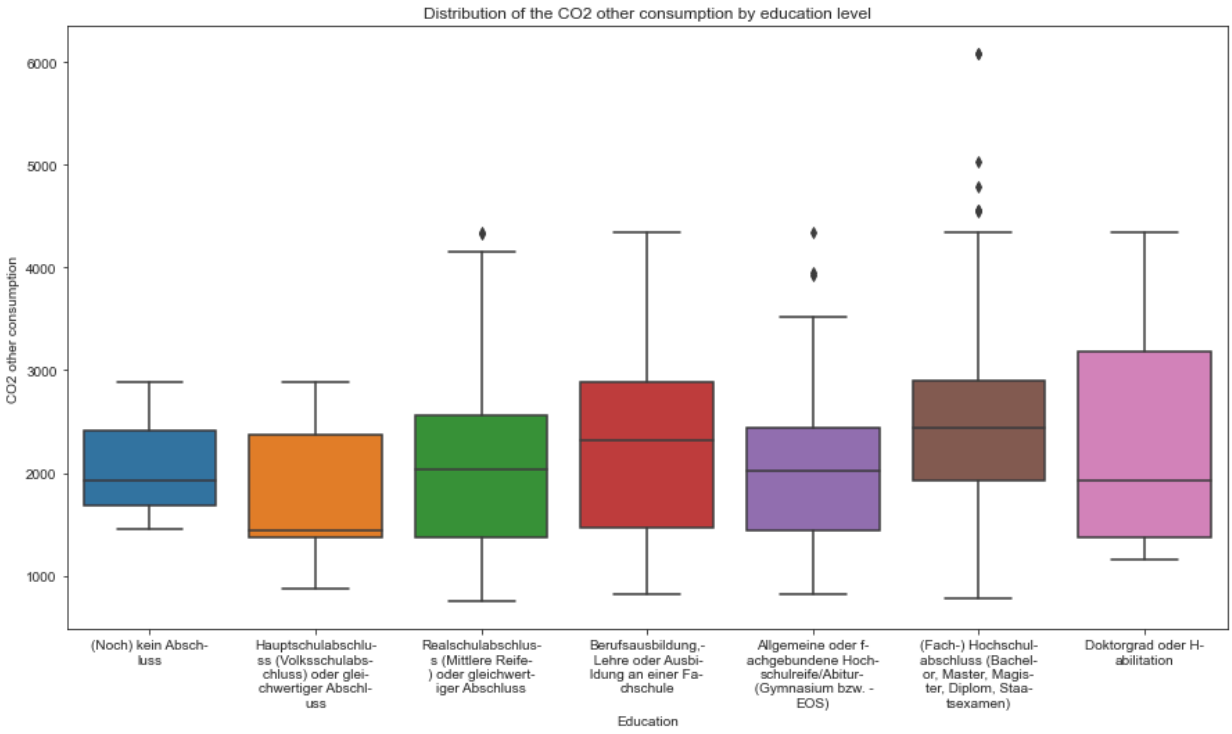

Distribution of the CO2 food by urban-rural classification
CO2 food
5000
4000
3000
2000
1000
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

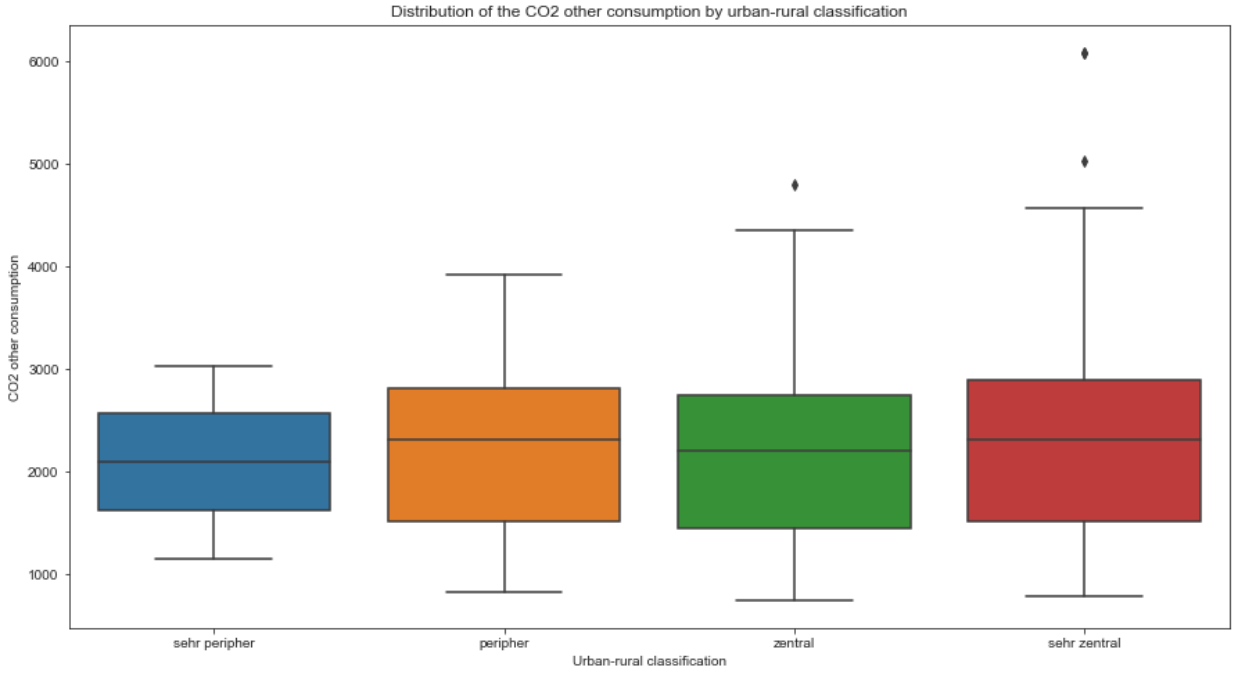

Distribution of the CO2 food by federal states
CO2 food
Federal state
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersachsen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenburg
Mecklenburg-Vorpommern

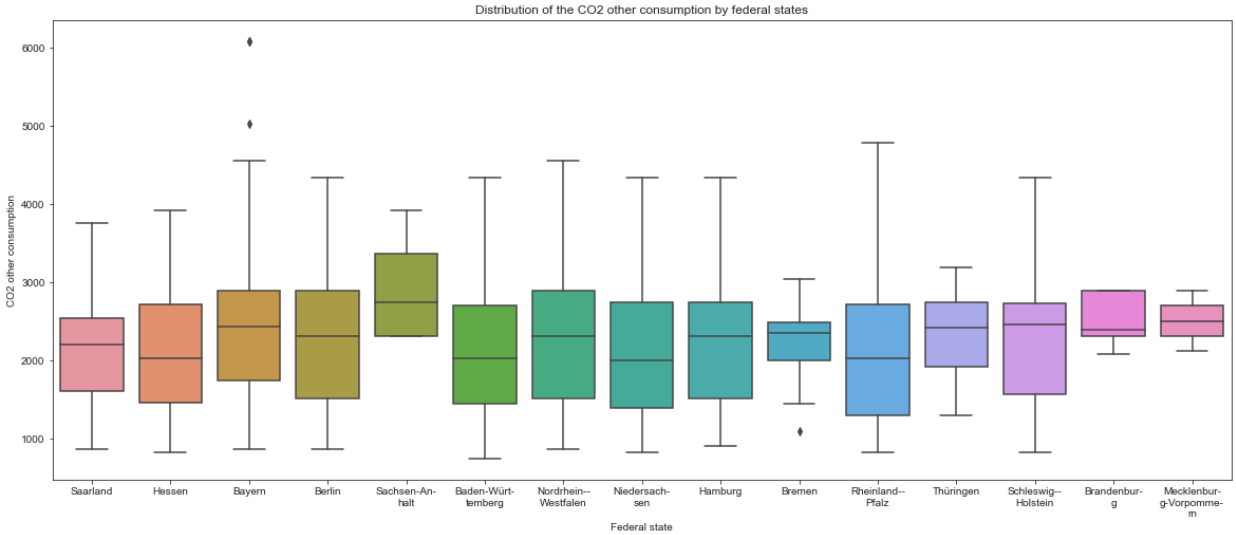

Distribution of the CO2 other consumption by age
CO2 other consumption
Age


Distribution of the CO2 other consumption by income
CO2 other consumption
Income

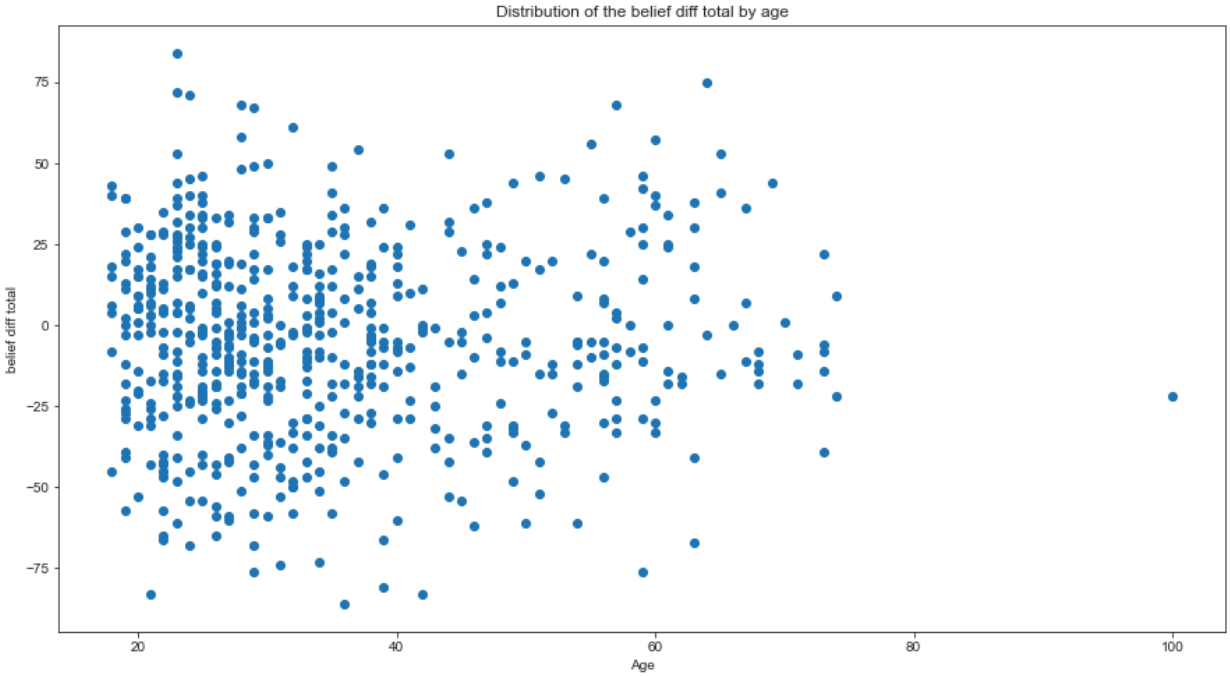

Distribution of the CO2 other consumption by supporting political party
CO2 other consumption
6000
5000
4000
3000
2000
1000
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

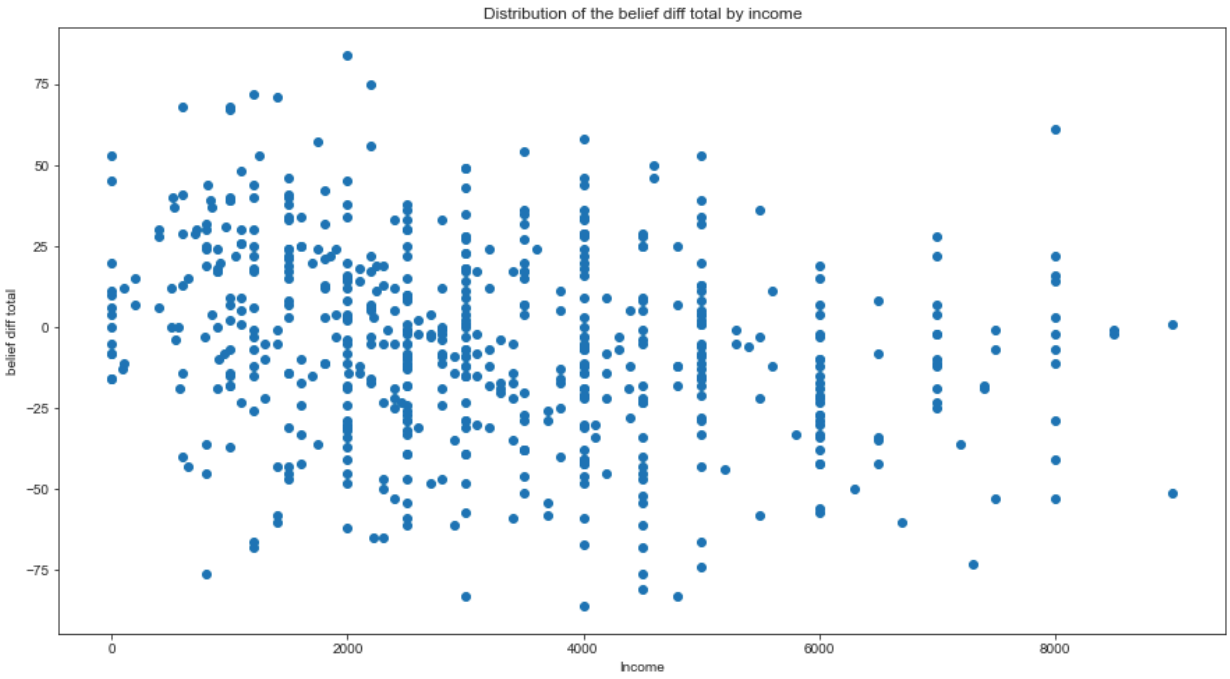

Distribution of the CO2 other consumption by education level
CO2 other consumption
6000
5000
4000
3000
2000
1000
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,- Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

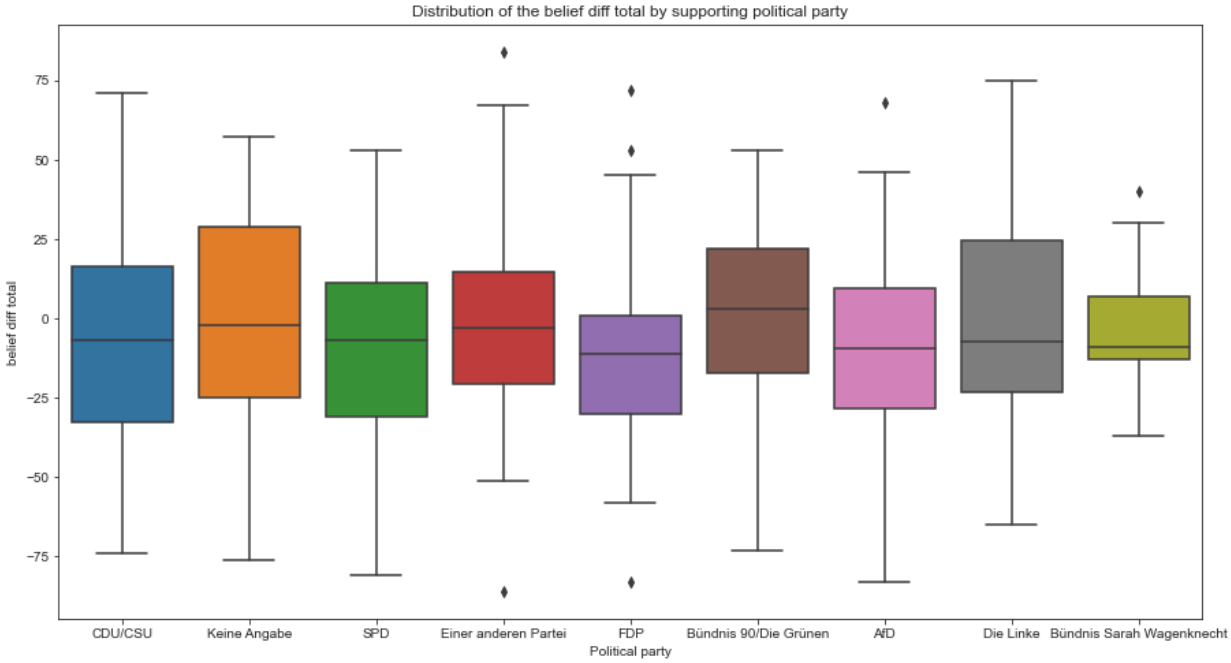


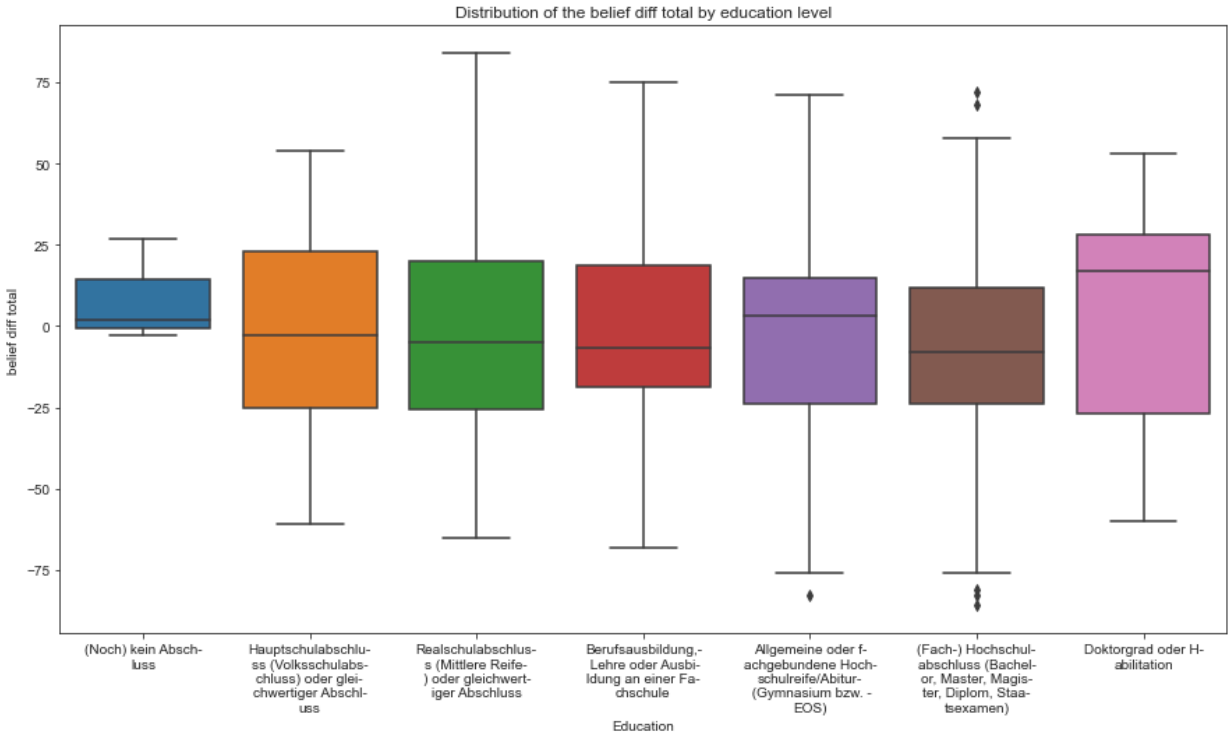


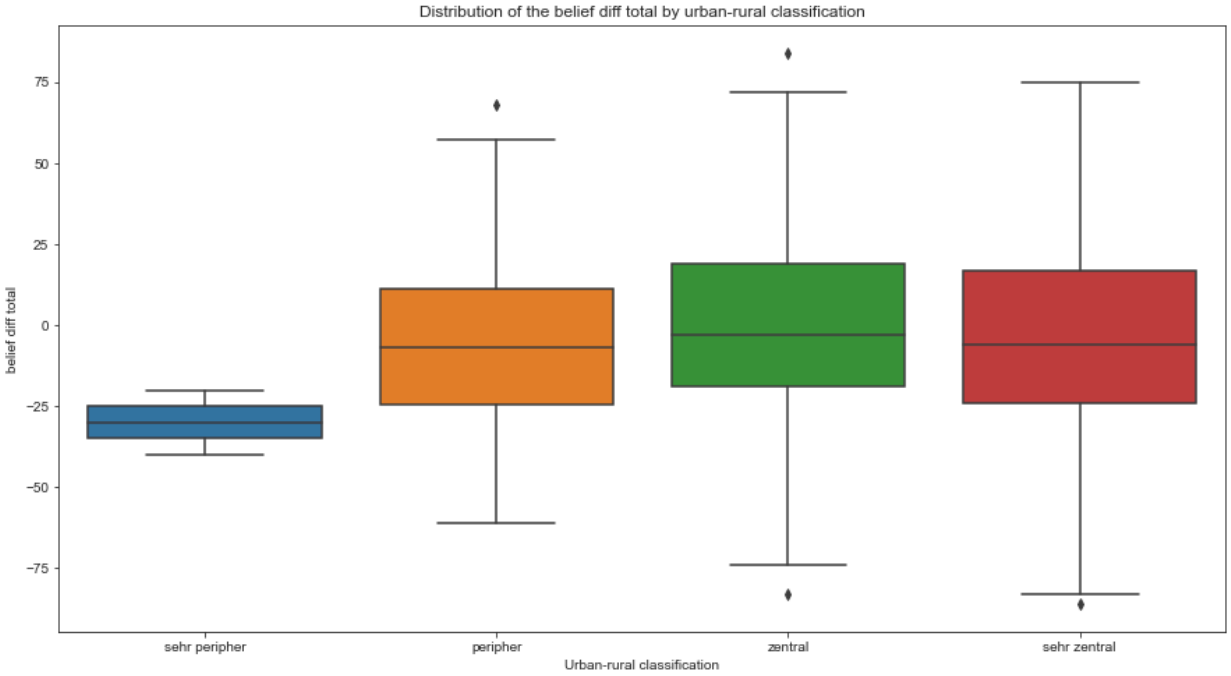
In [34]:
```
### Create the distribution plots for belief_diff variables

dependent_vars = ['belief_diff_total', 'belief_diff_housing_electricity', 'belief_diff

for dependent_var in dependent_vars:
    run_all_var(dependent_var)
```

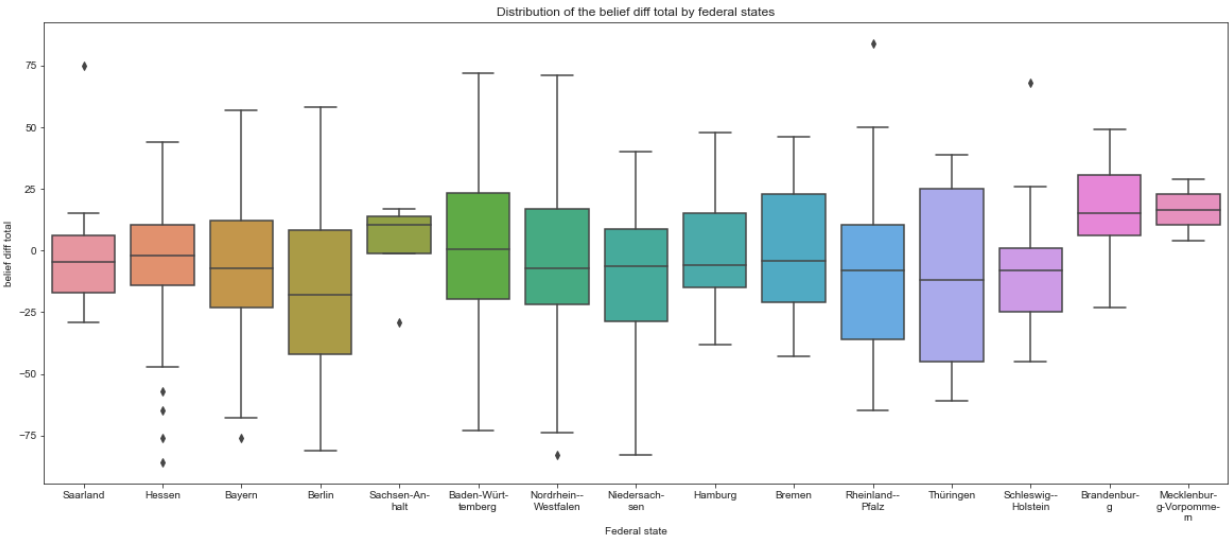

Distribution of the belief diff total by age
belief diff total
75
50
25
0
−25
−50
−75
20
40
60
80
100
Age

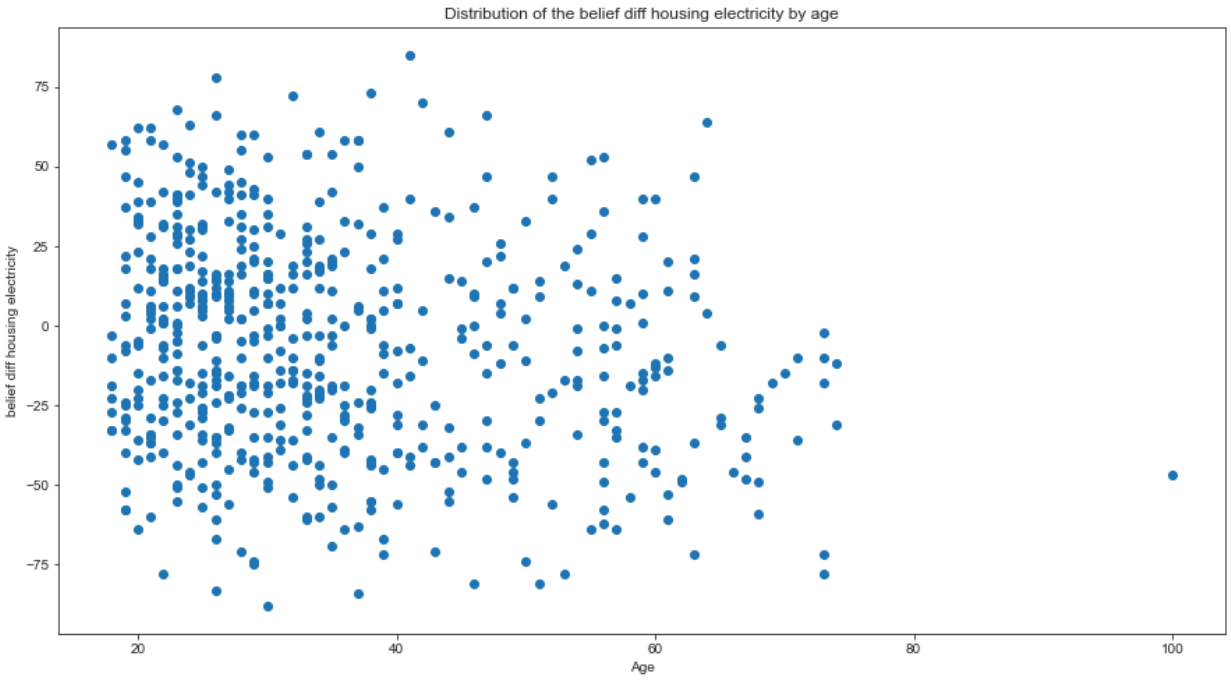

Distribution of the belief diff total by income
belief diff total
75
50
25
0
−25
−50
−75
0
2000
4000
6000
8000
Income

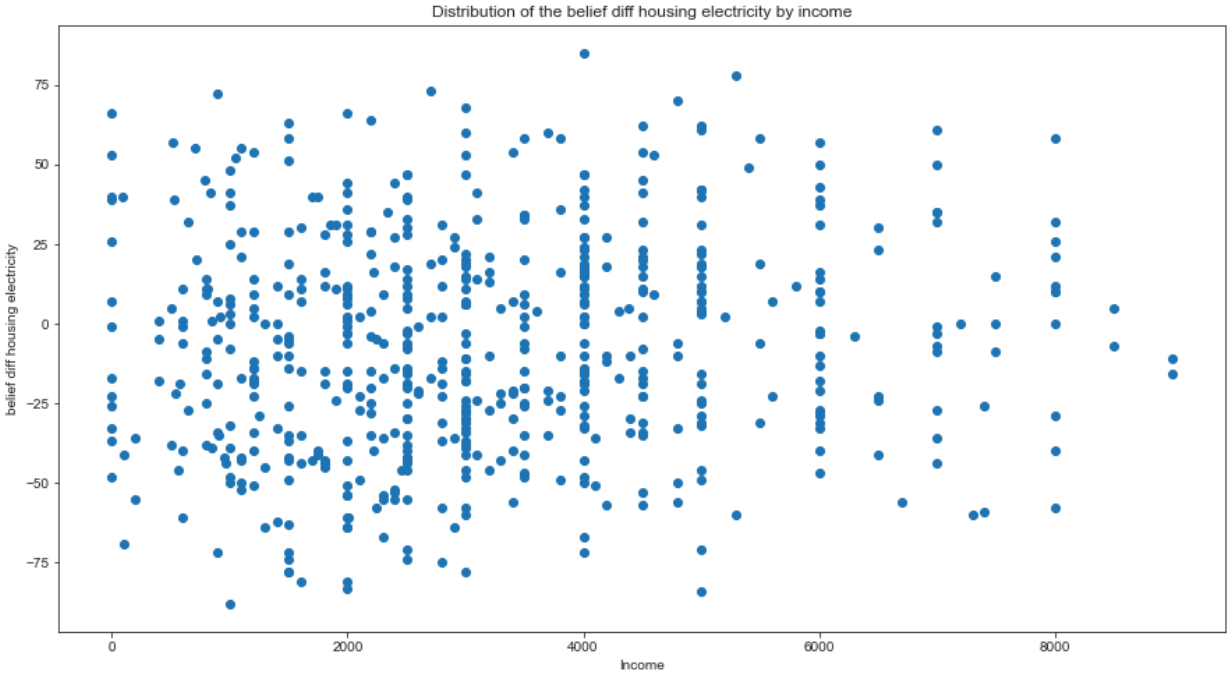

Distribution of the belief diff total by supporting political party
belief diff total
75
50
25
0
−25
−50
−75
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

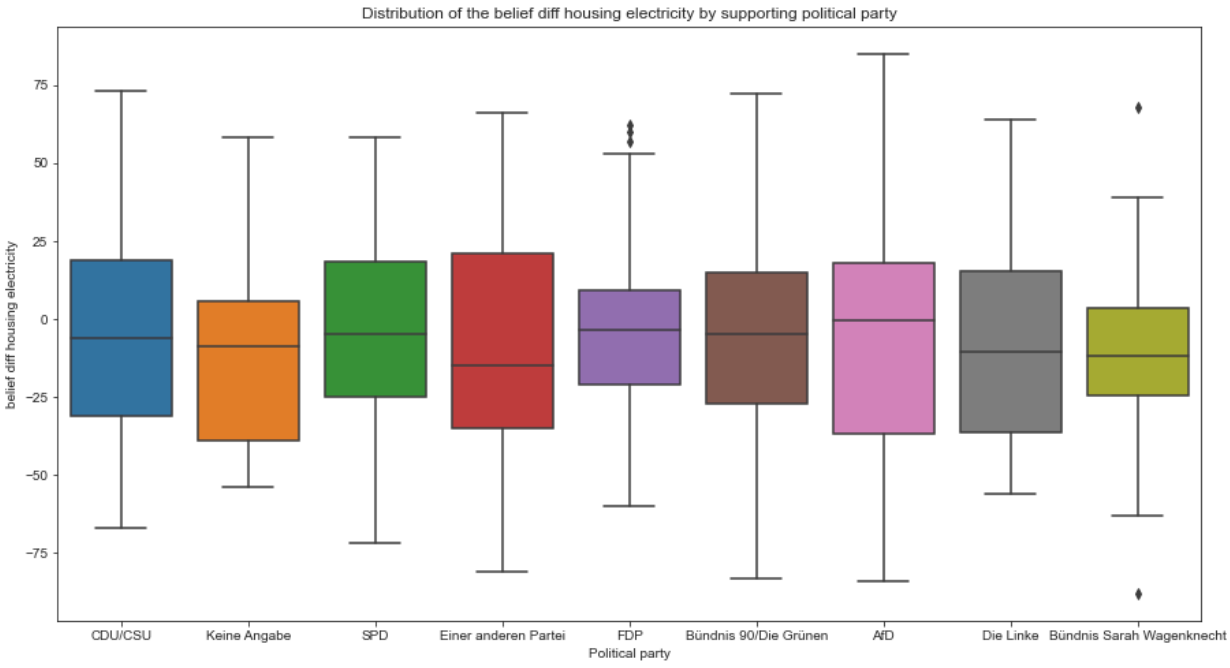

Distribution of the belief diff total by education level
belief diff total
75
50
25
0
−25
−50
−75
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

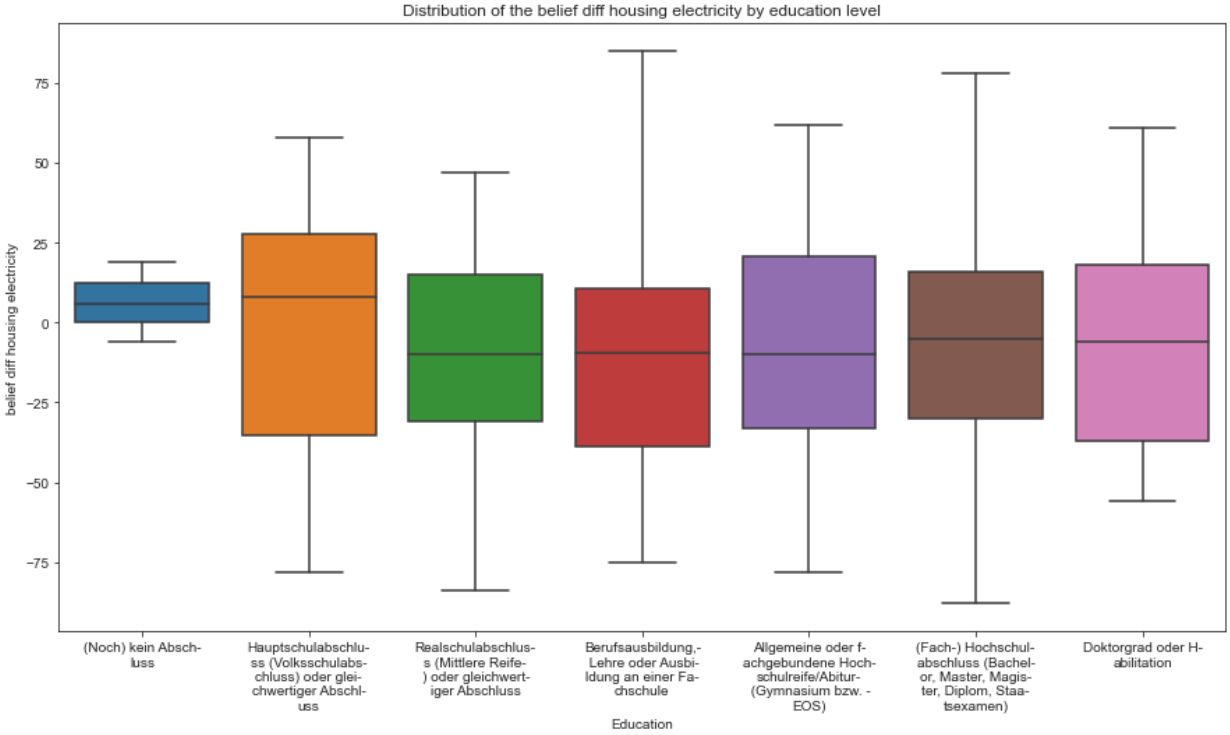

Distribution of the belief diff total by urban-rural classification
belief diff total
75
50
25
0
−25
−50
−75
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

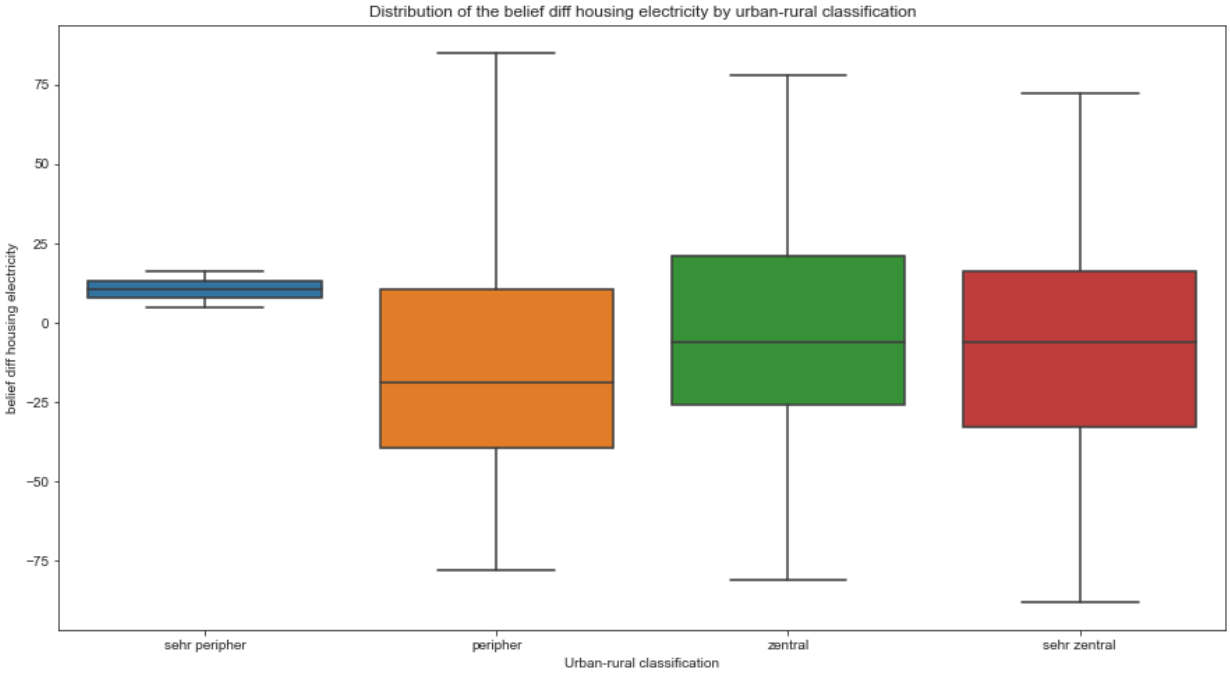

Distribution of the belief diff total by federal states
belief diff total
75
50
25
0
−25
−50
−75
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersachsen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenburg
Mecklenburg-Vorpommern
Federal state

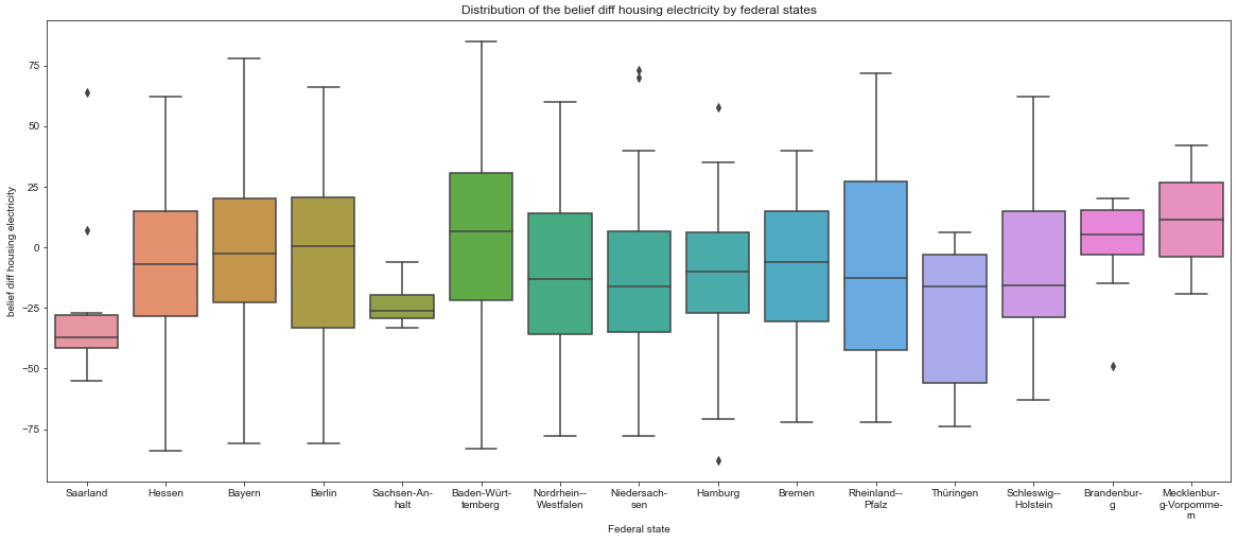

Distribution of the belief diff housing electricity by age
belief diff housing electricity
75
50
25
0
−25
−50
−75
20
40
60
80
100
Age

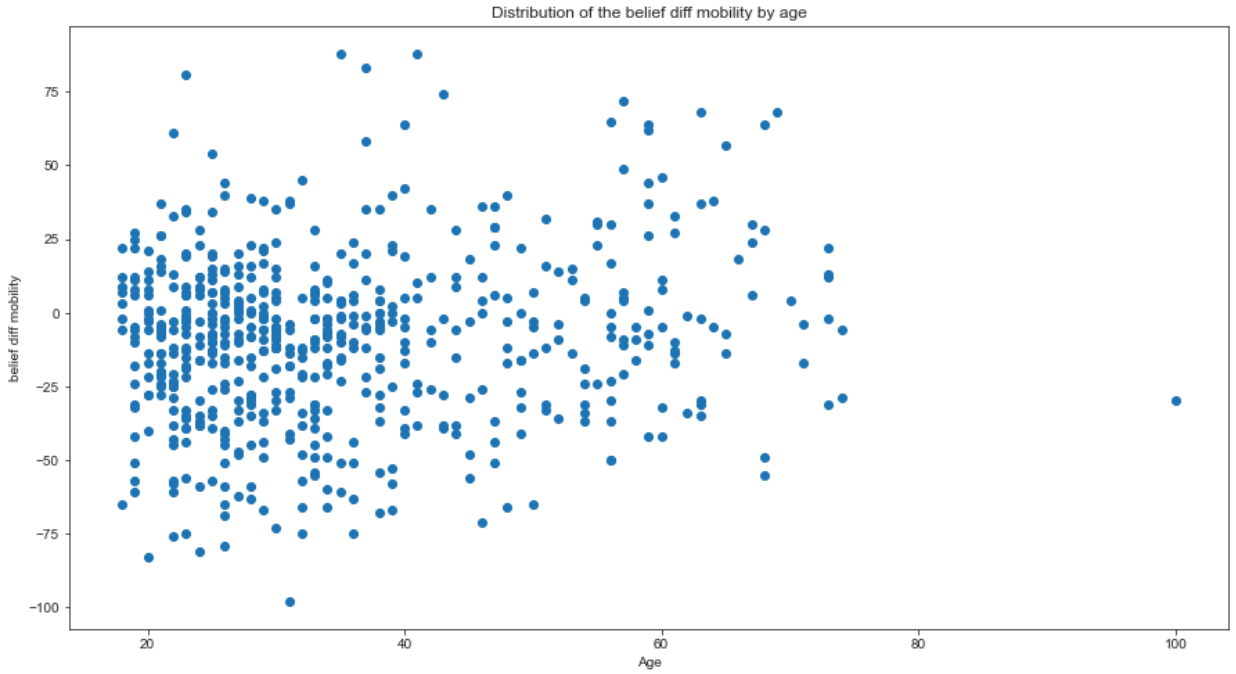

Distribution of the belief diff housing electricity by income
belief diff housing electricity
75
50
25
0
−25
−50
−75
0
2000
4000
6000
8000
Income

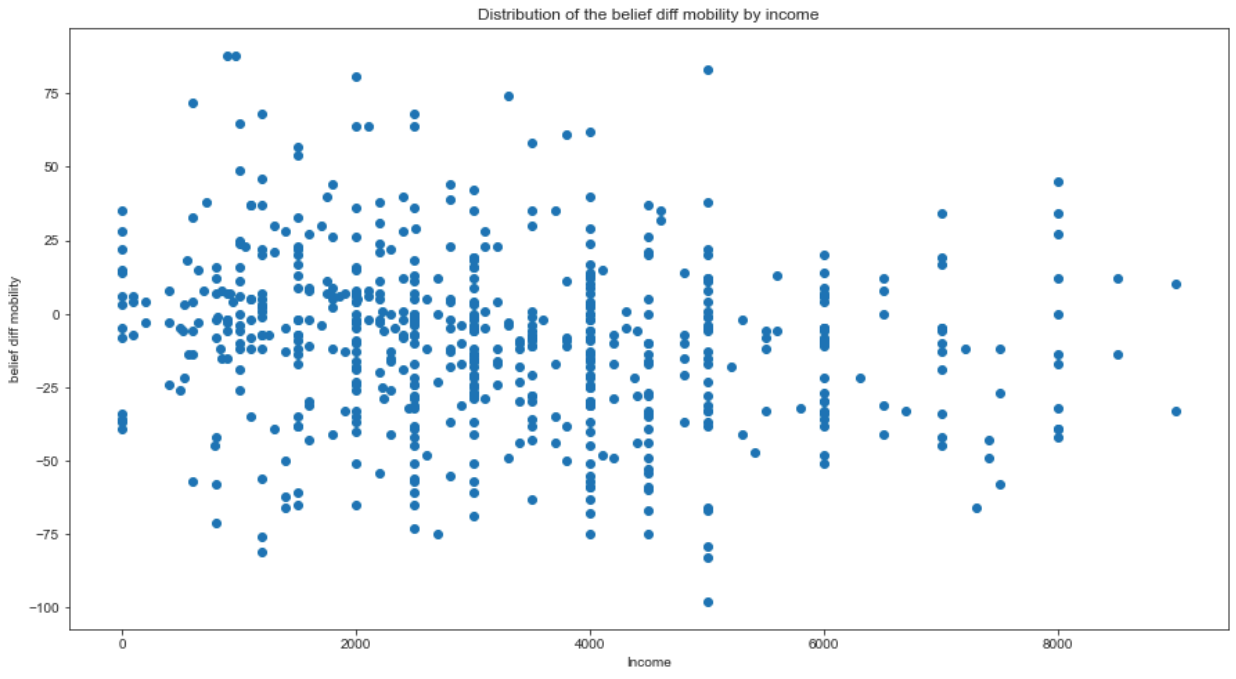

Distribution of the belief diff housing electricity by supporting political party
belief diff housing electricity
75
50
25
0
−25
−50
−75
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

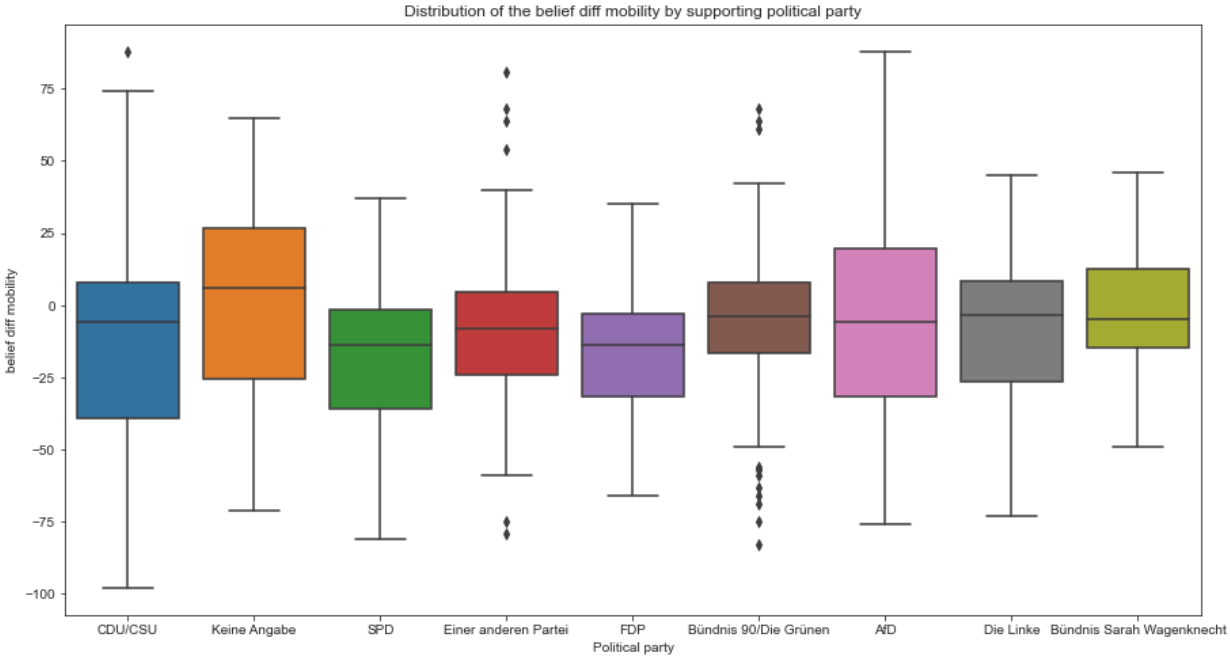

Distribution of the belief diff housing electricity by education level
belief diff housing electricity
75
50
25
0
−25
−50
−75
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

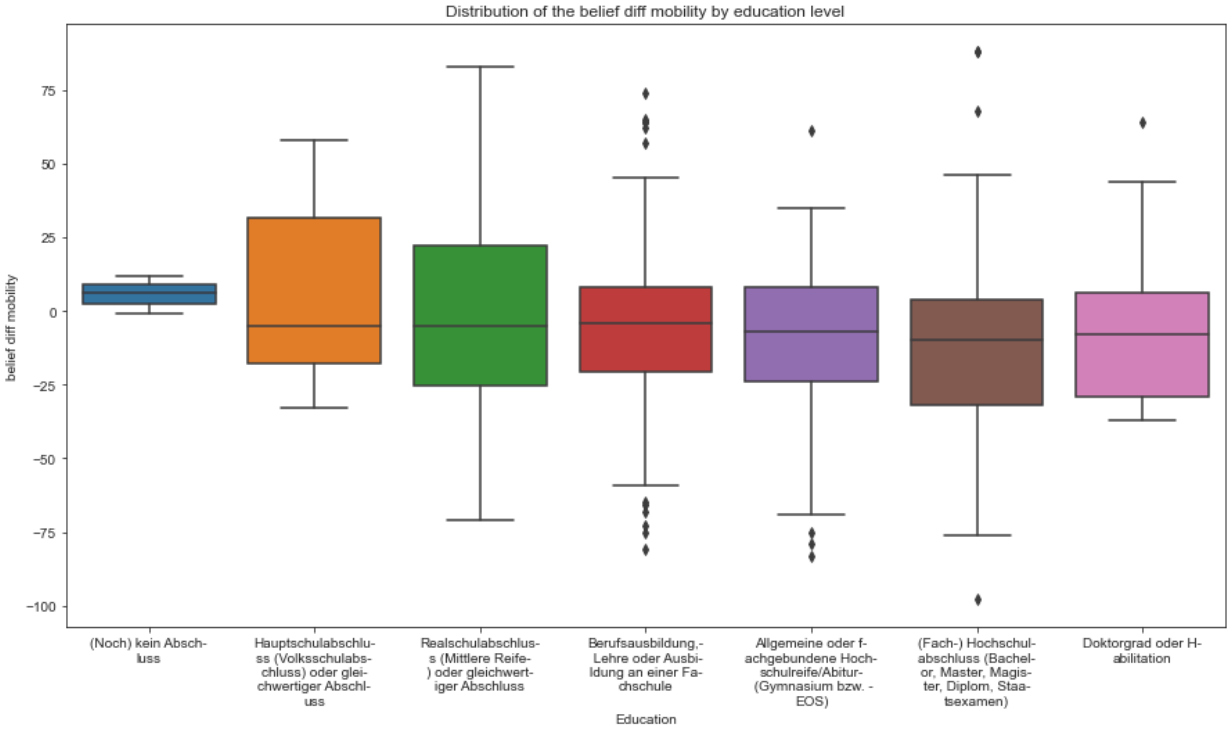

Distribution of the belief diff housing electricity by urban-rural classification
belief diff housing electricity
75
50
25
0
−25
−50
−75
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

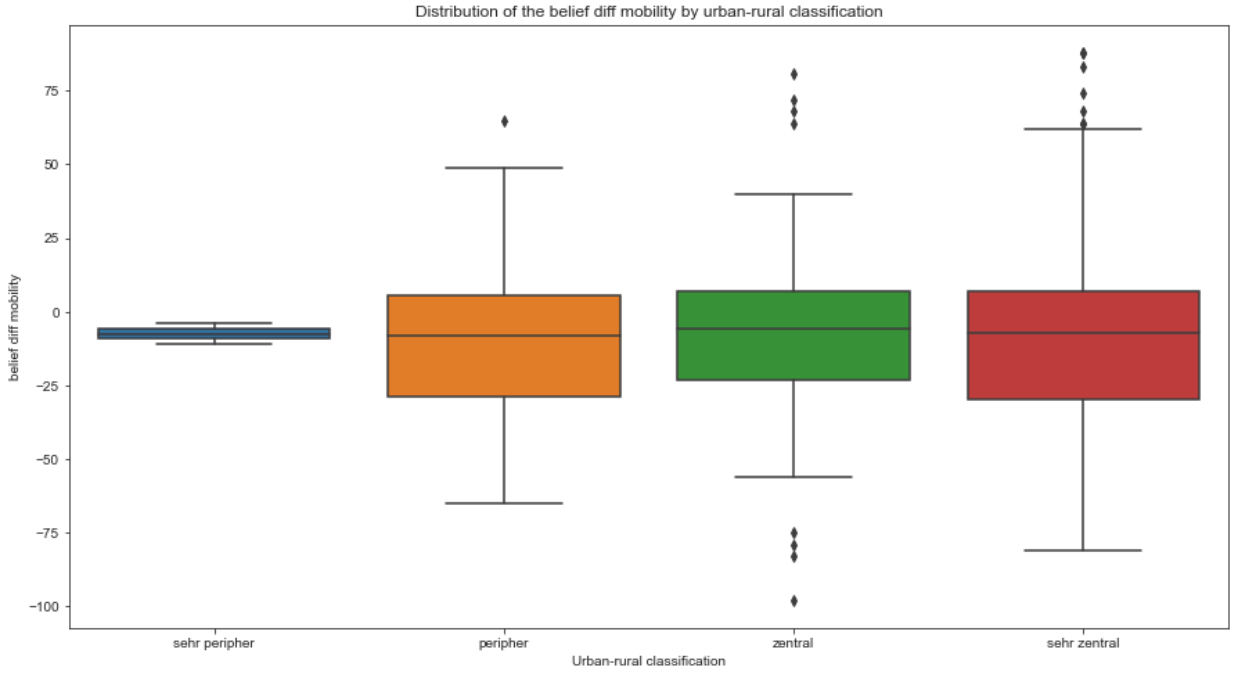

Distribution of the belief diff housing electricity by federal states
belief diff housing electricity
75
50
25
0
−25
−50
−75
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württ-temberg
Nordrhein--Westfalen
Niedersach-sen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenbur-g
Mecklenbur-g-Vorpomme-rn
Federal state

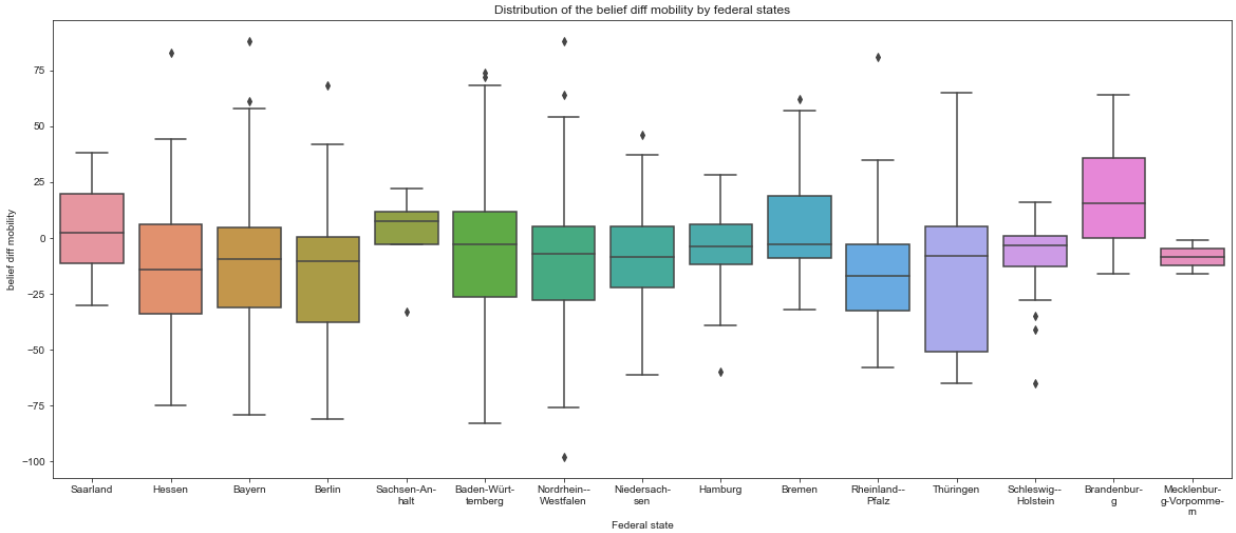

Distribution of the belief diff mobility by age
belief diff mobility
75
50
25
0
−25
−50
−75
−100
20
40
60
80
100
Age

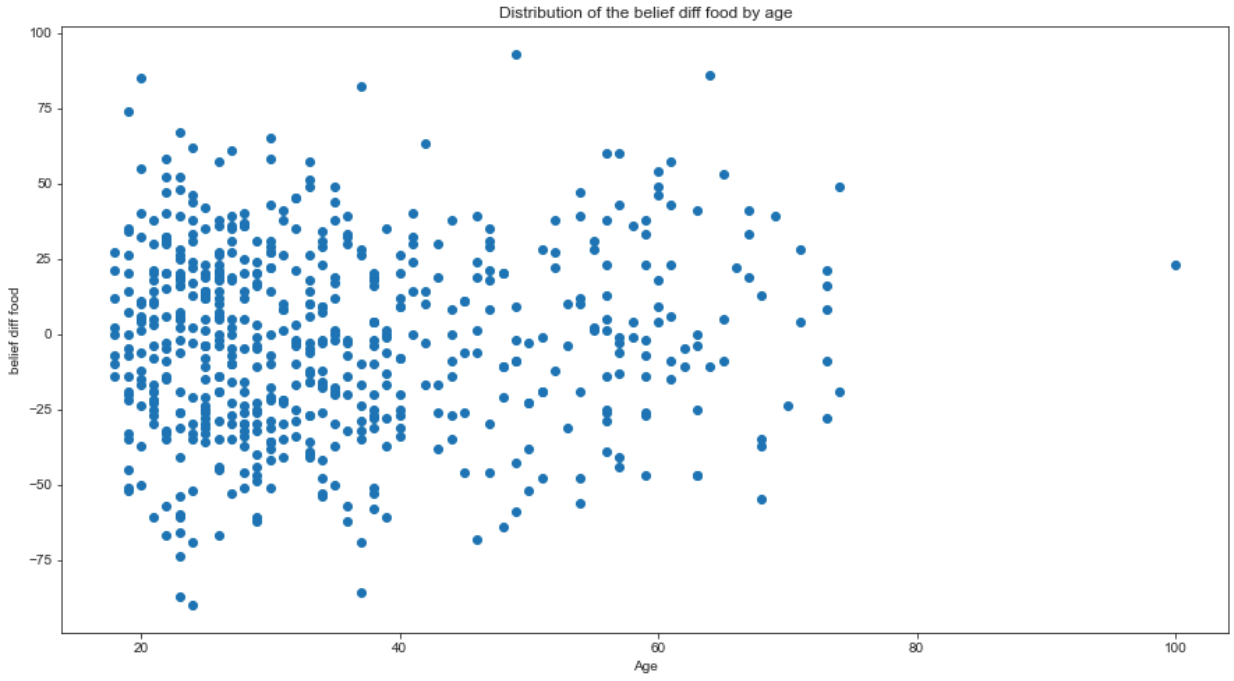

Distribution of the belief diff mobility by income
belief diff mobility
75
50
25
0
−25
−50
−75
−100
0
2000
4000
6000
8000
Income

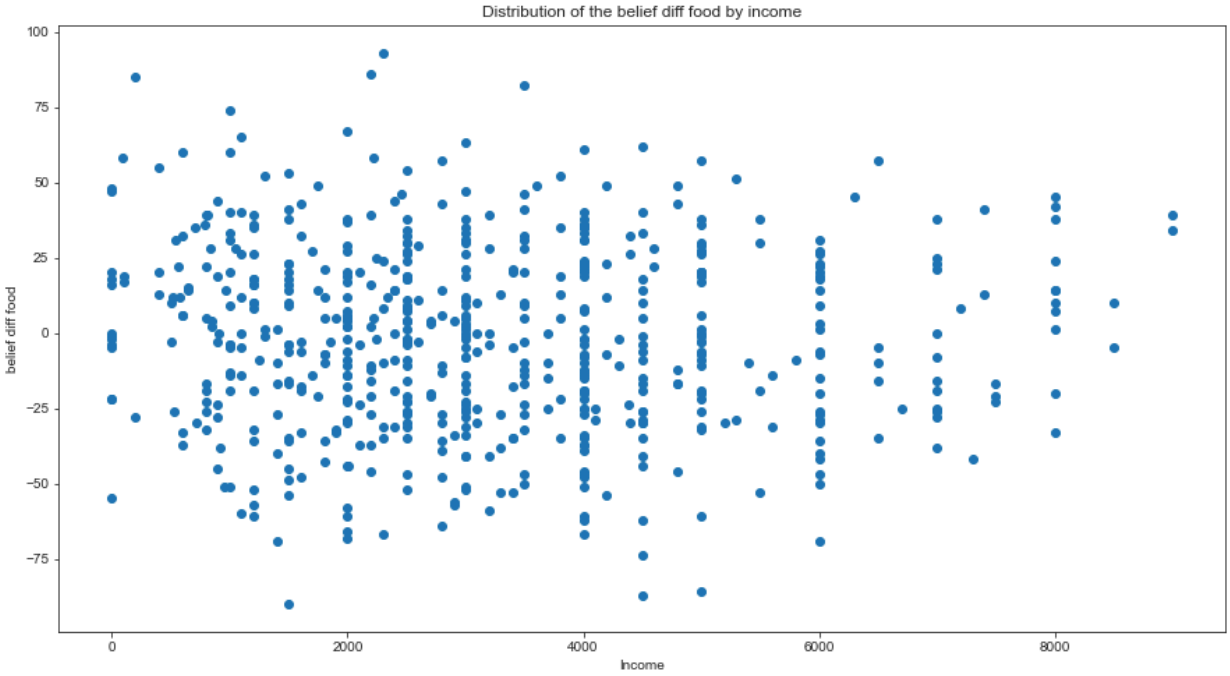

Distribution of the belief diff mobility by supporting political party
belief diff mobility
75
50
25
0
−25
−50
−75
−100
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

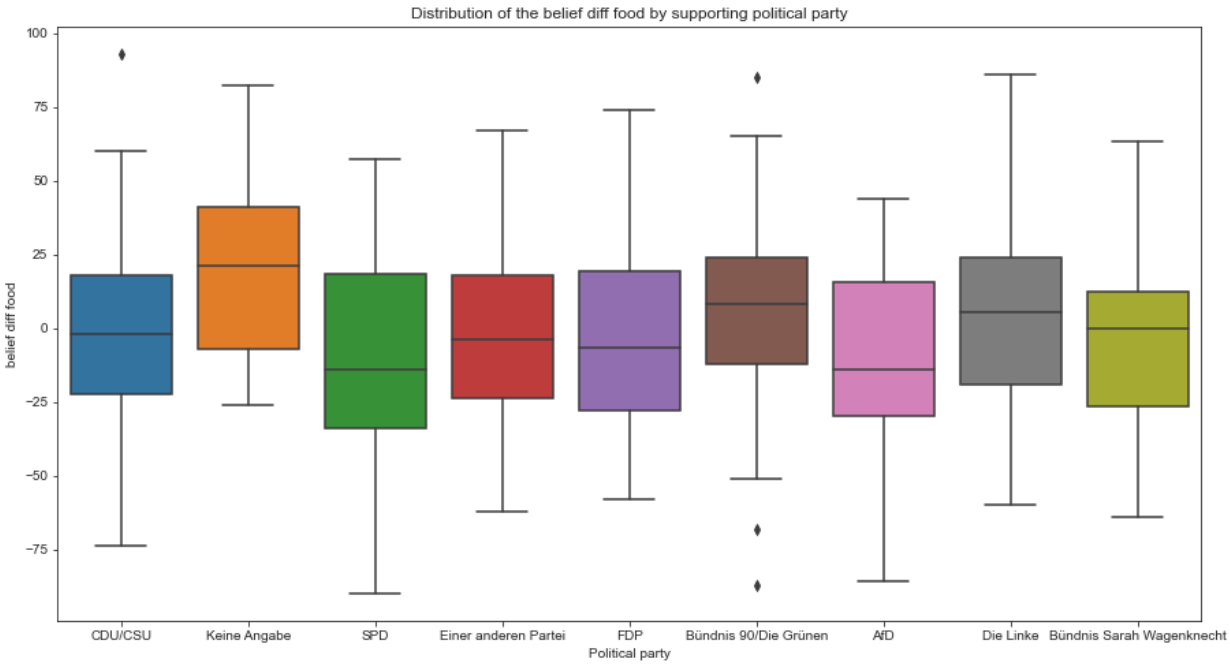

Distribution of the belief diff mobility by education level
belief diff mobility
75
50
25
0
−25
−50
−75
−100
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

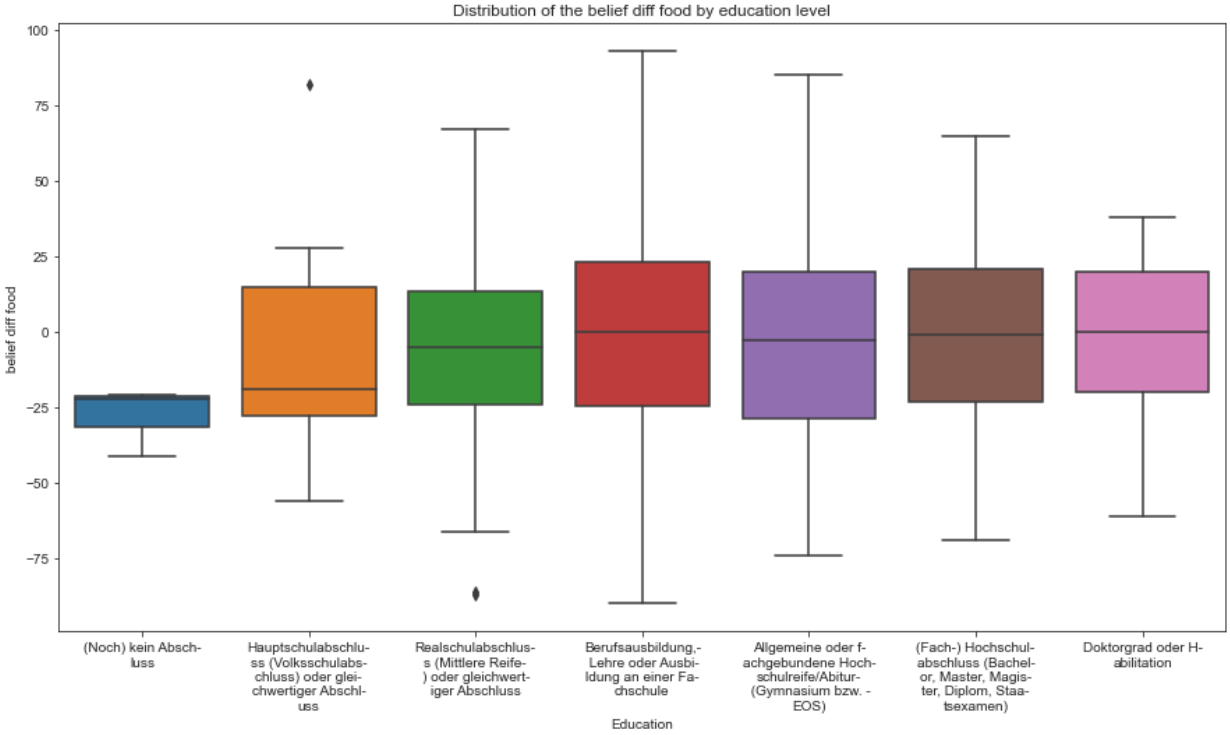

Distribution of the belief diff mobility by urban-rural classification
belief diff mobility
75
50
25
0
−25
−50
−75
−100
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

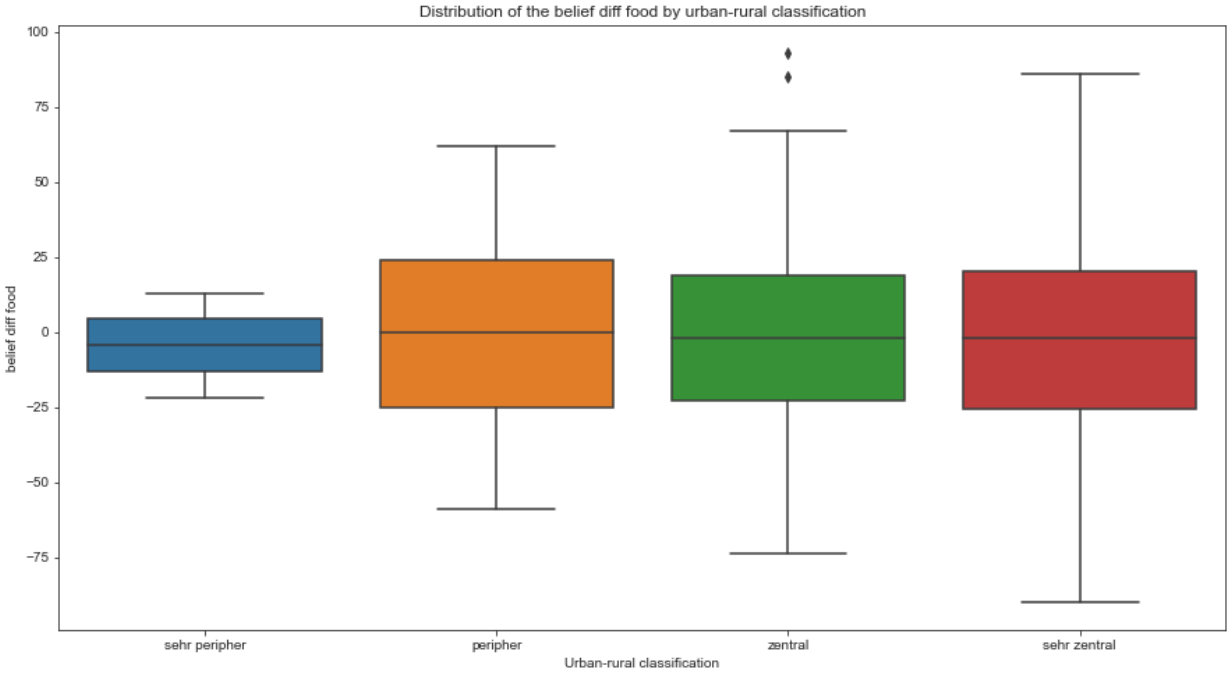

Distribution of the belief diff mobility by federal states
belief diff mobility
75
50
25
0
−25
−50
−75
−100
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Würt-temberg
Nordrhein--Westfalen
Niedersach-sen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenbur-g
Mecklenbur-g-Vorpomme-rn
Federal state

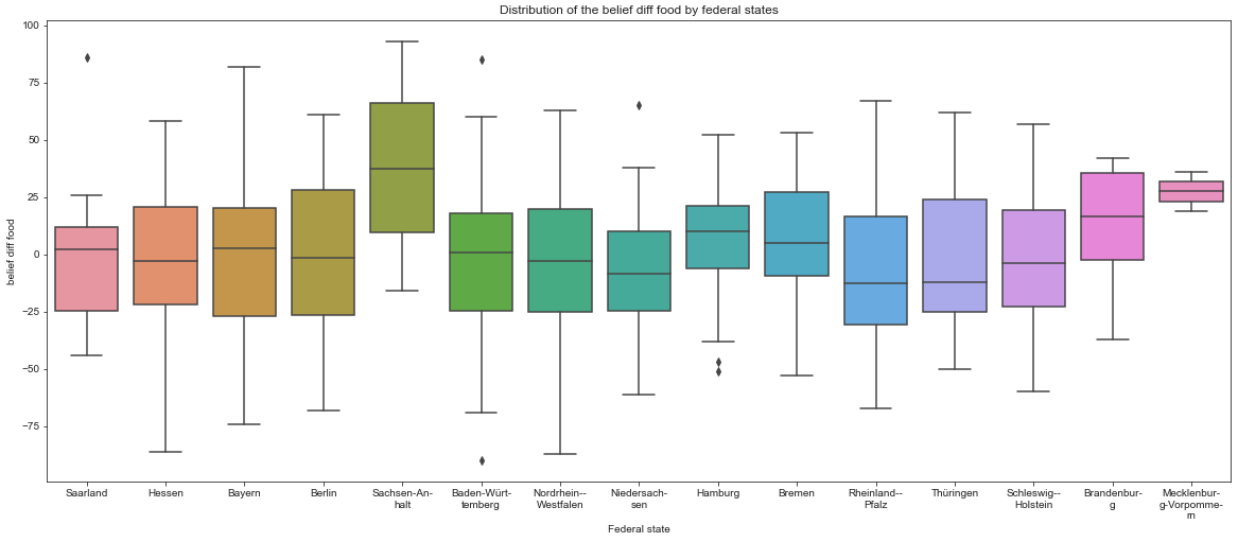

Distribution of the belief diff food by age
belief diff food
100
75
50
25
0
−25
−50
−75
20
40
60
80
100
Age

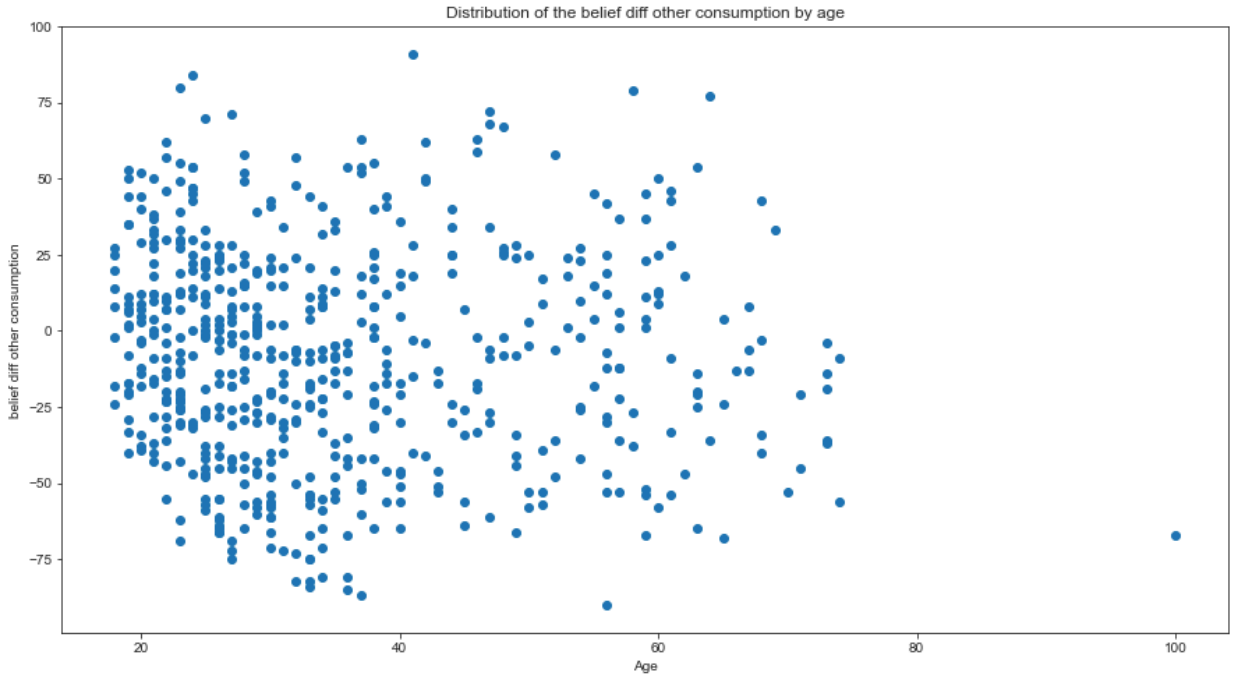

Distribution of the belief diff food by income
belief diff food
100
75
50
25
0
−25
−50
−75
0
2000
4000
6000
8000
Income

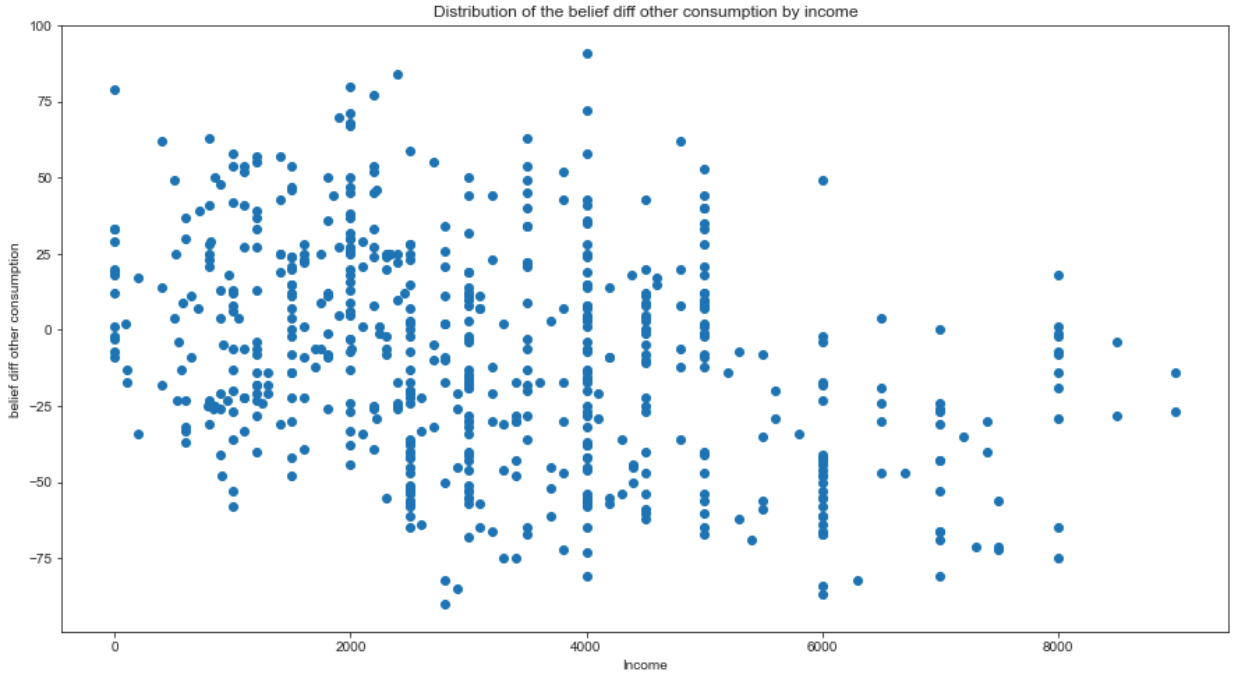

Distribution of the belief diff food by supporting political party
belief diff food
100
75
50
25
0
−25
−50
−75
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party

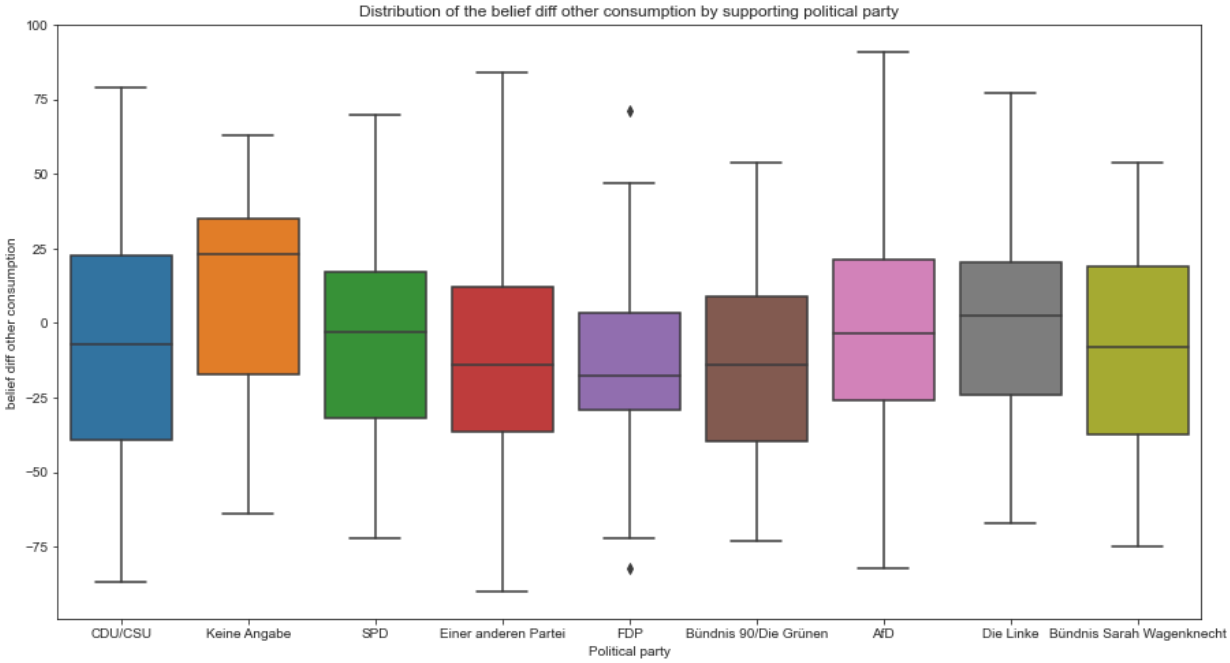

Distribution of the belief diff food by education level
belief diff food
100
75
50
25
0
−25
−50
−75
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education

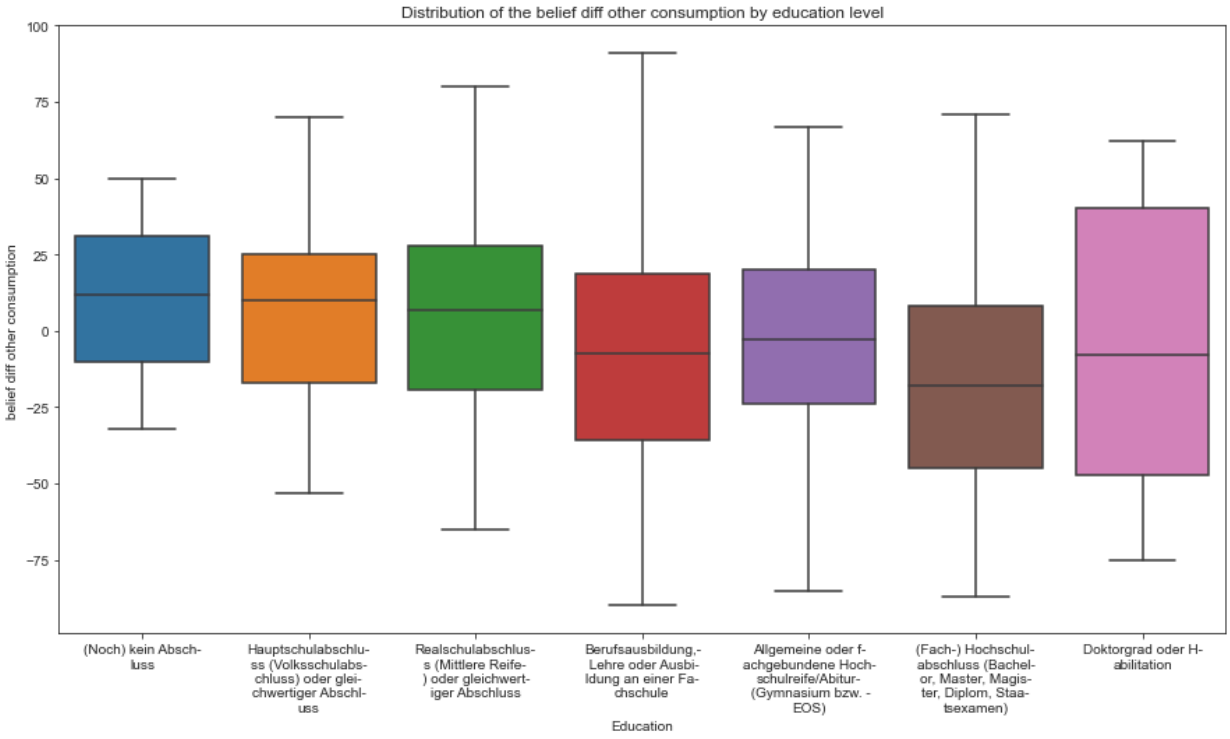

Distribution of the belief diff food by urban-rural classification
belief diff food
100
75
50
25
0
−25
−50
−75
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification

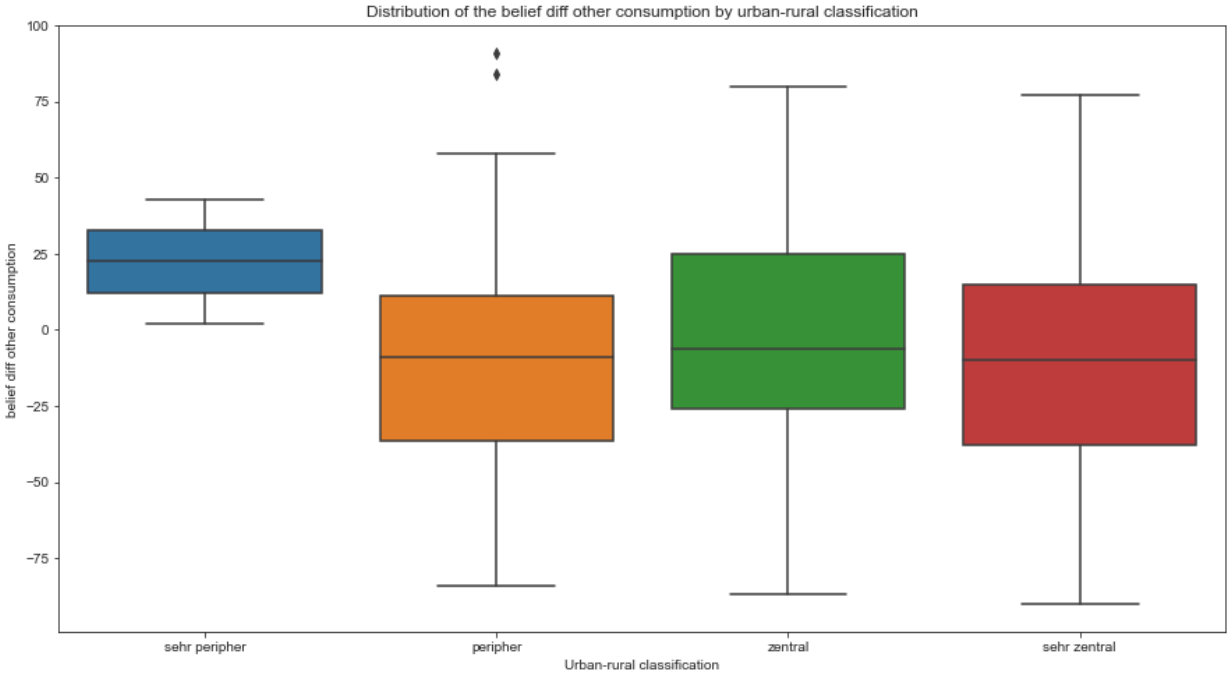

Distribution of the belief diff food by federal states
belief diff food
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Württemberg
Nordrhein--Westfalen
Niedersachsen
Hamburg
Bremen
Rheinland--Pfalz
Thüringen
Schleswig--Holstein
Brandenburg
Mecklenburg-Vorpommern
Federal state

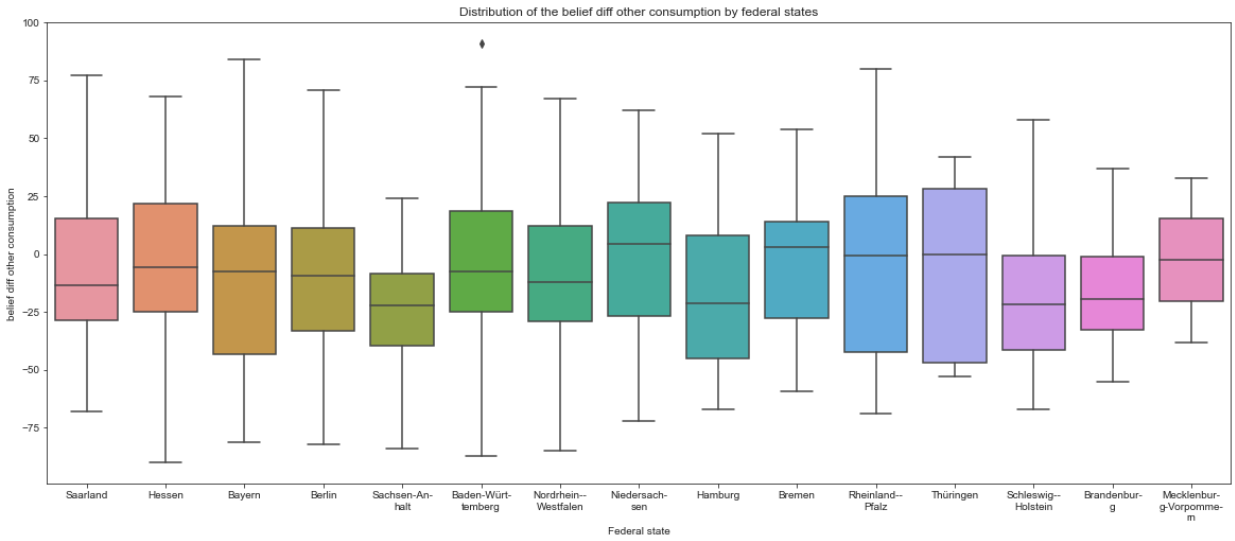

Distribution of the belief diff other consumption by age
belief diff other consumption
Age


Distribution of the belief diff other consumption by income
belief diff other consumption
Income


Distribution of the belief diff other consumption by supporting political party
belief diff other consumption
100
75
50
25
0
−25
−50
−75
CDU/CSU
Keine Angabe
SPD
Einer anderen Partei
FDP
Bündnis 90/Die Grünen
AfD
Die Linke
Bündnis Sarah Wagenknecht
Political party


Distribution of the belief diff other consumption by education level
belief diff other consumption
100
75
50
25
0
−25
−50
−75
(Noch) kein Abschluss
Hauptschulabschluss (Volksschulabschluss) oder gleichwertiger Abschluss
Realschulabschluss (Mittlere Reife) oder gleichwertiger Abschluss
Berufsausbildung,-Lehre oder Ausbildung an einer Fachschule
Allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. -EOS)
(Fach-) Hochschulabschluss (Bachelor, Master, Magister, Diplom, Staatsexamen)
Doktorgrad oder Habilitation
Education


Distribution of the belief diff other consumption by urban-rural classification
belief diff other consumption
100
75
50
25
0
−25
−50
−75
sehr peripher
peripher
zentral
sehr zentral
Urban-rural classification


Distribution of the belief diff other consumption by federal states
belief diff other consumption
100
75
50
25
0
−25
−50
−75
Saarland
Hessen
Bayern
Berlin
Sachsen-An-halt
Baden-Würt-temberg
Nordrhein-- Westfalen
Niedersach-sen
Hamburg
Bremen
Rheinland-- Pfalz
Thüringen
Schleswig-- Holstein
Brandenbur-g
Mecklenbur-g-Vorpomme-rn
Federal state