Pizza vs. Döner

A regional statistical analysis on the Pizza and Döner restaurant distribution in Germany.

Introduction

Background: Both Pizza and Döner Kebab have a high priority in the dietary plan of the average German. However, regions in Germany differ widely in terms of population density, available income, GDP, size, and count of Pizza and Döner places per capita. Especially, some areas appear to prefer one or the other. This analysis shall serve an (imaginary) client as a guide what type of restaurant to open and where – Döner VS Pizza

Problem: A client with Italian and Turkish parents grew up between both food cultures and developed skills beyond equal in preparing both, Pizza and Döner Kebab. Now after finishing School, he is dreaming big and wants to open either, a Pizza or a Döner place. He is willing to move anywhere - but he wants to get rich as fast as possible by stomping the local competition with his god tier cooking skills. He wants to use the money to fulfill his dream of building a spaceship company that bring humanity to Mars and become a Type I civilization, so it is in our interest to help him for free. Also, he is convinced (like anyone with any knowledge) that places who sell both, Pizza and Döner, are a betrayal and insult to either cuisine. This analysis is to help him with the decision where to go and conquer the region with his mouth-watering specialties.

Interest: The Pizza or Döner analysis is important for the client to make a decision where to open what type of restaurant. But to any German statistics to Döner and Pizza and their relation and correlation are of daily relevance have long been overdue.

Data sources

Sources

1. Geo data for all regions in Germany including their geometry coordinates for choropleth plotting in Json format: https://github.com/isellsoap/deutschlandGeoJSON

```
with open('./4_niedrig.geo.json', encoding='utf-8') as f:
    geo_data_germany = json.load(f)

geo_data_germany['features'][1]
```

```
{'type': 'Feature',
 'id': 1,
 'properties': {'ID_0': 86,
  'ISO': 'DEU',
  'NAME_0': 'Germany',
  'ID_1': 9,
  'NAME_1': 'Niedersachsen',
  'ID_2': 23,
  'NAME_2': 'Weser-Ems',
  'ID_3': 245,
  'NAME_3': 'Osnabrück Städte',
  'NL_NAME_3': None,
  'VARNAME_3': None,
  'TYPE_3': 'Kreisfreie Städte',
  'ENGTYPE_3': 'Urban district'},
 'geometry': {'type': 'Polygon',
  'coordinates': [[[7.963789938880371, 52.325450897216],
    [8.011850357055778, 52.34164047241211],
    [8.025308609008789, 52.31085092163086],
    [8.009219932220459, 52.32133580078125],
    [8.093870162963867, 52.31008148193359],
    [8.119648933410701, 52.29108074340832],
    [8.143480711859604, 52.30368041992199],
    [8.181080818176326, 52.28565216064453],
    [8.157988548278809, 52.266639709472271],
    [8.121719360351586, 52.262248992919865],
    [8.135999679565643, 52.232738494873085],
    [8.117508888244743, 52.23612213134771],
    [8.118338584899996, 52.224979400634821],
    [8.088389038175781, 52.242858886718864],
    [8.026668548583984, 52.22676849365234],
    [7.956738948822135, 52.250091552734469],
    [7.967180514132074, 52.273153197765881],
    [7.929549217224178, 52.29069900512695],
    [7.928679943084774, 52.305610656738395],
    [7.988648891449031, 52.307918548584004],
    [7.963789938880371, 52.325450897216]]]}}
```

Figure 1 Json format of the GEO data used.

2. Statistical data from the German ministry for statistics destatis.de: https://www-genesis.destatis.de/genesis/online

```
df_destatis = pd.read_csv("bip_and_avinc_per_inh.csv",sep=";")
df_destatis = df_destatis[:445]
df_destatis = df_destatis.astype({'lfd': 'int32'})
df_destatis.head(5)
```

	lfd	eu	reg	land	nuts1	nuts2	nuts3	Gebietseinheit	ve_pp	bip_pp	Einwohner
0	1	DE1	8.0	BW	1.0	NaN	NaN	Baden-Württemberg	24 892	46 480	11046.0
1	2	DE11	81.0	BW	NaN	2.0	NaN	Stuttgart, Regierungsbezirk	25 569	53 144	4135.0
2	3	DE111	8111.0	BW	NaN	NaN	3.0	Stuttgart, Landeshauptstadt, Stadtkreis	25 788	90 518	634.0
3	4	DE112	8115.0	BW	NaN	NaN	3.0	Böblingen, Landkreis	25 597	66 535	391.0
4	5	DE113	8116.0	BW	NaN	NaN	3.0	Esslingen, Landkreis	25 582	42 726	533.0

Figure 2 Example of the imported data from the German ministry for statistics.

3. Restaurant location and information using the Foursquare API: https://developer.foursquare.com/developer/

```
results = requests.get(url).json()
results
```

```
{'meta': {'code': 200, 'requestId': '5fc2cf9c2e603c2d79484a41'},
 'notifications': [{'type': 'notificationTray', 'item': {'unreadCount': 0}}],
 'response': {'venues': [{'id': '4fa862b3e4b0ebff2f749f06',
    'name': 'Harry's Italian Pizza Bar',
    'location': {'address': '225 Murray St',
     'lat': 40.71521779064671,
     'lng': -74.01473940209351,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.71521779064671,
       'lng': -74.01473940209351},
      {'label': 'entrance', 'lat': 40.715361, 'lng': -74.014975}],
     'distance': 58,
     'postalCode': '10282',
     'cc': 'US',
     'city': 'New York',
     'state': 'NY',
     'country': 'United States',
     'formattedAddress': ['225 Murray St',
      'New York, NY 10282',
      'United States']},
    'categories': [{'id': '4bf58dd8d48988d1ca941735',
      'name': 'Pizza Place',
      'pluralName': 'Pizza Places',
      'shortName': 'Pizza',
      'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/pizza_',
       'suffix': '.png'},
      'primary': True}],
    'referralId': 'v-1606602652',
    'hasPerk': False},
   {'id': '4f3232e219836c91c7bfde94',
    'name': 'Conca Cucina Italian Restaurant',
    'location': {'address': '63 W Broadway',
     'lat': 40.71448400000006,
     'lng': -74.00980600000001,
     'labeledLatLngs': [{'label': 'display',
       'lat': 40.71448400000006,
       'lng': -74.00980600000001}],
```

Figure 3 Example from the Foursquare REST API Json response to a search request.

Cleaning

Merging the Geo data and the statistical data

The regions from the statistical data are not identically numbered nor identically named to the geo data. In 2008 there was a reform where some regions were changed. Functions were implemented to try to match them in two passes: In the first pass identical or near-identical names are matched automatically. In the second pass, the remaining regions are manually re-matched to the new according regions names. A native python dictionary is generated and exported and included as “AreasDestatisMapper.csv”. A main data frame df_data is generated primarily containing the statistical data but IDs are present in the final data frame df_data so that the geo data can be matched and used.

The geo data itself only referenced via ID in the data frame as it would make the data frame too large to handle elegantly.

Adding the required features form the foursquare API

With the existing geo data, request can be made for every region the the results added. In this case, the only required result is the count from the search results for Pizza and Döner respectively. How the geo location and radius required for the Foursquare API search request is generated is described in the Methodology section.

Feature selection

The features used are:

Gebietseinheit: The official name of the area.

ve_pp: The availibe income per person.

bip_pp: The GDP per person.

doener: The count of Doener restaurants

pizza: The count of (real) Pizza restaurants (Italian, Restaurant, Fastfood but not “Döner”, “Turkish”, “Indian” ect. who sell pizza)

centroids: search centroids for the region as explained in the methodology section

radi: the search radius for the according centroid.

doener_norm: normalized count of Döners.

pizza_norm: normalized count of Pizzas

ratio: the Ratio between pizza and Döner restuarants (pizza-döner)/(pizza+döner) scaling form -1(pure döner) to +1(pure pizza).

ratio_norm: the ratio calculated using the normalized counts.

density_100k: the restaurant (döner+pizza) density per 100k inhabitants.

id_json: The id in the json geo data

Methodology

The project was split into 7 steps where the first 4 steps are about data wrangling and the remaining three about generating insights, results, and recommendations.

Step 1: Get the regions in Germany including their Choropleth structure

The geo data on the german regions is in a json file named 4_niedrig.geo.json and was obtained from : https://github.com/isellsoap/deutschlandGeoJSON

The data is loded via the json library and left in memory for the remainder of the analysis under the json object name “geo_data_germany”.

The relevant keys for this analysis are:

Name_3: The region name

Type_3: Is it a region or a large city (sometimes a name represents both)

Geometry: Containing the coordinates for the region polygon for the choropleth map.


```
{'type': 'Feature',
 'id': 1,
 'properties': {'ID_0': 86,
  'ISO': 'DEU',
  'NAME_0': 'Germany',
  'ID_1': 9,
  'NAME_1': 'Niedersachsen',
  'ID_2': 23,
  'NAME_2': 'Weser-Ems',
  'ID_3': 245,
  'NAME_3': 'Osnabrück Städte',
  'NL_NAME_3': None,
  'VARNAME_3': None,
  'TYPE_3': 'Kreisfreie Städte',
  'ENGTYPE_3': 'Urban district'},
 'geometry': {'type': 'Polygon',
  'coordinates': [[[7.963789939880371, 52.3254508972168],
    [8.011850357055778, 52.34164047241211],
    [8.025308609008789, 52.3198509216308],
    [8.09921932220459, 52.321319580078125],
    [8.093870162963867, 52.310081481933594],
    [8.119648933410701, 52.29198074340832],
    [8.149400711059684, 52.30368041992199],
    [8.181080818176326, 52.28565216064453],
    [8.157988548278809, 52.26663970947271],
    [8.121719360351506, 52.262248992919865],
    [8.13599967956543, 52.23273849487305],
    [8.117508888244743, 52.2361221313471],
    [8.11833858489996, 52.22497940063482],
    [8.080390930175781, 52.242858886718864],
    [8.026668548583984, 52.2267684936524],
    [7.956738948822135, 52.25009155273449],
    [7.967100143432674, 52.27315139770508],
    [7.929549217224178, 52.29069900512695],
    [7.928679943084774, 52.305610656738395],
    [7.988648891449031, 52.30791854858404],
    [7.963789939880371, 52.3254508972168]]]}}
```


Figure 4 Sample entry of the geo data json.

Step 2: Get some data on wealth (available income per household and GDP per person) for each region from destatis.de (German statistical institute).

Destatis.de offers an unlimited API. However, the API is cumbersome at best and not worth the time for a simple case like this. Hence, the data is loaded via CSV table “bip_and_avinc_per_inh.csv”. The data is stored in a pandas dataframe “df_destatis”. The CSV is a extraction of the two seemingly most relevant wealth representations, availible income (per person) and GDP (per person). The source tables are “BIP_vgrdl_r2b3_bs2019.xlsx” and “Einkommen_vgrdl_r2b3_bs2019.xlsx” and are included.

	lfd	eu	reg	land	nuts1	nuts2	nuts3	Gebietseinheit	ve_pp	bip_pp	Einwohner
0	1	DE1	8.0	BW	1.0	NaN	NaN	Baden-Württemberg	24 892	46 480	11046.0
1	2	DE11	81.0	BW	NaN	2.0	NaN	Stuttgart, Regierungsbezirk	25 569	53 144	4135.0
2	3	DE111	8111.0	BW	NaN	NaN	3.0	Stuttgart, Landeshauptstadt, Stadtkreis	25 788	90 518	634.0
3	4	DE112	8115.0	BW	NaN	NaN	3.0	Böblingen, Landkreis	25 597	66 535	391.0
4	5	DE113	8116.0	BW	NaN	NaN	3.0	Esslingen, Landkreis	25 582	42 726	533.0

Figure 5 Head of the pandas dataframe containing the statistical data on each region.

Step 3: Merge the Data frames and show available income and GDP on a map

Now comes the realization, that the regions are not identically numbered nor identically named to the geo data. In 2008 there was a reform where some regions were changed. The following functions are to map the ID's of both in two passes in form of a native python dictionary. First, it tries to automatically map the clear ones (identical names) and then in a send pass it requires the uses input to manually search until a match is found. Everything is saved and then loaded into a csv file and proted to a pandas data frame to be able to merge the table with the df_destatis data frame.

The source code for the merging algorithm is included in the ipynb to this project.

	id_json	id_destatis
0	0	262
1	1	253
2	2	263
3	3	264
4	4	265

Figure 6 Sample output of the mapping of the region IDs of the geo data and the statistical data sources.

Step 4: Get the count of Pizza and Döner via Foursquare API

The foursquare API requires a coordinate-set and a radius. The best solution appears to calculate the centroid for each region and the mean radius (centroid to all polygon point geodesic distance) and use this for the search.

However, there is a complication which makes things much more algorithmic and interesting (thank you foursquare): The maximum search results is limited to 50 which, in densely populated regions, is not enough and both Döner or more Pizza would end up getting the count=50.

Hence, we need to split the split the region into sub-regions. A good approximation to do this is to use multiple centroids. Any solution to calculate these (that I could come up with) is iterative. It appears natural to use the K-Means means centroids for this by accessing them via cluster_centers_ list.

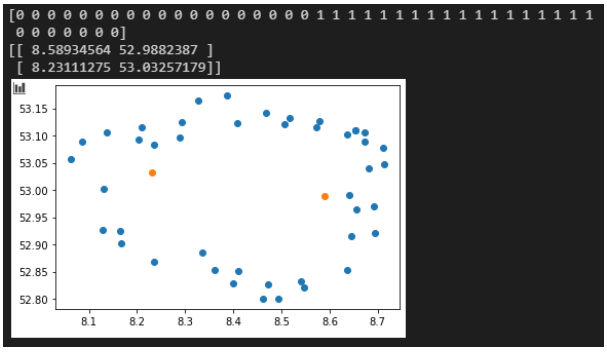


Figure 7 Result of a region centroid and radius calculation with the count of center set to K=2. The data population algorithm with dynamically increase this number until the search result limit is below 50 for both pizza and döner.

There is a bias however towards region with high geometric complexity (many points). This could be reduced by using the high-resolution dataset for the geo data or fractality reduction methods or others. Different algorithms will come up with different solutions but since only the ratio between the two counts really matters for us, there is no reason to find a perfect solution.

Thought: Another method could be to rasterize Germany into slightly overlapping circles so that everything is covered with at least one search request and then push the search results to the according region by writing an algorithm that checks in which polygon the result is in. This would probably require even more requests and much more compute but would be complete and precise where the centroid method is only statistically valid.

Once a function is implemented that can dynamically create as many centroids as required, the main iteration function is implemented that populated the dataframe for each region with the count of the search results for both Pizza and Döner.

The algorithm works the following steps:
1. Goes though all regions

2. Calculates K=1 centroid and checks if it is below the search result limit (<50)
3. if not, repeat step 2 with K += 1
4. Add the cumulative count for Pizza and Döner as well as the centroid position and radii to the list.
5. Go back to step one for next region

The main loop required a sub-function that performs the Foursquare API requests (search endpoint). Here it needs to be mentioned, that this includes a choice and a bias:
Many places that are primarily "Döners" also sell pizza and Foursquare will find them as such.
This is not true the other way round. The question here is for "real" Pizza places. So only places have categories matching a "specialized" pizza place are counted as pizza where every Döner is counted as Döner. However, this reduces the number of places where pizza would be available.

The final valued are collected into the main dataframe pandas object df_data and stored for later use. The requrest time is relatively long because only 500 requrests per hour are possible even with the validated account. The entire filling takes 3 hours.

	Unnamed: 0	1fd	eu	reg	land	nuts1	nuts2	nuts3	Gebietseinheit	ve_pp	...	id_destatis	doener	pizza	centroids	radi
0	0	3	DE111	8111.0	BW	NaN	NaN	3.0	Stuttgart, Landeshauptstadt, Stadtkreis	25788	...	3	46.0	34.0	[[[9.167738959902849, 48.784316289992596]]]	[[8.232163390963873]]
1	1	4	DE112	8115.0	BW	NaN	NaN	3.0	Böblingen, Landkreis	25597	...	4	22.0	30.0	[[[8.948662200064065, 48.672860847329 23]]]	[[14.698258711738177]]
2	2	5	DE113	8116.0	BW	NaN	NaN	3.0	Esslingen, Landkreis	25582	...	5	21.0	33.0	[[[9.360648341800887, 48.636525029721 51]]]	[[14.916208662529932]]
3	3	6	DE114	8117.0	BW	NaN	NaN	3.0	Göppingen, Landkreis	24173	...	6	15.0	13.0	[[[9.725887227058445, 48.682217216491 73]]]	[[14.922800802691 37]]
4	4	7	DE115	8118.0	BW	NaN	NaN	3.0	Ludwigsburg, Landkreis	26203	...	7	27.0	26.0	[[[9.284696318802085, 48.956120518581 02], [8.9...	[[9.164053250083958, 11.021898778933592]]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 8 Result from the Foursquare API requests are the new columns pizza and doener containing the numbers for each. The centroids and radi columns contain the used centroids and radi used for the search and are kept for traceability.

Step 5: Exploring the data

In this step, some basic relational columns are added to the df_data data frame first.

doener_norm: Normalized doener count column (MinMax scaled)

pizza_norm: Normalized Pizza count column (MinMax scaled)

ratio: Ratio between pizza and döner via (pizza-doner)/(pizza+doener)

ratio_norm: Ratio between pizza norm and döner norm via (pizza_norm-doner_norm)/(pizza_norm +doener_norm)

ratio_minimum: calculated ratio but only in cases where both pizza and doener are above a threshold (5) otherwise it sets the ratio 0. This avoids statistics on counts that are so low, that they make little sense.

density_100k: The density of pizza+doener places per 100k inhabitants.

```
df_data.describe()
```

	Unnamed: 0	1fd	reg	nuts1	nuts2	nuts3	ve_pp	bip_pp	Einwohner	id_json	id_destatis	doener	pizza	doener_norm	pizza_norm	ratio	ratio_norm	density_100k	ratio_minimum
count	434.000000	434.000000	434.000000	4.0	5.0	431.0	434.000000	434.000000	4.340000e+02	434.000000	434.000000	434.000000	434.000000	434.000000	434.000000	434.000000	434.000000	434.000000	434.000000
mean	216.500000	231.576037	8694.638249	1.0	2.0	3.0	22473.188940	36741.050691	2.259585e+05	216.500000	231.576037	13.527650	14.525346	0.177995	0.177138	0.062599	0.028927	14.388345	0.026021
std	125.429263	129.174533	4028.388153	0.0	0.0	0.0	2636.773132	16344.668370	2.924998e+05	125.429263	129.174533	13.205619	13.086338	0.173758	0.159589	0.333686	0.334947	9.099369	0.178750
min	0.000000	3.000000	1.000000	1.0	2.0	3.0	16450.000000	16320.000000	3.400000e+04	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	-1.000000	-1.000000	0.421941	-0.600000
25%	108.250000	120.250000	5913.250000	1.0	2.0	3.0	20504.000000	27229.750000	1.110000e+05	108.250000	120.250000	5.000000	5.000000	0.065789	0.060976	-0.151515	-0.188406	7.651001	0.000000
50%	216.500000	230.500000	8419.000000	1.0	2.0	3.0	22385.000000	32420.500000	1.650000e+05	216.500000	230.500000	9.000000	10.000000	0.118421	0.121951	0.056763	0.018829	12.738911	0.000000
75%	324.750000	350.750000	11549.750000	1.0	2.0	3.0	24017.000000	39724.750000	2.535000e+05	324.750000	350.750000	18.000000	20.000000	0.236842	0.243902	0.284286	0.248999	18.912691	0.101531
max	433.000000	444.000000	16077.000000	1.0	2.0	3.0	36883.000000	182301.000000	3.629000e+06	433.000000	444.000000	76.000000	82.000000	1.000000	1.000000	1.000000	1.000000	51.219512	0.538462

Figure 9 Statistical summary over all columns including the newly added relational columns that contain norms, ratios and density values per region.

Step 6: Regression and correlation analysis

The statistical analyses performed are:

1. Correlation Analysis between the technical fields in df_data.
2. Regression Plot between Pizza-Döner-Ratio and average available income of each region.
3. Regression Plot between Pizza-Döner-Ratio and average GDP per Person of each region.
4. Regression Plot between Density of Pizza+Döner restaurants per 100k inhabitants and average available income of each region.
5. High and Low Pizza-to-Doener ratio comparison mapped against the availible income per person in a box plot.
6. High and Low Pizza-to-Doener ratio comparison mapped against the GDP per person in a box plot.
7. High, Normal, Low availible income regions and thier according Pizza-to-Doener Ratio in a box plot.
8. High, Normal, Low GDP regions and their according Pizza-to-Doener Ratio

Step 7: Regional analysis and recommendations

The regio-statistical analyses performed are:

1. Folium Choropleth plot with germany's regions and the Ratio of Doener (-1) to Pizza (+1) as the colored dimension.
2. Folium Choropleth plot with germany's regions and the Pizza+Doener density per 100k inhabitants as the colored dimension.
3. Folium Choropleth plot with germany's regions and the Most Doener-loving regions colored by thier above-average income as the colored dimension.
4. Folium Choropleth plot with germany's regions and the Most Pizza-loving regions colored by thier above-average income as the colored dimension.
5. Folium Choropleth plot with germany's regions and the Best Pizza (red) and best Doener (oragne) regions as the colored dimension.
6. Folium Choropleth plot with germany's regions and the Ratio of Availible Income per Houshold in Euro as the colored dimension.
7. Folium Choropleth plot with germany's regions and the Ratio of GDP Per Person as the colored dimension.
8. Folium Choropleth plot with germany's regions and the Ratio of Inhabitants as the colored dimension.

Results

The results of this analysis are best summarized reading the figures in sequence as they represent the argumentation chain.

The analysis starts with a correlation analysis, continues with a more in-deph regression analysis.

Next, a outlier-sub-goup analysis is performed to understand if there is a hidden correlation with wealth and pizza or döner preference.

This is followed by regional analysis of wealth in general and pizza and döner folium choropleth distribution plots.

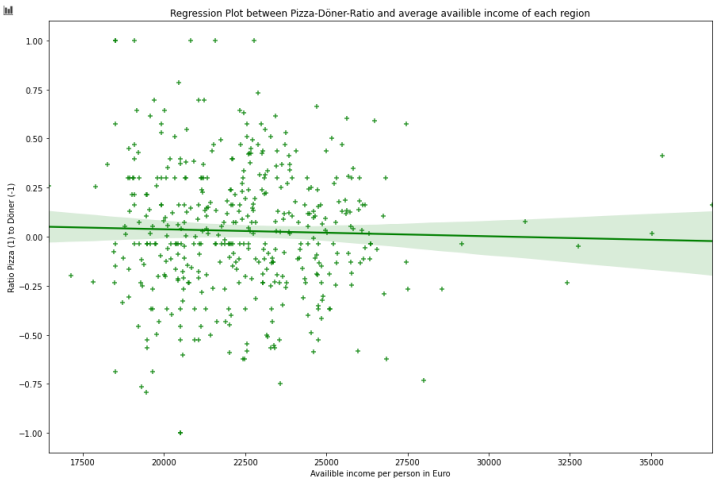
Next, a top 10 candidate list is plotted from which a final recommendation list of regions is derived.

Regression and correlation analysis

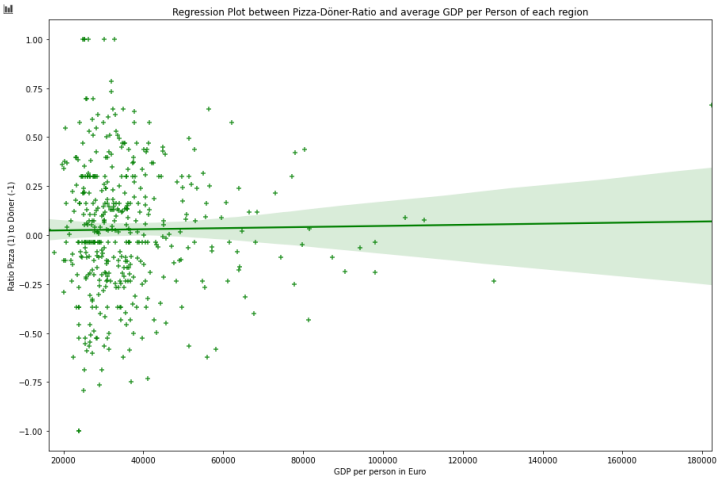
Correlation Analysis

	ve_pp	bip_pp	Einwohner	doener	pizza	ratio	density_100k
ve_pp	1.00	0.30	0.05	0.19	0.18	-0.03	0.19
bip_pp	0.30	1.00	0.09	0.19	0.23	0.02	0.14
Einwohner	0.05	0.09	1.00	0.46	0.51	-0.01	-0.17
doener	0.19	0.19	0.46	1.00	0.81	-0.31	0.51
pizza	0.18	0.23	0.51	0.81	1.00	0.14	0.45
ratio	-0.03	0.02	-0.01	-0.31	0.14	1.00	-0.12
density_100k	0.19	0.14	-0.17	0.51	0.45	-0.12	1.00

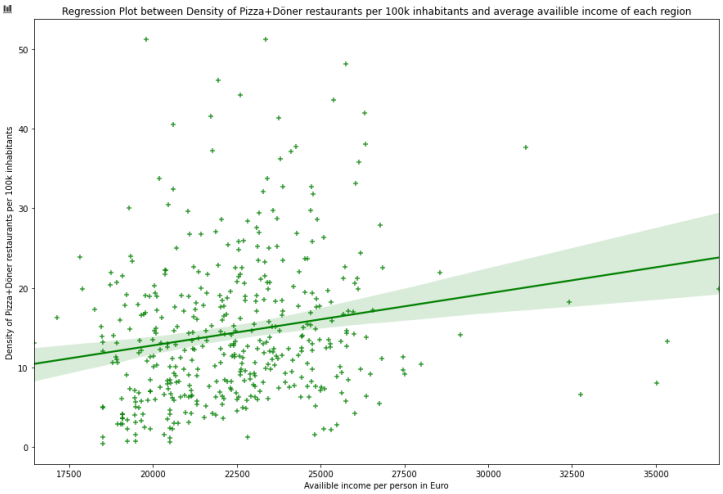
Figure 10 Correlation analysis between obtained valued. The only apparently strong correlations despite pizza and döner count of a region are between values that are either derived from each other.



Tile 1 Regression Plot between Pizza-Döner-Ratio and average available income of each region

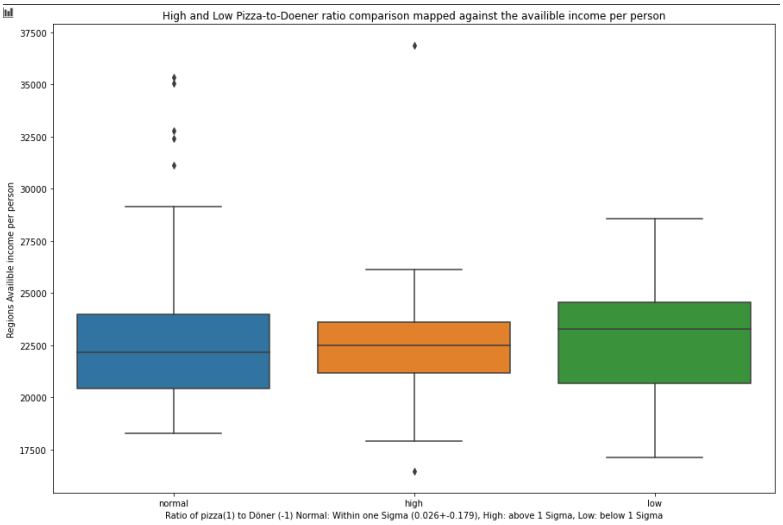


Tile 2 Regression Plot between Pizza-Döner-Ratio and average GDP per Person of each region

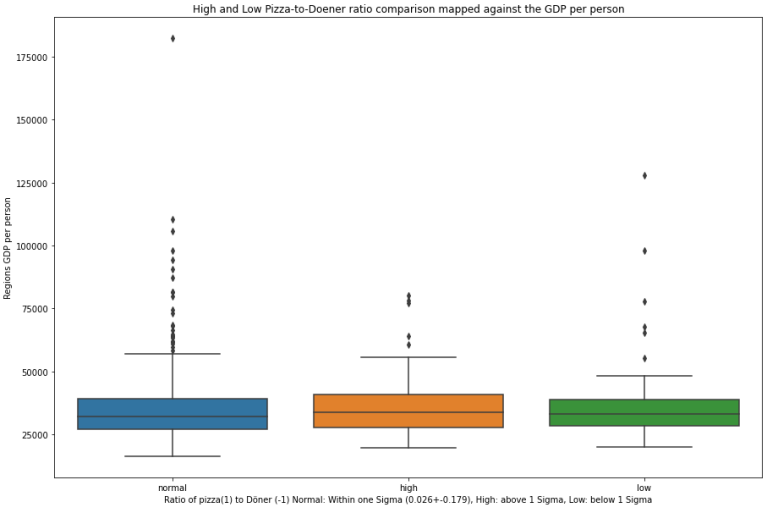


Tile 3 Regression Plot between Density of Pizza+Döner restaurants per 100k inhabitants and average available income of each region

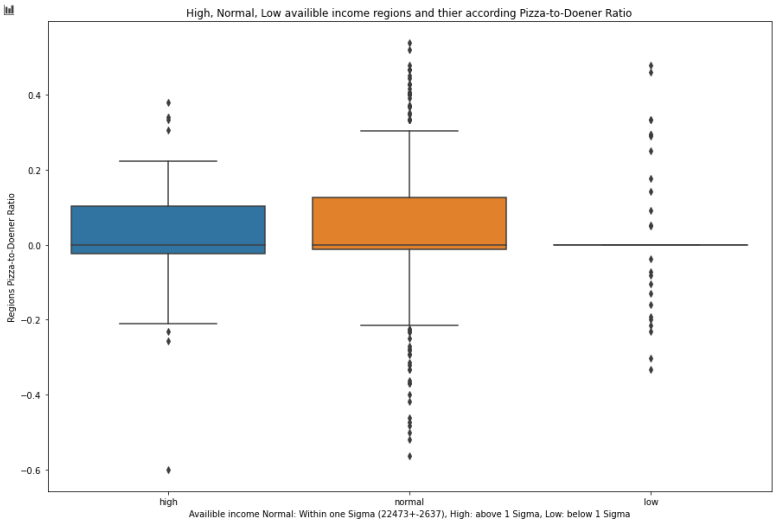
Figure 11: Regression plots showing neither a regression nor any significance (low p-value) for the regressions. In summary this says:
1. There is no over-all positive or negative relationship between the wealth of a region and the ratio of pizza to döner. So rich and poor regions like Pizza and Döner just the same. The long-lasting questions if Pizza or Döner is more classy hence can not be answered with this analysis :(
2. There is a significant (p=7e-5) positive correlation (coef=0.19) between the density of Pizza+Döner restaurants and the available income reaching for ~10 per 100k for poor regions to ~20 per 100k for rich regions. This implies that it is a good idea to choose a region that has high wealth (and ideally not too many restaurants already).



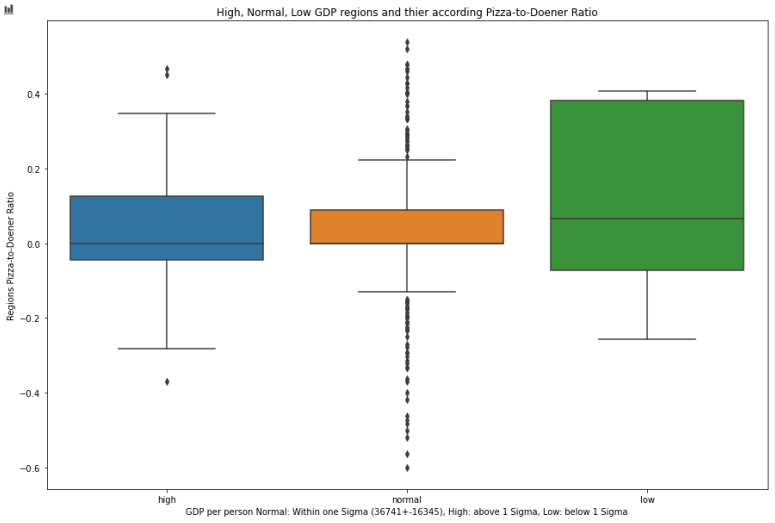
Tile. A High and Low Pizza-to-Doener ratio comparison mapped against the availible income per person.



Tile. B High and Low Pizza-to-Doener ratio comparison mapped against the GDP per person.

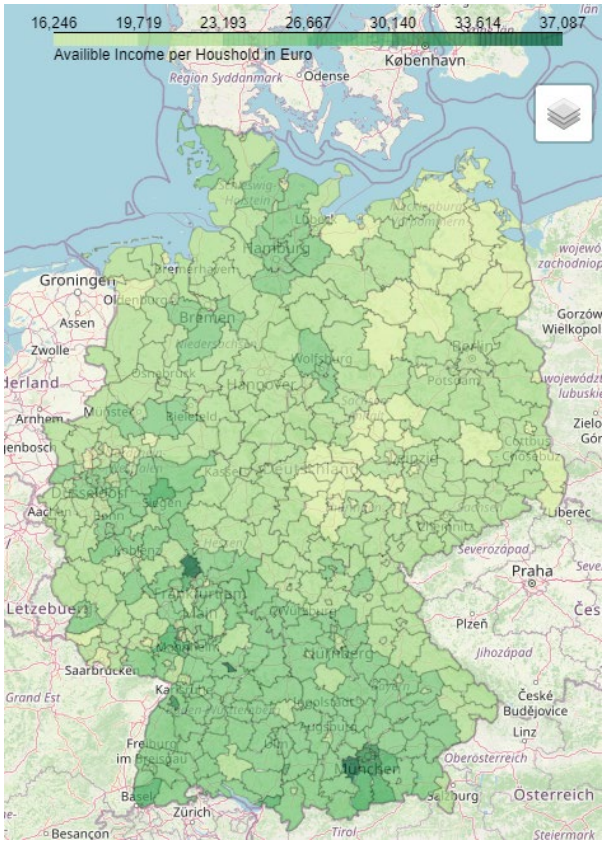


Tile. C High, Normal, Low availible income regions and thier according Pizza-to-Doener Ratio.

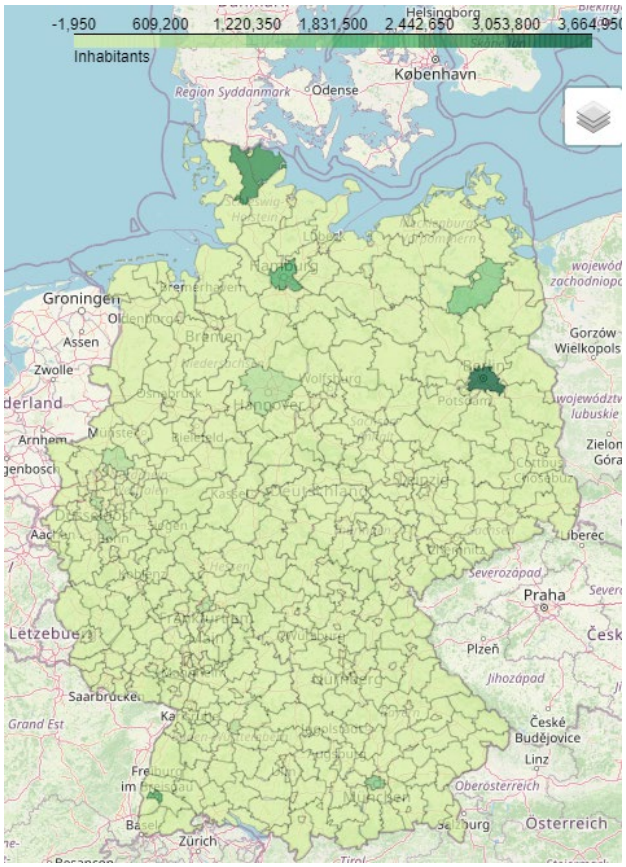


Tile. D High, Normal, Low GDP regions and their according Pizza-to-Doener Ratio.

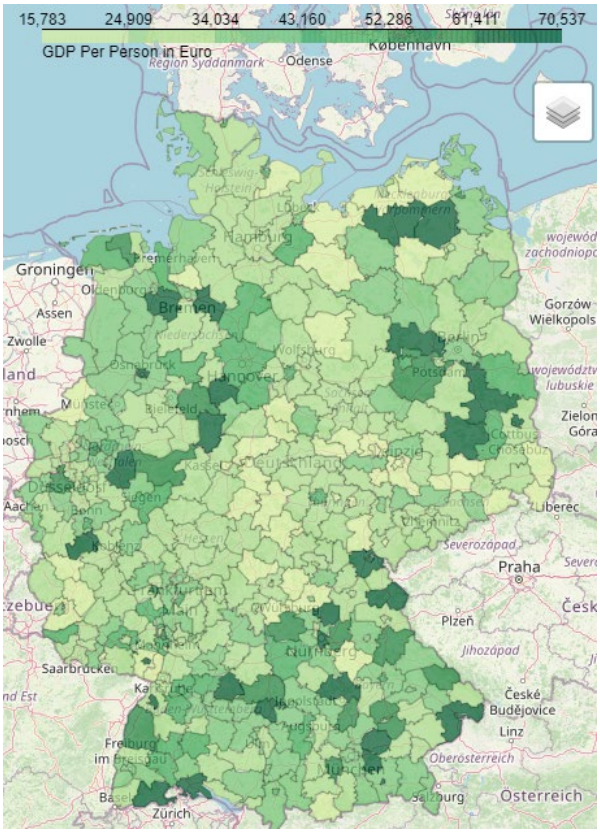
Figure 12: Further analysis of the welch against the Pizza-Döner distribution and vice versa. To make sure that there is no hidden trouth to the saying that Pizza is way more classy than Döner, in this analysis groups are split into upper and lower and middle (normal) groups by means of mean value +- a standard deviation. Tile A and B: The regions split into regions with high, normal and low ratio against their available income (A) and GDP (B) of the region. Tile C and D: The available income (C) and the GDP (D) split into low, normal and high groups against their according pizza-döner-ratio.
In summary, there is no significant difference found in any of the plots again suggesting that Pizza and Döner are equally classy food.



Tile 1 Available income per household in Euro.



Tile 2 Inhabitants of each region.



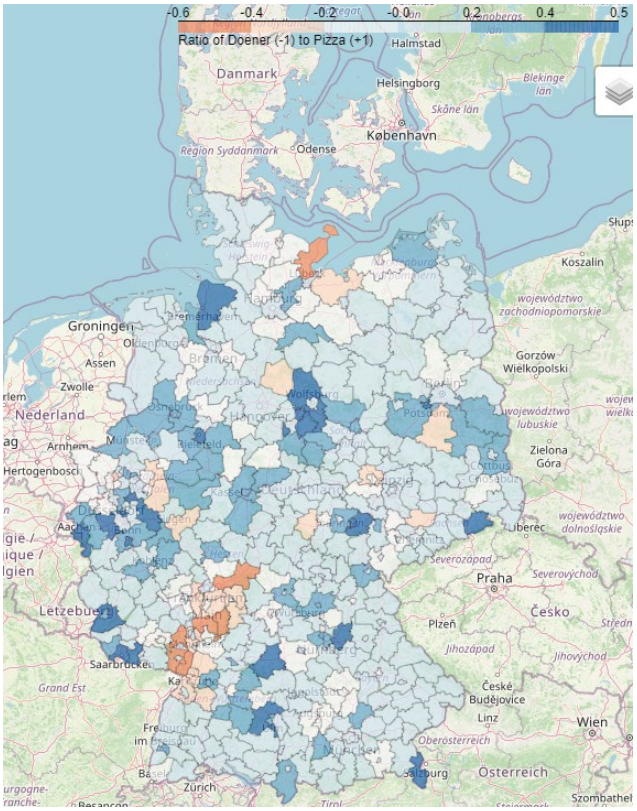
Tile 3 GDP per person.

Figure 13: Wealth and population density information on Germany to allow for a better understand where the wealth and productivity maxima are located.

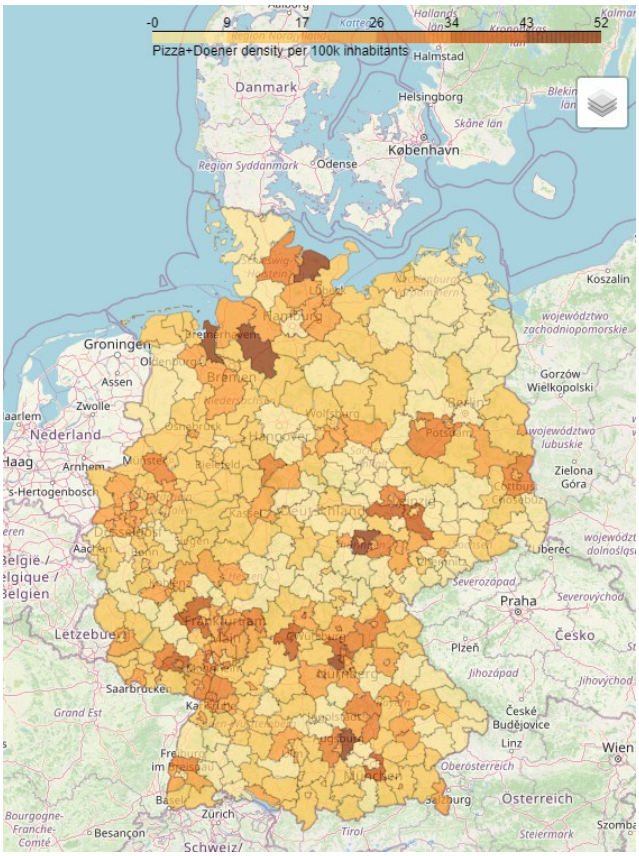
Tile 1: Available income per household in Euro plotted for each region. Clearly visible is the east-west and also the north-south gradient. Simplified, the south and the south-west are wealth regions, but exceptions are areas near major cities like Hamburg.

Tile 2: GDP is dominated by regions with large companies. The plot is a capped plot with 70k € per person so that outlier like Wolfsburg don't dominate the coloring of the plot too much.

Tile 3: Population density distribution shows that massive density peaks in big cities.



Tile 1 Ratio of Doener (-1) to Pizza (+1)



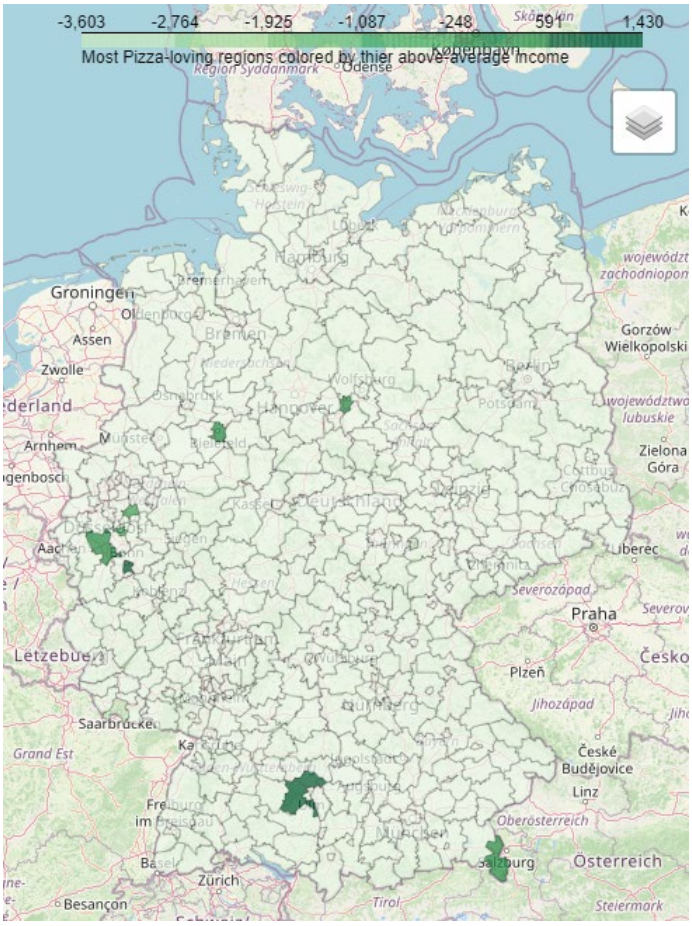
Tile 2 Pizza+Doener density per 100k inhabitants

Figure 14: Showing doener and Pizza statistics on for each region in Germany.

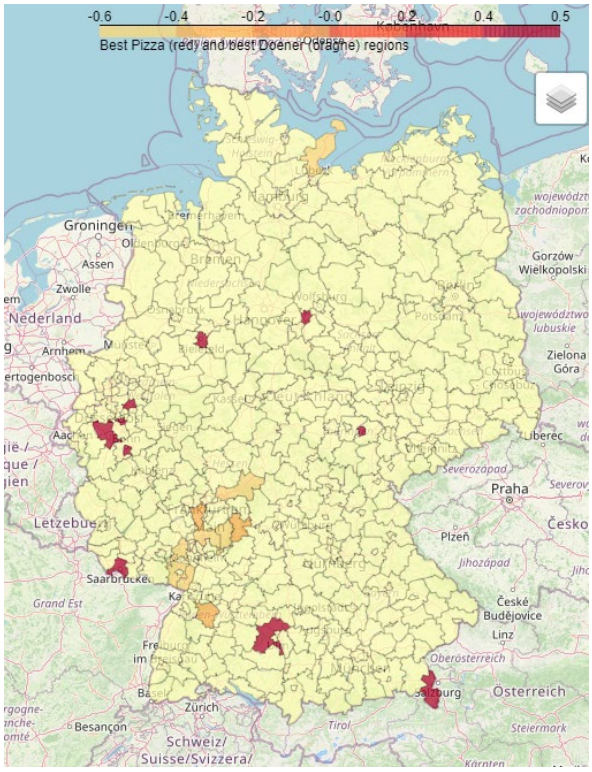
Tile 1: The ratio of döner (pure döner would be minus 1) to pizza (pure pizza would be positive 1). There are regions and clusters with clear dominance of either Pizza or Doener.
It hence would make sense to choose the region accordingly and choose the richest one.

Tile 2: The Pizza+Döner density of each region by means of restaurants per 100k inhabitants. The density differs widely, meaning it is a factor that should be respected in the final choice of the region.

Tile 1 Most Doener-loving colored by their above-average income



Tile 2 Most Pizza-loving regions colored by their above-average income.



Tile 3 Best Pizza (red) and best Doener (oragne) regions.

Figure 15: Sorted by their popularity (ratio excluding regions with statistically insignificant amounts), the top 10 regions for Döner (Tile 1), Pizza (Tile 2) and both (Tile 3) respectively.

The area between Heidelberg and Frankfurt is a heavy döner region where the Ruhr area has a Pizza fetish.

In general, the south and the west are heavily represented which is good news because these are also the richest and most productive areas.

This can be used in the final decision.

	Gebietseinheit	id_json	doener	pizza	ratio_minimum	ve_pp	bip_pp	density_100k	availible_income_rel_to_avg	GDP_rel_to_avg
330	Bad Dürkheim, Landkreis	316	24.0	6.0	-0.600000	26863	22551	22.556391	4390	-14190
169	Main-Kinzig-Kreis	184	25.0	7.0	-0.562500	24607	36532	7.637232	2134	-209
171	Odenwaldkreis	186	19.0	6.0	-0.520000	22564	26668	25.773196	91	-10073
165	Bergstraße, Landkreis	178	39.0	13.0	-0.500000	24752	28488	19.330855	2279	-8253
404	Ostholstein, Kreis	404	20.0	7.0	-0.481481	23179	26919	13.432836	706	-9822
332	Germersheim, Landkreis	319	28.0	10.0	-0.473684	23173	37489	29.457364	700	748
335	Südliche Weinstraße, Landkreis	330	19.0	7.0	-0.461538	24538	27569	23.636364	2065	-9172
167	Groß-Gerau, Landkreis	182	34.0	14.0	-0.416667	22014	45716	17.582418	-459	8975
122	Miltenberg, Landkreis	151	21.0	9.0	-0.400000	23343	34398	23.255814	870	-2343
22	Calw, Landkreis	46	13.0	6.0	-0.368421	24437	29286	12.025316	1964	-7455

Tile 1 The "Döner Kings" - a by dominance sorted list of locations that have a heavy döner tendency including thier wealth data and density.

	Gebietseinheit	id_json	doener	pizza	ratio_minimum	ve_pp	bip_pp	density_100k	availible_income_rel_to_avg	GDP_rel_to_avg
48	Berchtesgadener Land, Landkreis	87	6.0	20.0	0.538462	22547	33571	24.761905	74	-3170
267	Leverkusen, Kreisfreie Stadt	291	6.0	19.0	0.520000	21756	51469	15.243902	-717	14728
269	Rhein-Erft-Kreis	293	6.0	17.0	0.478261	22675	34641	4.904051	202	-2100
414	Jena, Kreisfreie Stadt	420	6.0	17.0	0.478261	18919	44877	20.720721	-3554	8136
265	Bonn, Kreisfreie Stadt	286	12.0	33.0	0.466667	23471	80288	13.803681	998	43547
285	Bielefeld, Kreisfreie Stadt	262	8.0	22.0	0.466667	23037	41016	9.009009	564	4275
259	Wuppertal, Kreisfreie Stadt	283	12.0	33.0	0.466667	21480	36429	12.711864	-993	-312
339	Saarbrücken, Regionalverband	341	17.0	46.0	0.460317	19205	44904	19.090909	-3268	8163
205	Braunschweig, Kreisfreie Stadt	222	11.0	29.0	0.450000	22612	78057	16.129032	139	41316
39	Alb-Donau-Kreis	21	10.0	26.0	0.444444	23854	32093	18.461538	1381	-4648

Tile 2 The "Pizza Kings" - a by dominance sorted list of locations that have a heavy Pizza tendency including thier wealth data and density.

Figure 16: Lists of the most dominant regions for either Döner (Tile 1) or Pizza (Tile 2). Available income (ve_pp) is augmented by the field available income relative to average and the same for GDP to allow for a better understanding if the area is rich or not. The density_100k is another important factor for selecting the favorite.

Recommendations:

1. Since our client does not fear competition (as said his Pizza AND Döner skills are beyond reach), he is best served to go where the people like the according food and have money.

2. The recommendation hence is to open a Döner in "Bad Dürkheim" region where people apparently crave the Döner Kebab and are loaded.

3."Main-Kinzig-Kreis" would be a good Döner alternative, it is not quite as rich but they only have 1/3rd of the restaurant density there compared to "Bad Dürkheim".

4.In case the client last minute decides that Pizza is better (rightfully so), then a good bet would be "Bonn" where the people would sell their grandma for a good pizza; but they don't have to because they have plenty of cash and the productivity of the region is insanely good.

5. "Rhein-Erft-Kreis" would be the alternative for Pizza, due to an severe lack of good restaurants per 100k inhabitants and a high pizza preference and an still average income.

Discussion

This analysis offers relevant insights and business relevant information for the döner and pizza business in Germany.

Despite the extracted recommendation list shown in the result section, a few key facts are interesting.

The density of döner and pizza differs widely in the regions from 1 to above 50 restaurants per 100k inhabitants.

This density is somewhat correlated with the wealth of a region, especially the available income per person. However, still many wealthy areas show a relatively low density of restaurants.

Pizza and Döner preferences and regions clearly exist and they are also clustered around areas.

The most pizza heavy cluster is in the Ruhr region and around Braunschweig.

The most Döner heavy cluster is in the region between Heidelberg and Frankfurt.

Unexpectedly, Pizza did not proof to be the classier food as there is no correlation between wealthy and a pizza preference, even not for the richest areas.

The results have several limitations:

1. The region database is outdated and some groupings to the latest (2018) statistical wealth and population data could only be approximated.
2. The Foursquare Database is not the best, especially not for Germany. Hence, the result count is limited. Whilst probably statistically still valid, it is especially unprecise for regions with only few search results as a heavy bias can happen by coincidence easily.
3. As the Foursquare API only offers search via location and radius, a centroid splitting method was used. This does not ensure 100% coverage of a region. This does both increase the low-numbers issue and reduces the validity of the density score. Both is much more heavily biased by the fact that the Database is much less complete than for example the google meta search. Hence the Method was kept. A slightly more precise method was proposed.
4. A filter choice for pizza was made excluding “combined restaurants” that offer for example döner and pizza. Rightfully so because they rarely serve real pizza.
5. The non-relation between wealth and pizza/döner preference is limited to the region and not a per-person analysis as it would need to be to answer this question. I.e. a rich region still has rich and less rich people and this analysis can not guess who chooses what restaurant. Also, the number of visitors per day can not be extracted from the foursquare API (could be done from google but Foursquare was the requirement for this work).
6. There are more wealth indicators from destais.de that could have been tested for correlations.
7. These recommendations were made under the assumption that a client wants to build into a region with a high preference for this type of food. The opposite to offer the lacking service could also be employed and would require further analysis.

Conclusion

1. There is no regional correlation between wealth and a Döner or Pizza preference.
2. Density of restaurants to inhabitants differs widely but does not correlate with pizza or Döner.
3. Hence there is not clear recommendation for a Pizza or Döner region.
4. There is a correlation between wealth and count of restaurants. Following this logic, it is an important factor to start a restaurant in a wealthy area with a not yet high restaurant density.
5. Döner recommended areas for starting a business are Bad Dürkheim and Main-Kinzig-Kreis due to their heavy Döner preference and good income with moderate restaurant density.
6. Pizza recommended areas are Bonn and Rhein-Erft-Kreis employing the same logic.