

```
In [10]: import pandas as pd
import numpy as np
import altair as alt
```

Top ingredients among different ramen recipes

dataset source: recipenlg.csv provided by professor

```
In [11]: df = pd.read_csv('recipedata/ramen/recipenlg.csv')
df.head()
```

Out[11]:

	Unnamed: 0	title	ingredients	directions	link	source	NER
0	295	Oriental Tossed Salad	["3/4 c. sugar", "1/2 c. red wine vinegar", "1...	["Mix all together right before serving. Serve...	www.cookbooks.com/Recipe-Details.aspx?id=5907	Gathered	["sugar", "red wine vinegar", "oil", "soy sauc...
1	2179	Chinese Cole Slaw	["2 pkg. beef flavor Ramen noodles, broken int...	["Brown noodles, almonds and sesame seeds in t...	www.cookbooks.com/Recipe-Details.aspx?id=141088	Gathered	["beef flavor", "almonds", "sesame seeds", "bu...
2	2273	Ramen Cole Slaw	["1 (1 lb.) bag cole slaw mix", "1/2 bunch gre...	["Mix the oil, vinegar and sugar with the seas...	www.cookbooks.com/Recipe-Details.aspx?id=227363	Gathered	["cole slaw", "green onions", "red pepper", "s...
3	2330	Amber'S Top Ramenlubbock, Texas	["1 pkg. Top Ramen - any flavor cuz I like the...	["Take the package of Top Ramen and smash it t...	www.cookbooks.com/Recipe-Details.aspx?id=1072403	Gathered	["I"]
4	2349	Ramen Cabbage Salad	["1/2 head cabbage, shredded", "4 green onions...	["Combine and mix well."]	www.cookbooks.com/Recipe-Details.aspx?id=1058170	Gathered	["cabbage", "green onions", "chicken flavored ...

```
In [12]: df = df.dropna()
print(len(df))
from collections import defaultdict
dic = defaultdict(int)
for index, row in df.iterrows():
    ls = row['NER'][1:-1].split(',')
    for ing in ls:
        dic[ing[1:-1]] += 1

len(dic)
```

3352

Out[12]: 3540

```
In [13]: ingredients = pd.DataFrame(dic.items())
ingredients.columns = ['ingredient', 'count']
ingredients['count'] = ingredients['count']
top30_ing = ingredients[ingredients['count'] > 164]
top30_ing['type'] = ['flavor', 'flavor', 'flavor', 'noodle', 'topping', 'flavor', 'flavor', 'flav
top30_ing
```

/var/folders/wd/7mmm5v2n75b_cxpswz3tf4100000gn/T/ipykernel_7724/876482172.py:5: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
top30_ing['type'] = ['flavor', 'flavor', 'flavor', 'noodle', 'topping', 'flavor', 'flavor',
'flavor', 'topping', 'flavor', 'flavor', 'flavor', 'topping', 'flavor', 'flavor', 'flavor', 't
opping', 'flavor', 'flavor', 'topping', 'topping', 'soup', 'soup', 'flavor', 'flavor', 'noodl
e', 'flavor', 'flavor', 'topping', 'flavor']
```

Out[13]:

	ingredient	count	type
0	sugar	1078	flavor
2	oil	534	flavor
3	soy sauce	887	flavor
4	Ramen noodles	561	noodle
6	green onion	253	topping
10	almonds	957	flavor
11	sesame seeds	433	flavor
12	butter	317	flavor
13	green onions	1058	topping
15	red pepper	165	flavor
16	sunflower seeds	416	flavor
17	vinegar	341	flavor
19	cabbage	649	topping
22	rice vinegar	213	flavor
23	salt	642	flavor
24	pepper	361	flavor
27	carrots	211	topping
28	sesame seed	217	flavor
38	vegetable oil	463	flavor
49	carrot	180	topping
50	onion	318	topping
52	water	588	soup
54	chicken	371	soup
80	garlic	552	flavor
94	olive oil	221	flavor
100	ramen noodles	1047	noodle
213	sesame oil	379	flavor
221	scallions	210	flavor
277	eggs	215	topping
568	ginger	202	flavor

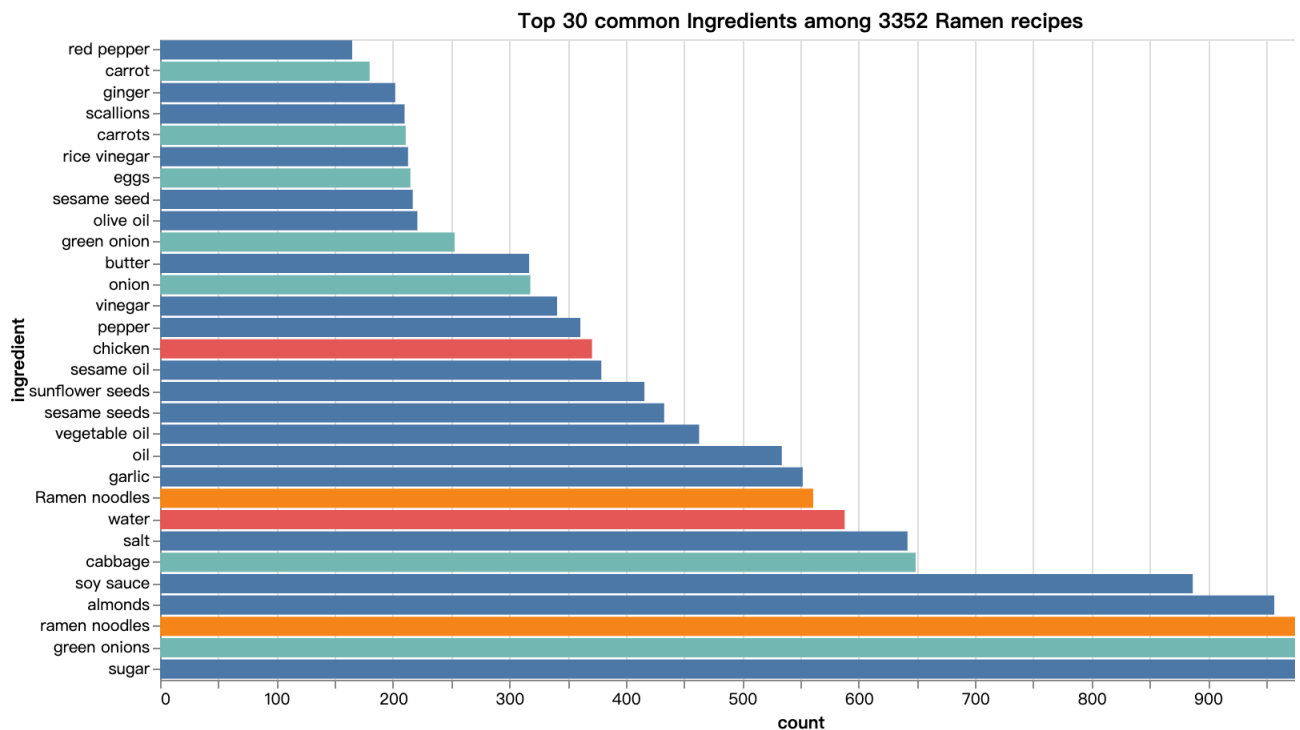
```

In [14]: viz1 = alt.Chart(top30_ing, width=800, height=400).transform_window(
    sort=[alt.SortField('count', order='descending')],
    year_rank='rank(*)'
).transform_filter(
    alt.datum.year_rank <= 30
).mark_bar().encode(
    alt.X('count:Q'),
    alt.Y('ingredient:N', sort=alt.EncodingSortField(
        field='year_rank', order='descending'
    )),
    alt.Color('type:N')
).properties(title="Top 30 common Ingredients among 3352 Ramen recipes")

# viz1.encode(
#     tooltip=['ingredient:N', 'count:Q']
# ).interactive()
viz1

```

Out[14]:



Instant Ramen Population

dataset source: <https://www.kaggle.com/datasets/residentmario/ramen-ratings>
<https://www.kaggle.com/datasets/residentmario/ramen-ratings>

Instant ramen noodles were exported from Japan by Nissin Foods starting in 1971, bearing the name "Oodles of Noodles".[47] One year later, it was re-branded "Nissin Cup Noodles", packaged in a foam food container (It is referred to as Cup Ramen in Japan), and subsequently saw a growth in international sales. Over time, the term "ramen" became used in North America to refer to other instant noodles. While some research has claimed that consuming instant ramen two or more times a week increases the likelihood of developing heart disease and other conditions, including diabetes and stroke, especially in women, those claims have not been reproduced and no study has isolated instant ramen consumption as an aggravating factor.[48][49] However, instant ramen noodles, known to have a serving of 43 g, consist of very high sodium. [50] At least 1,760 mg of sodium are found in one packet alone. It consists of 385k calories, 55.7 g of carbohydrates, 14.5 g of total fat, 6.5 g of saturated fat, 7.9 g of protein, and 0.6 mg of thiamine.[51]

Canned version In Akihabara, Tokyo, vending machines distribute warm ramen in a steel can known as ramen kan (らーめん缶). It is produced by a popular local ramen restaurant in flavors such as tonkotsu and curry, and contains noodles, soup, menma, and pork. It is intended as a quick snack, and includes a small folded plastic fork. [52]

```
In [15]: df = pd.read_csv('instant ramen ratings.csv')
df = df.drop(columns=['T'])
df = df.dropna()[df["Country"] == "Japan"]
df
# len(pd.unique(df['Brand'])) # 105 brands
# len(pd.unique(df['Style'])) # 5 styles
# len(pd.unique(df['Country'])) # 1 countries
```

Out[15]:

	Review #	Brand	Variety	Style	Country	Stars
7	4113	Sapporo Ichiban	Pokemon Noodle Salt & Seafood Flavor	Cup	Japan	2.75
10	4110	Acecook	Hello Kitty Kawaii Tonkotsu Noodle	Cup	Japan	3.75
12	4108	Nissin	Menya Itto Ramen	Bowl	Japan	4.5
13	4107	Myojo	Bubuka Style Aburasoba 5th Level Spicy	Tray	Japan	2.5
21	4099	Acecook	Nomihosu Ippai Miso Butter Ramen	Cup	Japan	2.25
...
4094	26	Nissin	Soba Noodles With Mayo Mustard Squirt	Pack	Japan	4
4095	25	Nissin	Demae Ramen Spicy Flavor	Pack	Japan	3
4096	24	Nissin	Demae Ramen Spicy Seafood With Chili Pepper	Pack	Japan	2
4107	13	Sapporo Ichiban	Chow Mein	Pack	Japan	5
4108	12	Sapporo Ichiban	Shrimp Flavor	Pack	Japan	2.5

800 rows x 6 columns

```

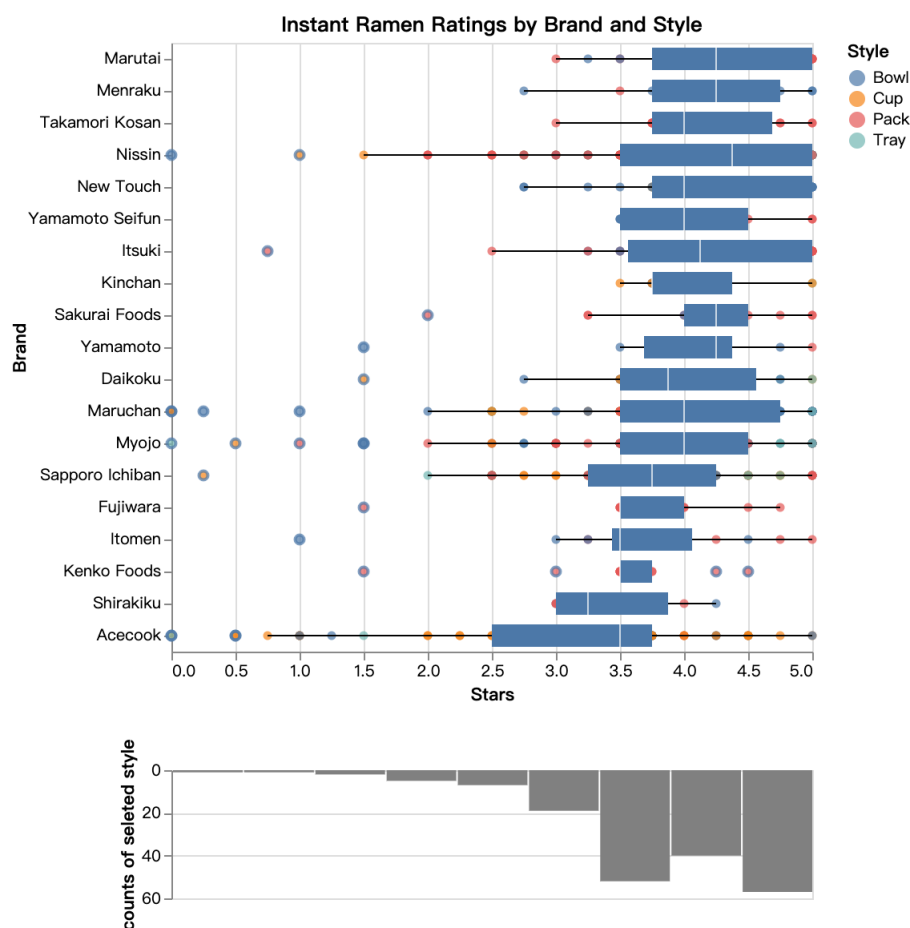
In [16]: base = alt.Chart(df).transform_joinaggregate(
    groupby=['Brand'],
    avg_star = 'average(Stars)',
    count='count(Stars)'
).transform_filter(
    alt.datum.count > 5)

# dot_chart = base.mark_circle().encode(
#     alt.X('Brand:N', sort=alt.EncodingSortField(
#         field='avg_star', order='descending'
#     )),
#     alt.Y('avg_star:Q')
# )
# dot_chart

vizcircle = base.mark_circle().encode(
    alt.X('Stars:Q'),
    alt.Y('Brand:N', sort=alt.EncodingSortField(
        field='avg_star', order='descending'
    )),
    alt.Color('Style:N'))
vizboxplot = base.mark_boxplot().encode(
    alt.X('Stars:Q'),
    alt.Y('Brand:N', sort=alt.EncodingSortField(
        field='avg_star', order='descending'
    )))
viz_bystyle = base.mark_bar(color='grey').transform_filter(alt.datum.Style == 'Pack').encode(
    alt.X('Stars:Q', bin=alt.Bin(maxbins=10), title=None, axis=None),
    alt.Y('count(Stars):Q', scale=alt.Scale(reverse=True), title="counts of seleted style")).prop
viz_bybrand = (vizcircle + vizboxplot).properties(title="Instant Ramen Ratings by Brand and Style")
viz2 = (viz_bybrand & viz_bystyle)
viz2

```

Out[16]:



Regional Variation of Ramen

detect regional popularity of ramen variation using google trend

```
In [17]: df = pd.read_csv('geoMap.csv')
df.head()
```

Out[17]:

	region	oil_soba	kikata_ramen	tsuke_men
0	福島縣	7%	73%	20%
1	埼玉縣	21%	7%	72%
2	東京都	26%	7%	67%
3	宮城縣	40%	13%	47%
4	栃木縣	16%	20%	64%

```
In [18]: for index, row in df.iterrows():
    row['oil_soba'] = str(row['oil_soba'])[:-1]
    row['kikata_ramen'] = str(row['kikata_ramen'])[:-1]
    row['tsuke_men'] = str(row['tsuke_men'])[:-1]

df_long = df.melt('region', var_name='Soba', value_name='percentage')
df_long
```

Out[18]:

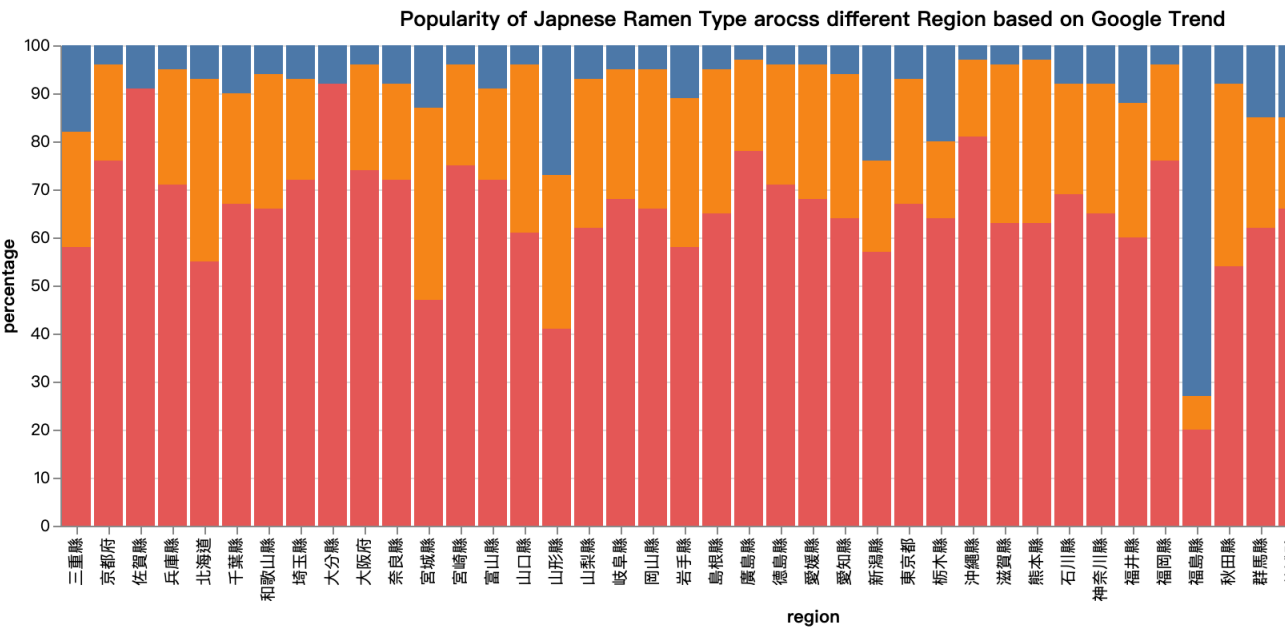
	region	Soba	percentage
0	福島縣	oil_soba	7
1	埼玉縣	oil_soba	21
2	東京都	oil_soba	26
3	宮城縣	oil_soba	40
4	栃木縣	oil_soba	16
...
136	福井縣	tsuke_men	60
137	島根縣	tsuke_men	65
138	佐賀縣	tsuke_men	91
139	長崎縣	tsuke_men	92
140	鹿兒島縣	tsuke_men	49

141 rows x 3 columns

In [19]:

```
bar = alt.Chart(df_long).mark_bar().encode(
    alt.X('region:N'),
    alt.Y('percentage:Q'),
    color=alt.Color('Soba:N', legend=alt.Legend(title="Ramen Type"))
).properties(title="Popularity of Japanese Ramen Type arocss different Region based on Google Tren
bar
```

Out[19]:



In []:

```
viz3.
```