

INDIAN FOOD - DATA ANALYSIS AND VISUALIZATION



About the dataset :

The dataset contains information about various types of food , ingredients used , diet type , preparation time , cook time , flavour of food , course , state and region.

The Dataset Contains : 255 rows and 9 columns. Region column have 1 missing values and some of the columns have -1 as their value . After cleaning the data we get 180 rows and 9 columns. 75 Rows have missing and inappropriate value so here i have removed it.

Contents :

[Import required libraries]

[Import the dataset]

[Data Exploration]

[Data Cleaning]

[Data Analysis and Visualization]

Import the required libraries

In [1]:

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from wordcloud import WordCloud
```

Import the dataset

In [2]:

```
df = pd.read_csv('https://raw.githubusercontent.com/Somali19/Indian-food-data-analysis/main/indian_food.csv', na
df
```

Out[2]:

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	r
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	
3	Ghevar	Flour, ghee, kewra, milk, clarified butter, su...	vegetarian	15	30	sweet	dessert	Rajasthan	
4	Gulab jamun	Milk powder, plain flour, baking powder, ghee,...	vegetarian	15	40	sweet	dessert	West Bengal	
...	
250	Til Pitha	Glutinous rice, black sesame seeds, gur	vegetarian	5	30	sweet	dessert	Assam	
251	Bebinca	Coconut milk, egg yolks, clarified butter, all...	vegetarian	20	60	sweet	dessert	Goa	
252	Shufta	Cottage cheese, dry dates, dried rose petals, ...	vegetarian	-1	-1	sweet	dessert	Jammu & Kashmir	
253	Mawa Bati	Milk powder, dry fruits, arrowroot powder, all...	vegetarian	20	45	sweet	dessert	Madhya Pradesh	C
254	Pinaca	Brown rice, fennel seeds, grated coconut, blac...	vegetarian	-1	-1	sweet	dessert	Goa	

255 rows × 9 columns



DATA EXPLORATION

In [3]:

```
df.shape
```

Out[3]:

```
(255, 9)
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255 entries, 0 to 254
Data columns (total 9 columns):
name          255 non-null object
ingredients    255 non-null object
diet          255 non-null object
prep_time     255 non-null int64
cook_time     255 non-null int64
flavor_profile 255 non-null object
course        255 non-null object
state         255 non-null object
region        254 non-null object
dtypes: int64(2), object(7)
memory usage: 18.1+ KB
```

In [5]:

```
df.region.unique()
```

Out[5]:

```
array(['East', 'West', 'North', '-1', 'North East', 'South', 'Central',
      nan], dtype=object)
```

In [6]:

```
df.flavor_profile.unique()
```

Out[6]:

```
array(['sweet', 'spicy', 'bitter', '-1', 'sour'], dtype=object)
```

In [7]:

```
df.state.unique()
```

Out[7]:

```
array(['West Bengal', 'Rajasthan', 'Punjab', 'Uttar Pradesh', '-1',  
      'Odisha', 'Maharashtra', 'Uttarakhand', 'Assam', 'Bihar',  
      'Andhra Pradesh', 'Karnataka', 'Telangana', 'Kerala', 'Tamil Nadu',  
      'Gujarat', 'Tripura', 'Manipur', 'Nagaland', 'NCT of Delhi',  
      'Jammu & Kashmir', 'Chhattisgarh', 'Haryana', 'Madhya Pradesh',  
      'Goa'], dtype=object)
```

In [8]:

```
df.name.unique()
```

Out[8]:

```
array(['Balu shahi', 'Boondi', 'Gajar ka halwa', 'Ghevar', 'Gulab jamun',  
      'Imarti', 'Jalebi', 'Kaju katli', 'Kalakand', 'Kheer', 'Laddu',  
      'Lassi', 'Nankhatai', 'Petha', 'Phirni', 'Rabri', 'Sheera',  
      'Singori', 'Sohan halwa', 'Sohan papdi', 'Chhena jalebi',  
      'Chhena kheeri', 'Chhena poda', 'Cham cham', 'Kheer sagar',  
      'Ledikeni', 'Lyangcha', 'Malapua', 'Mihidana', 'Misti doi',  
      'Pantua', 'Pithe', 'Rasabali', 'Ras malai', 'Rasgulla', 'Sandesh',  
      'Adhirasam', 'Ariselu', 'Bandar laddu', 'Chikki', 'Dharwad pedha',  
      'Double ka meetha', 'Gavvalu', 'Kakinada khaja', 'Kuzhi paniyaram',  
      'Mysore pak', 'Obbattu holige', 'Palathalikalu', 'Poornalu',  
      'Pongal', 'Pootharekulu', 'Qubani ka meetha', 'Sheer korma',  
      'Unni Appam', 'Kajjikaya', 'Anarsa', 'Basundi', 'Dhondas',  
      'Doodhpak', 'Mahim halwa', 'Modak', 'Shankarpali', 'Shrikhand',  
      'Sutar feni', 'Maach Jhol', 'Pork Bharta', 'Chak Hao Kheer',  
      'Galho', 'Aloo gobi', 'Aloo tikki', 'Aloo matar', 'Aloo methi',  
      'Aloo shimla mirch', 'Bhaturo', 'Bhindi masala', 'Biryani',  
      'Butter chicken', 'Chana masala', 'Chapati', 'Chicken razala',  
      'Chicken Tikka masala', 'Chicken Tikka', 'Chole bhature',  
      'Daal baati churma', 'Daal puri', 'Dal makhani', 'Dal tadka',  
      'Dum aloo', 'Poha', 'Fara', 'Kachori', 'Kadai paneer',  
      'Kadhi pakoda', 'Karela bharta', 'Khichdi', 'Kofta',  
      'Kulfi falooda', 'Lauki ke kofte', 'Lauki ki subji',  
      'Litti chokha', 'Makki di roti sarson da saag', 'Misi roti',  
      'Mushroom do pyaza', 'Mushroom matar', 'Naan', 'Navrattan korma',  
      'Palak paneer', 'Paneer butter masala', 'Paneer tikka masala',  
      'Pani puri', 'Panjeeri', 'Papad', 'Paratha', 'Pattor',  
      'Pindi chana', 'Rajma chawal', 'Rongi', 'Samosa', 'Sattu ki roti',  
      'Shahi paneer', 'Shahi tukra', 'Vegetable jalfrezi',  
      'Tandoori Chicken', 'Tandoori Fish Tikka', 'Attu', 'Avial',  
      'Bisi bele bath', 'Curryupillai sadam', 'Dosa', 'Idiappam',  
      'Idli', 'Kanji', 'Kaara kozhambu', 'Keerai kootu',  
      'Keerai masiyal', 'Keerai sadam', 'Keerai poriyal', 'Beef Fry',  
      'Kootu', 'Kos kootu', 'Koshimbri', 'Kothamali sadam',  
      'Kuzhakkattai', 'Kuzhambu', 'Masala Dosa', 'Pachadi', 'Paniyaram',  
      'Papadam', 'Paravannam', 'Payasam', 'Paruppu sadam', 'Pesarattu',  
      'Poriyal', 'Puli sadam', 'Rasam', 'Puttu', 'Sambar', 'Sandige',  
      'Sevai', 'Thayir sadam', 'Theeyal', 'Uttapam', 'Vada',  
      'Chicken Varuval', 'Upma', 'Amti', 'Zunka', 'Kolim Jawla', 'Saath',  
      'Bajri no rotlo', 'Coconut vadi', 'Bhakri', 'Bombil fry',  
      'Chakali', 'Chevdo', 'Chorafali', 'Copra paak', 'Daal Dhokli',  
      'Kutchi dabeli', 'Dahi vada', 'Dalithoy', 'Dhokla', 'Dudhi halwa',  
      'Gatta curry', 'Gud papdi', 'Ghooghra', 'Handwo', 'Halvasan',  
      'Jeera Aloo', 'Kansar', 'Keri no ras', 'Khakhra', 'Khandvi',  
      'Kombdi vade', 'Laapsi', 'Koshimbir', 'Methi na Gota', 'Mohanthal',  
      'Muthiya', 'Patra', 'Pav Bhaji', 'Puri Bhaji', 'Sabudana Khichadi',  
      'Sev khamani', 'Sev tameta', 'Namakpara', 'Sukhdi', 'Surnoli',  
      'Thalipeeth', 'Undhiyu', 'Veg Kolhapuri', 'Vindaloo',  
      'Lilva Kachori', 'Mag Dhokli', 'Khichu', 'Thepla', 'Farsi Puri',  
      'Khaman', 'Turiya Patra Vatana sabji', 'Churma Ladoo',  
      'Cheera Doi', 'Gheela Pitha', 'Khar', 'Kumol Sawul', 'Luchi',  
      'Alu Pitika', 'Masor tenga', 'Bengena Pitika', 'Bilahi Maas',  
      'Black rice', 'Bora Sawul', 'Brown Rice', 'Chingri malai curry',  
      'Goja', 'Hando Guri', 'Haq Maas', 'Chingri Bhape', 'Kabiraji',  
      'Khorisa', 'Koldil Chicken', 'Konir Dom', 'Koldil Duck',  
      'Masor Koni', 'Mishti Chholar Dal', 'Pakhala', 'Pani Pitha',
```

'Payokh', 'Prawn malai curry', 'Red Rice', 'Shukto', 'Til Pitha',
'Bebinca', 'Shufta', 'Mawa Bati', 'Pinaca'], dtype=object)

In [9]:

```
df.head()
```

Out[9]:

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	region
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	East
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	West
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	North
3	Ghevar	Flour, ghee, kewra, milk, clarified butter, su...	vegetarian	15	30	sweet	dessert	Rajasthan	West
4	Gulab jamun	Milk powder, plain flour, baking powder, ghee,...	vegetarian	15	40	sweet	dessert	West Bengal	East

In [10]:

```
df.tail()
```

Out[10]:

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	req
250	Til Pitha	Glutinous rice, black sesame seeds, gur	vegetarian	5	30	sweet	dessert	Assam	N
251	Bebinca	Coconut milk, egg yolks, clarified butter, all...	vegetarian	20	60	sweet	dessert	Goa	V
252	Shufta	Cottage cheese, dry dates, dried rose petals, ...	vegetarian	-1	-1	sweet	dessert	Jammu & Kashmir	N
253	Mawa Bati	Milk powder, dry fruits, arrowroot powder, all...	vegetarian	20	45	sweet	dessert	Madhya Pradesh	Ce
254	Pinaca	Brown rice, fennel seeds, grated coconut, blac...	vegetarian	-1	-1	sweet	dessert	Goa	V

In [11]:

```
df.describe()
```

Out[11]:

	prep_time	cook_time
count	255.000000	255.000000
mean	31.105882	34.529412
std	72.554409	48.265650
min	-1.000000	-1.000000
25%	10.000000	20.000000
50%	10.000000	30.000000
75%	20.000000	40.000000
max	500.000000	720.000000

In [12]:

```
df.corr()
```

Out[12]:

	prep_time	cook_time
prep_time	1.000000	0.144113
cook_time	0.144113	1.000000

DATA CLEANING

In [13]:

```
df.duplicated().sum()
```

Out[13]:

0

In [14]:

```
df.isnull().sum()
```

Out[14]:

```
name          0
ingredients    0
diet          0
prep_time     0
cook_time     0
flavor_profile 0
course        0
state         0
region        1
dtype: int64
```

In [15]:

```
df.dropna(inplace=True)
```

In [16]:

```
df.isnull().sum()
```

Out[16]:

```
name          0
ingredients    0
diet          0
prep_time     0
cook_time     0
flavor_profile 0
course        0
state         0
region        0
dtype: int64
```

In [17]:

```
df.shape
```

Out[17]:

```
(254, 9)
```

There are still some unwanted values in our data we have to clean the data again.

Some of the rows contain -1 as their value we need to remove it.

In [18]:

```
df=df[df['prep_time'] >= 1]
df=df[df['cook_time'] >= 1]
df=df[df['flavor_profile'] != '-1']
df=df[df['course'] != '-1']
df=df[df['state'] != '-1']
df=df[df['region'] != '-1']
df
```

Out[18]:

	name	ingredients	diet	prep_time	cook_time	flavor_profile	course	state	r
0	Balu shahi	Maida flour, yogurt, oil, sugar	vegetarian	45	25	sweet	dessert	West Bengal	
1	Boondi	Gram flour, ghee, sugar	vegetarian	80	30	sweet	dessert	Rajasthan	
2	Gajar ka halwa	Carrots, milk, sugar, ghee, cashews, raisins	vegetarian	15	60	sweet	dessert	Punjab	
3	Ghevar	Flour, ghee, kewra, milk, clarified butter, su...	vegetarian	15	30	sweet	dessert	Rajasthan	
4	Gulab jamun	Milk powder, plain flour, baking powder, ghee,...	vegetarian	15	40	sweet	dessert	West Bengal	
...	
247	Prawn malai curry	Coconut milk, prawns, garlic, turmeric, sugar	non vegetarian	15	50	spicy	main course	West Bengal	
249	Shukto	Green beans, bitter gourd, ridge gourd, banana...	vegetarian	10	20	spicy	main course	West Bengal	
250	Til Pitha	Glutinous rice, black sesame seeds, gur	vegetarian	5	30	sweet	dessert	Assam	
251	Bebinca	Coconut milk, egg yolks, clarified butter, all...	vegetarian	20	60	sweet	dessert	Goa	
253	Mawa Bati	Milk powder, dry fruits, arrowroot powder, all...	vegetarian	20	45	sweet	dessert	Madhya Pradesh	C

180 rows × 9 columns



Now the data is cleaned and ready to be used for analysis.

DATA ANALYSIS AND VISUALIZATION

In [19]:

```
import plotly.graph_objects as go
```

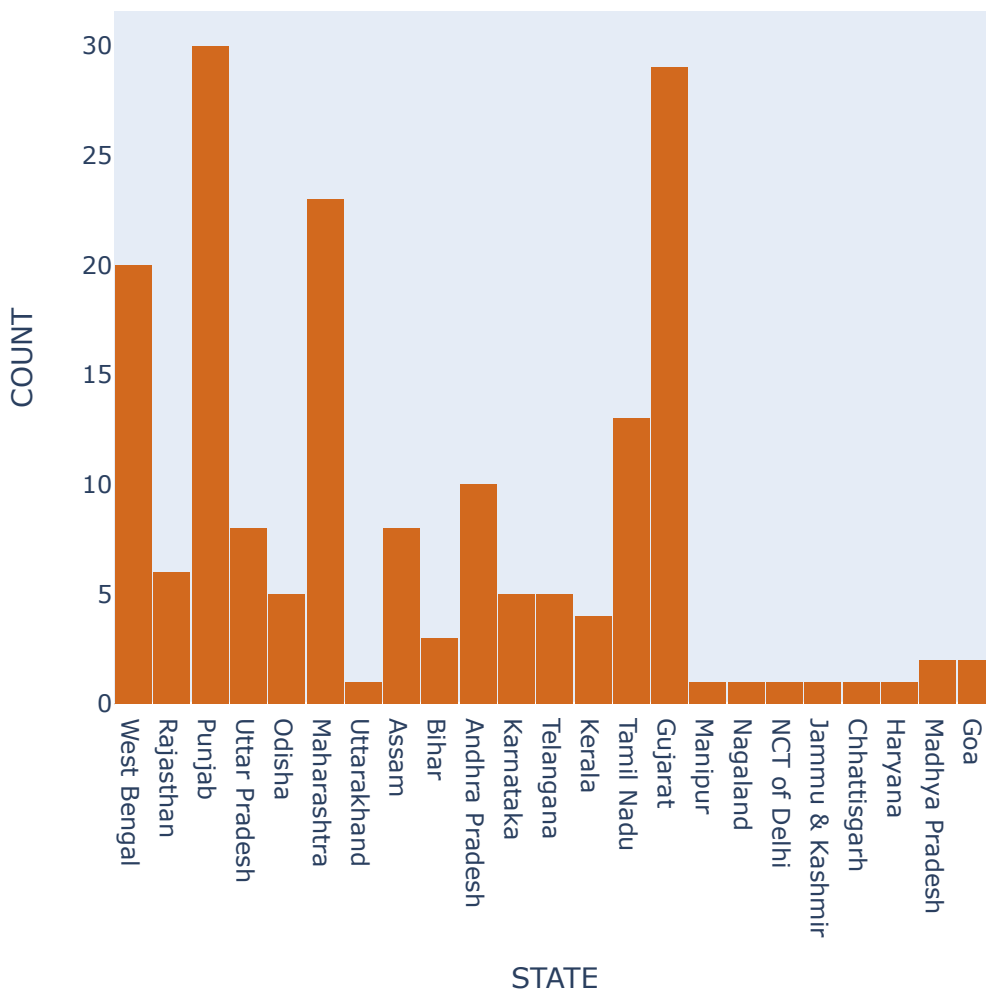
In [20]:

```
fig = go.Figure()
fig.add_trace(go.Histogram(
    x = df['state'],
    marker_color='chocolate',
    opacity=1
))

fig.update_layout(
    title_text='STATE_DISTRIBUTION',
    xaxis_title_text='STATE',
    yaxis_title_text='COUNT',
    bargap=0.05,
    xaxis = {'showgrid': False },
    yaxis = {'showgrid': False },
    width=600,
    height=600
)

fig.show()
```

STATE_DISTRIBUTION



The above graph shows the distribution of Number of dishes over the state.

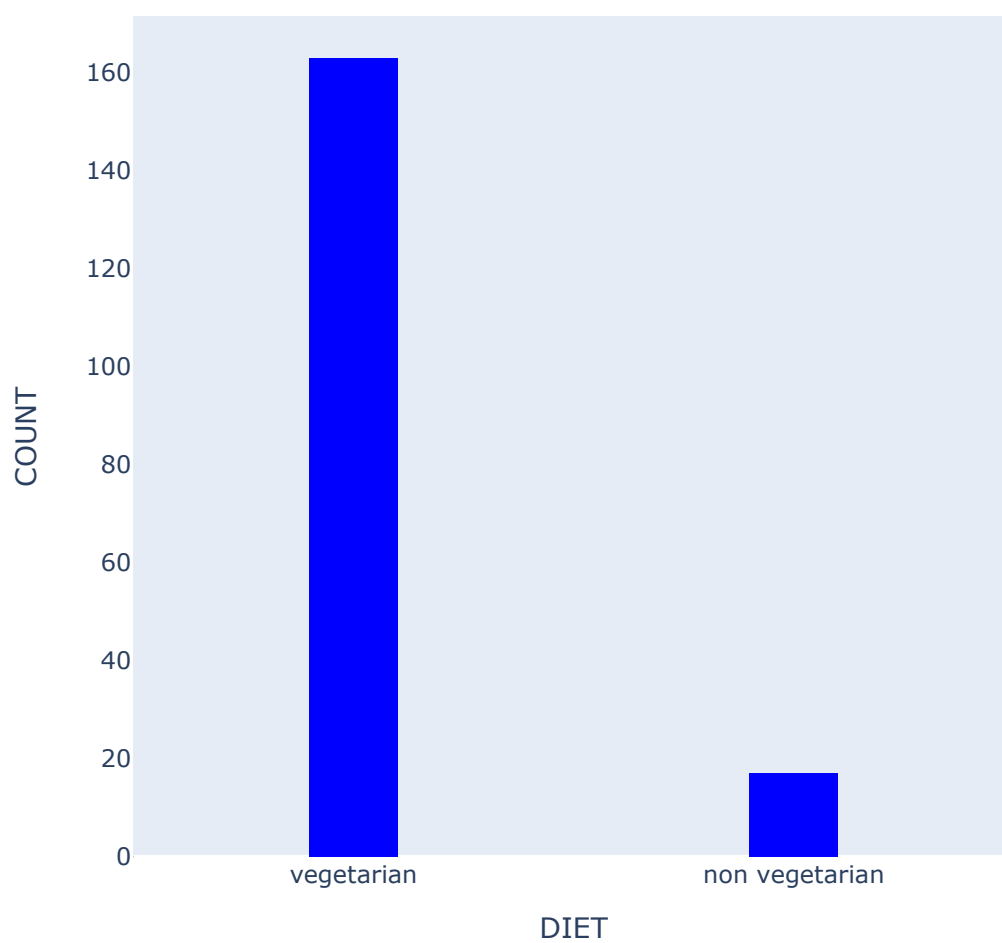
In [21]:

```
fig = go.Figure()
fig.add_trace(go.Histogram(
    x = df['diet'],
    marker_color='blue',
    opacity=1
))

fig.update_layout(
    title_text='DIET_DISTRIBUTION',
    xaxis_title_text='DIET',
    yaxis_title_text='COUNT',
    bargap=0.8,
    xaxis = {'showgrid': False },
    yaxis = {'showgrid': False },
    width=600,
    height=600
)

fig.show()
```

DIET_DISTRIBUTION



The number of vegetarian dishes is more than non vegetarian.

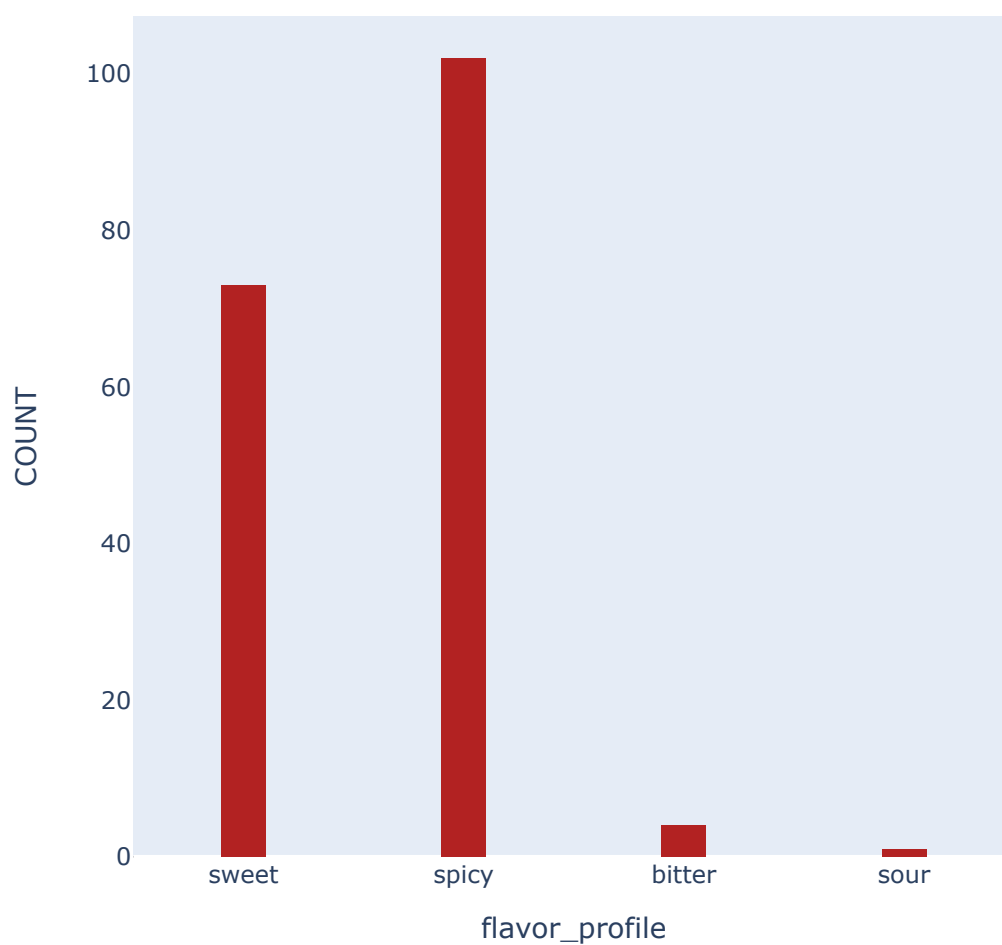
In [22]:

```
fig = go.Figure()
fig.add_trace(go.Histogram(
    x = df['flavor_profile'],
    marker_color='firebrick',
    opacity=1
))

fig.update_layout(
    title_text='FLAVOR_DISTRIBUTION',
    xaxis_title_text='flavor_profile',
    yaxis_title_text='COUNT',
    bargap=0.8,
    xaxis = {'showgrid': False },
    yaxis = {'showgrid': False }
    ,
    width=600,
    height=600
)

fig.show()
```

FLAVOR_DISTRIBUTION



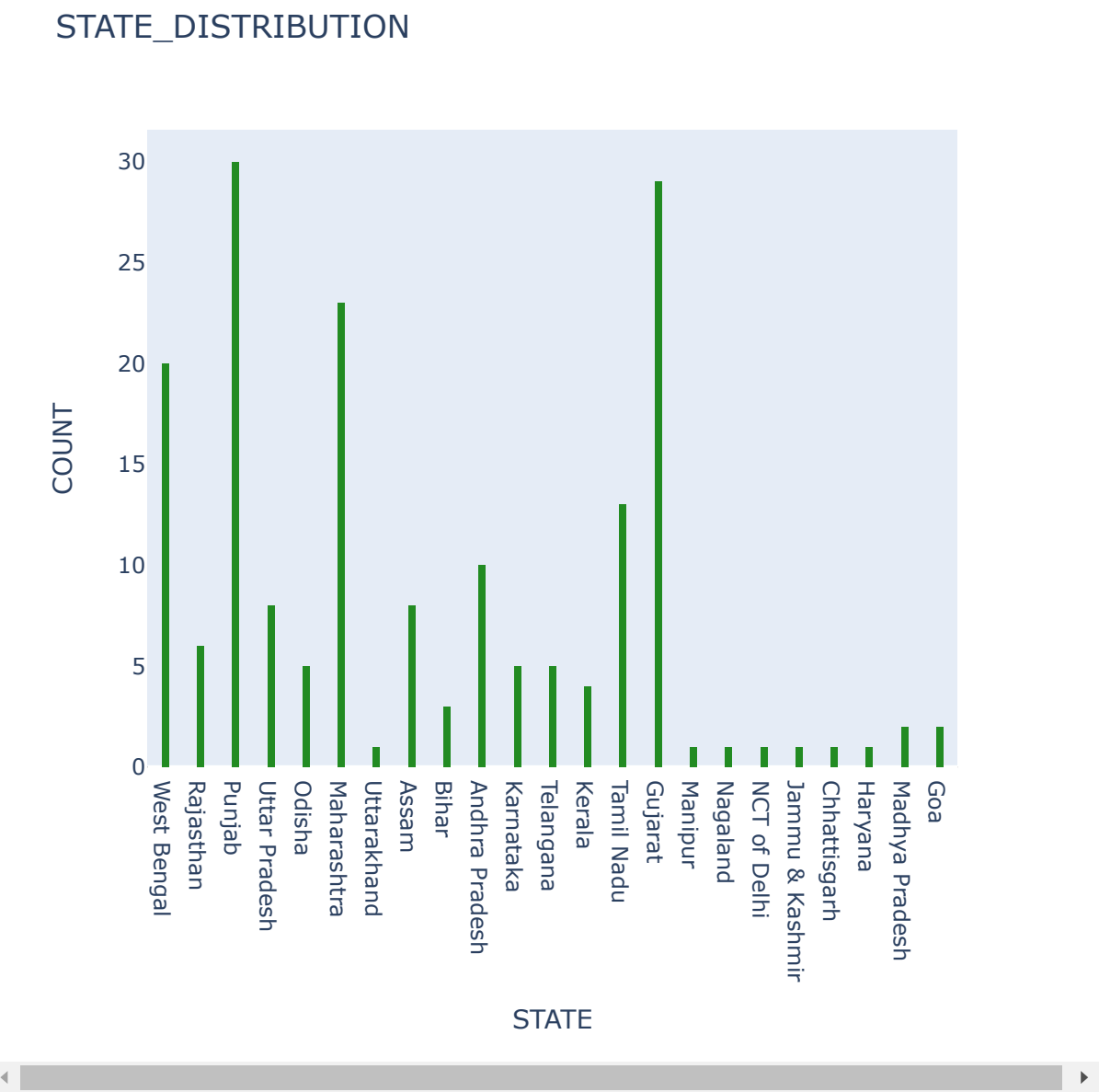
There are 73 sweet , 102 spicy , 4 bitter and 1 sour dishes.

In [23]:

```
fig = go.Figure()
fig.add_trace(go.Histogram(
    x = df['state'],
    marker_color='forestgreen',
    opacity=1
))

fig.update_layout(
    title_text='STATE_DISTRIBUTION',
    xaxis_title_text='STATE',
    yaxis_title_text='COUNT',
    bargap=0.8,
    xaxis = {'showgrid': False },
    yaxis = {'showgrid': False },
    width=600,
    height=600
)

fig.show()
```



The above graph shows State wise number of dishes.

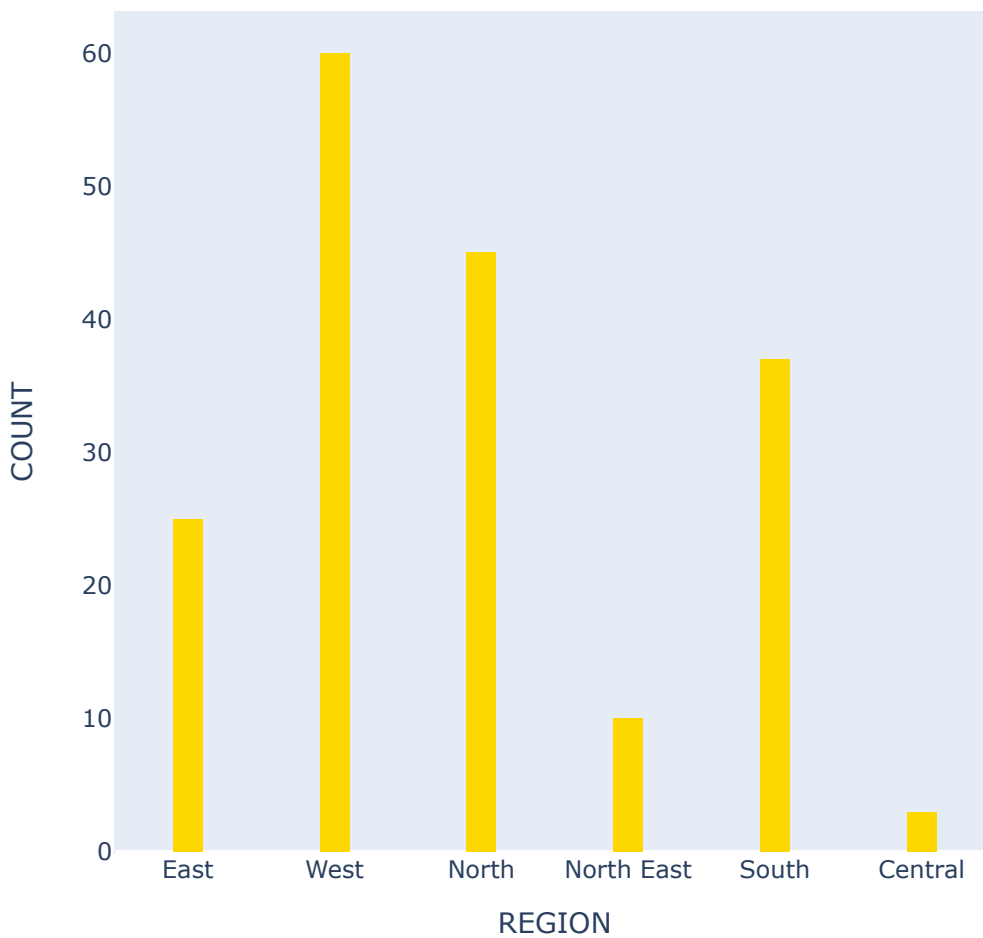
In [24]:

```
fig = go.Figure()
fig.add_trace(go.Histogram(
    x = df['region'],
    marker_color='gold',
    opacity=1
))

fig.update_layout(
    title_text='REGION_DISTRIBUTION',
    xaxis_title_text='REGION',
    yaxis_title_text='COUNT',
    bargap=0.8,
    xaxis = {'showgrid': False },
    yaxis = {'showgrid': False },
    width=600,
    height=600
)

fig.show()
```

REGION_DISTRIBUTION

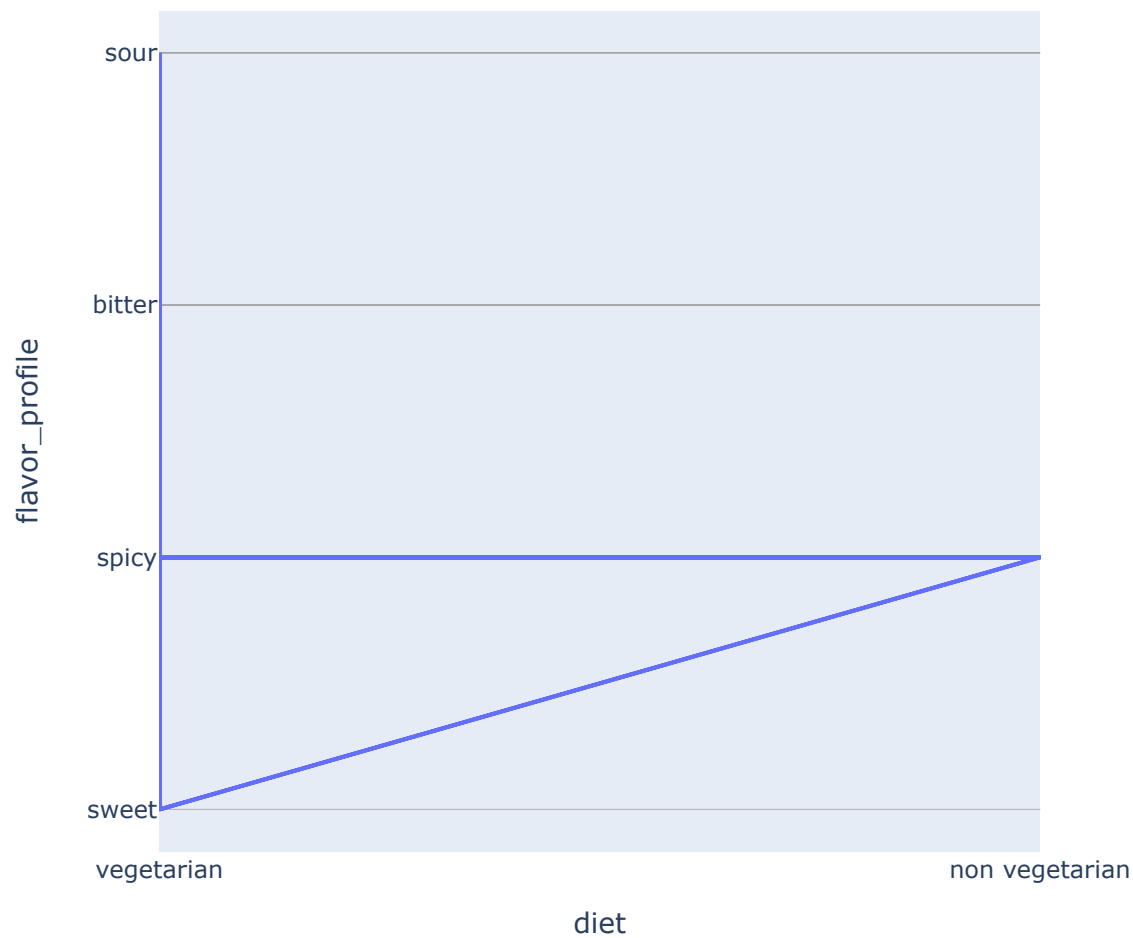


The above graph shows number of region with number of dishes.

In [25]:

```
import plotly.express as px
fig = px.line(df, x="diet", y="flavor_profile", title='Flavour_profile v/s Diet',
              width=600,
              height=600)
fig.show()
```

Flavour_profile v/s Diet



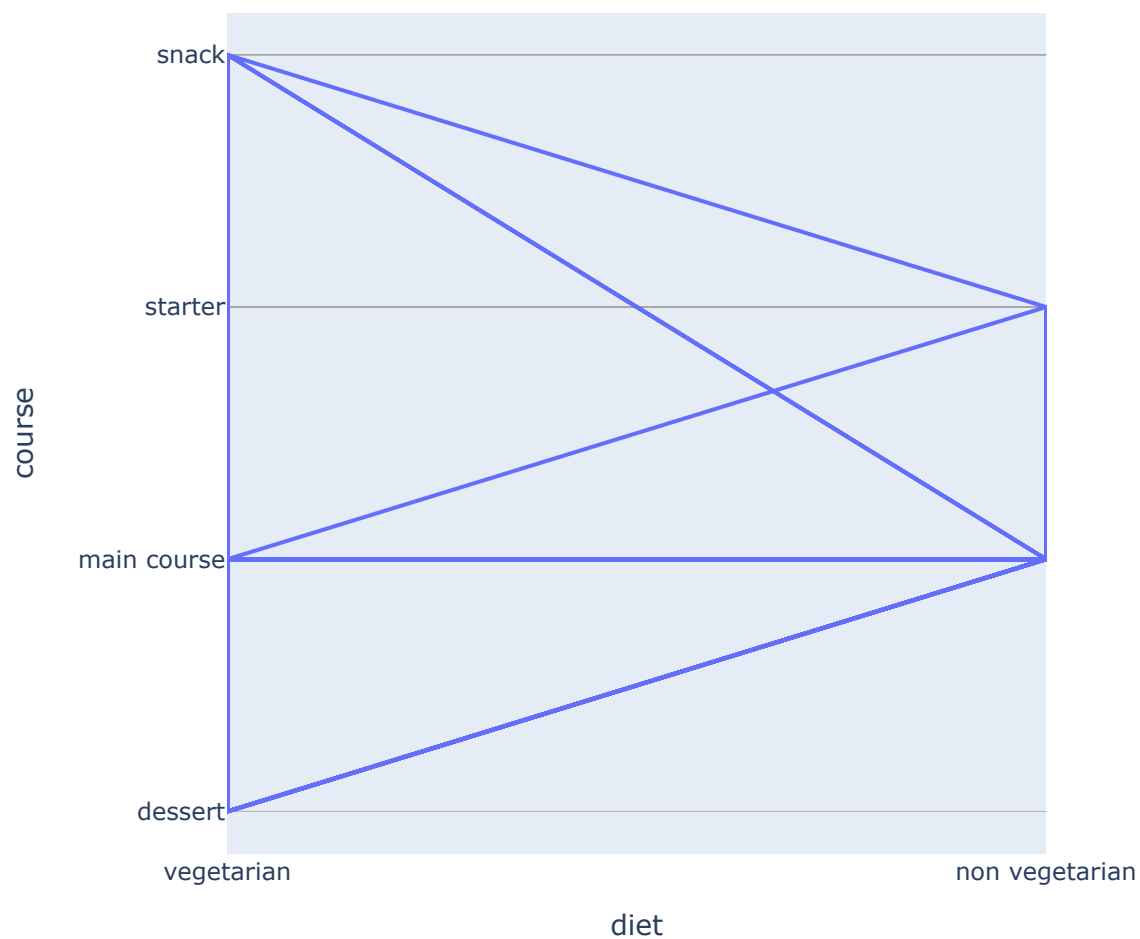
Most of the spicy dishes are non vegetarian.

The sweet , spicy , bitter and sour dishes are vegetarian.

In [26]:

```
import plotly.express as px
fig = px.line(df, x="diet", y="course", title='Course v/s Diet',
              width=600,
              height=600)
fig.show()
```

Course v/s Diet



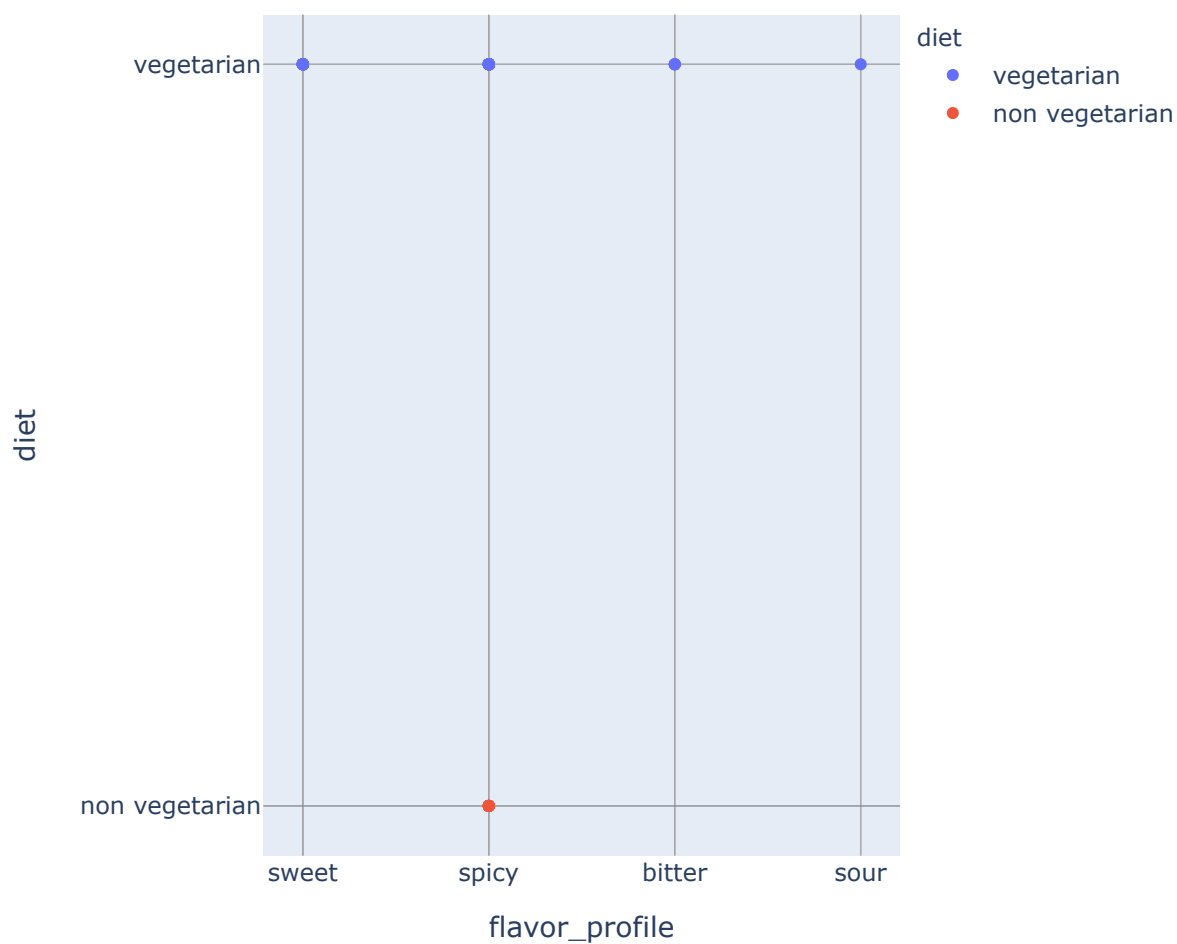
Starter and main course are mostly non vegetarian.

Dessert , main course and snack are mostly vegetarian.

In [27]:

```
fig = px.scatter(df, x="flavor_profile", y="diet", color='diet', title="Flavour V/S Diet",  
width=600,  
height=600)  
fig.show()
```

Flavour V/S Diet



sweet dishes - vegetarian

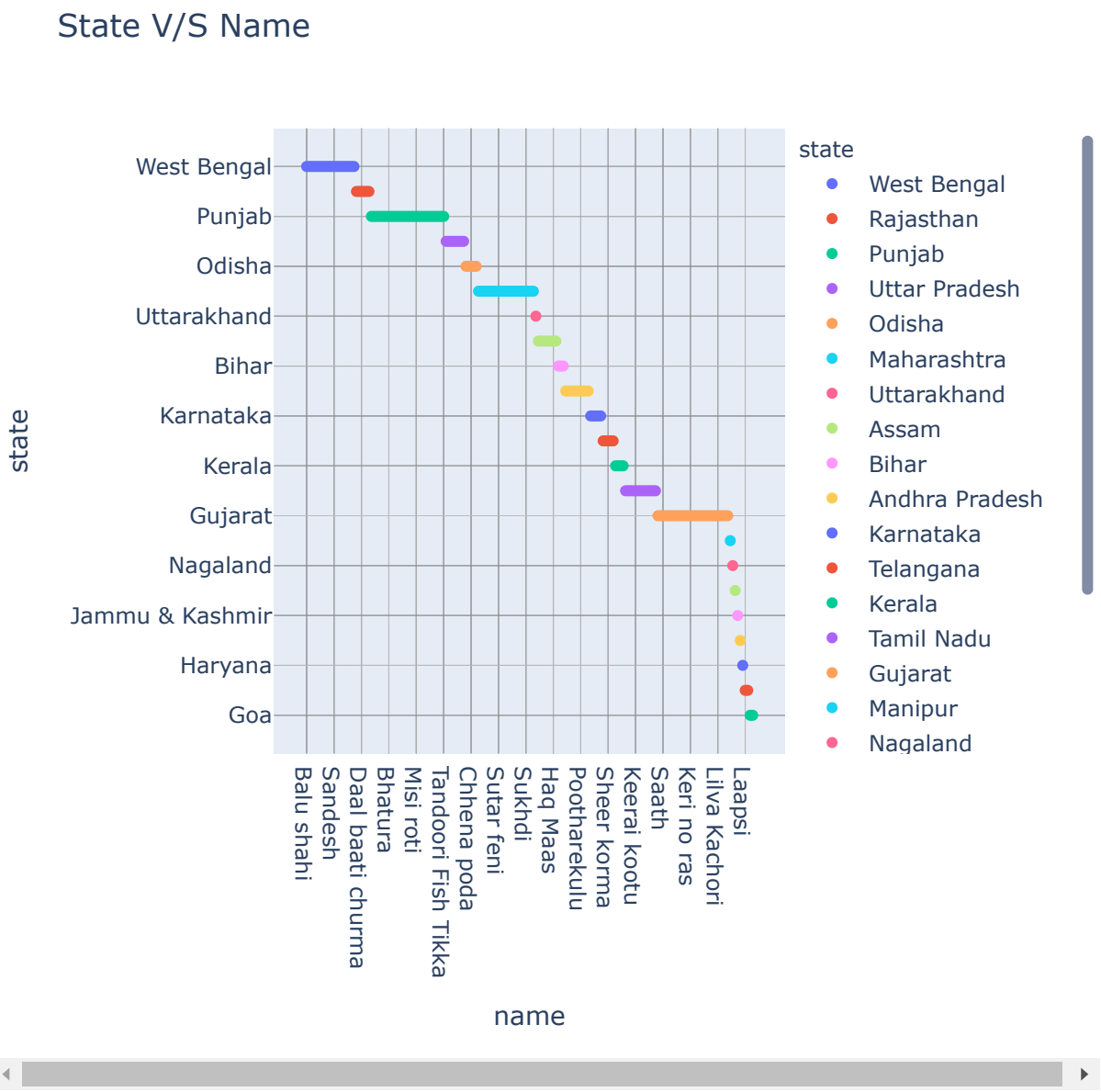
spicy dishes - non vegetarian and vegetarian

bitter dishes - vegetarian

sour dishes - vegetarian

In [28]:

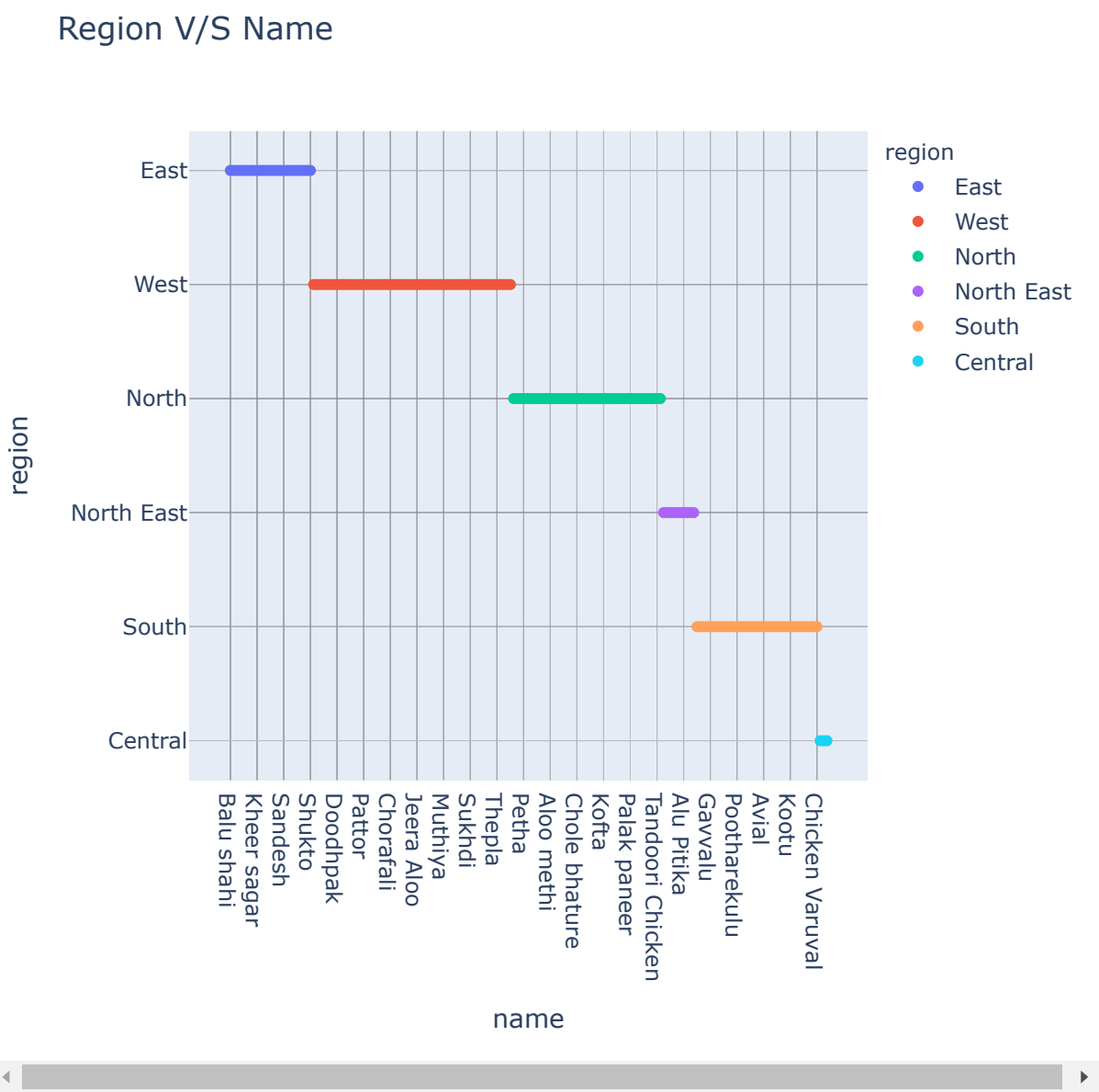
```
fig = px.scatter(df, x="name", y="state",color='state',title="State V/S Name",
width=600,
height=600)
fig.show()
```



Above graph shows the distribution of dishes over states.

In [29]:

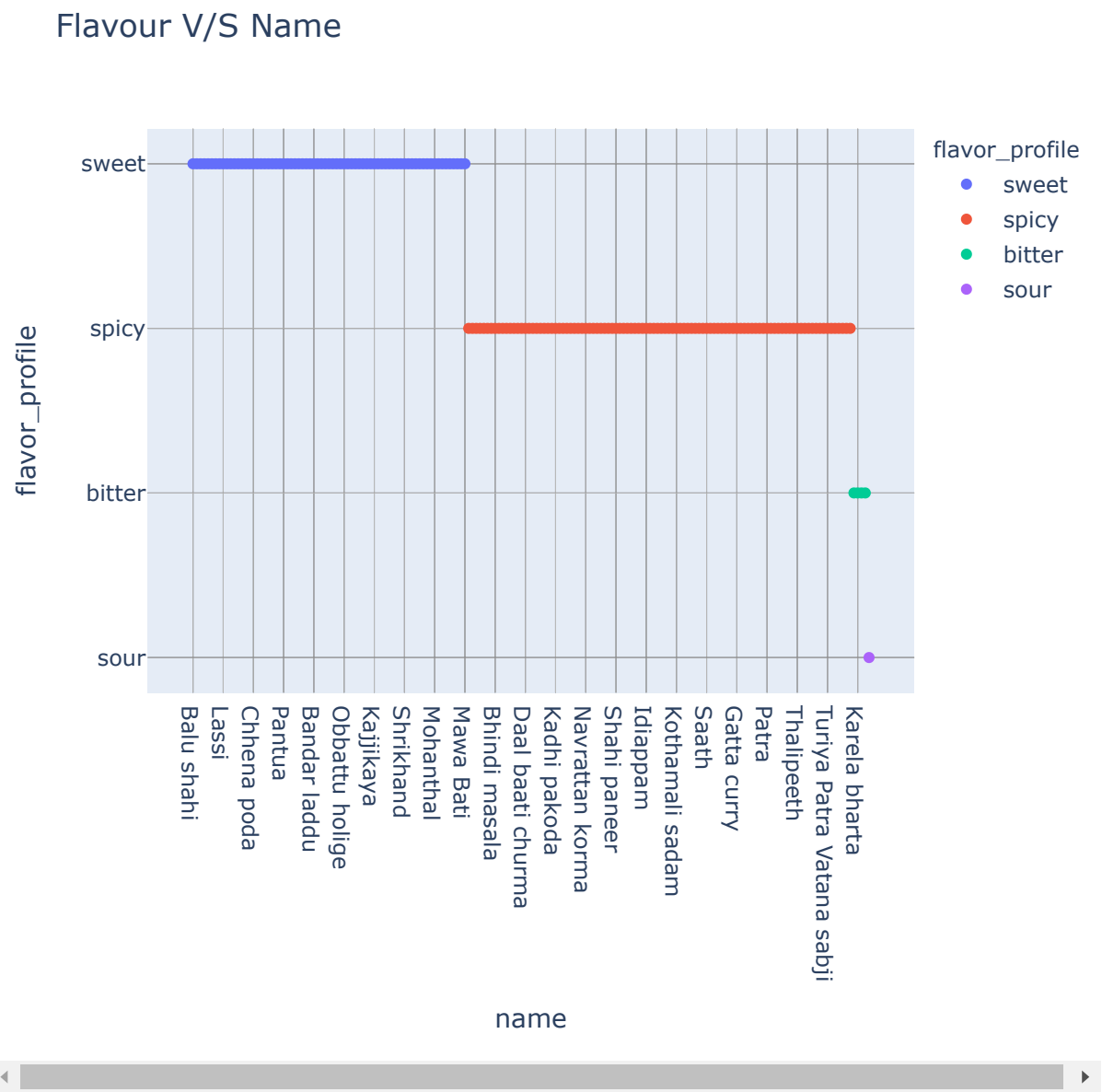
```
fig = px.scatter(df, x="name", y="region",color='region',title="Region V/S Name",
width=600,
height=600)
fig.show()
```



Above graph shows the distribution of dishes over region.

In [30]:

```
fig = px.scatter(df, x="name", y="flavor_profile",color='flavor_profile',title="Flavour V/S Name",
width=600,
height=600)
fig.show()
```

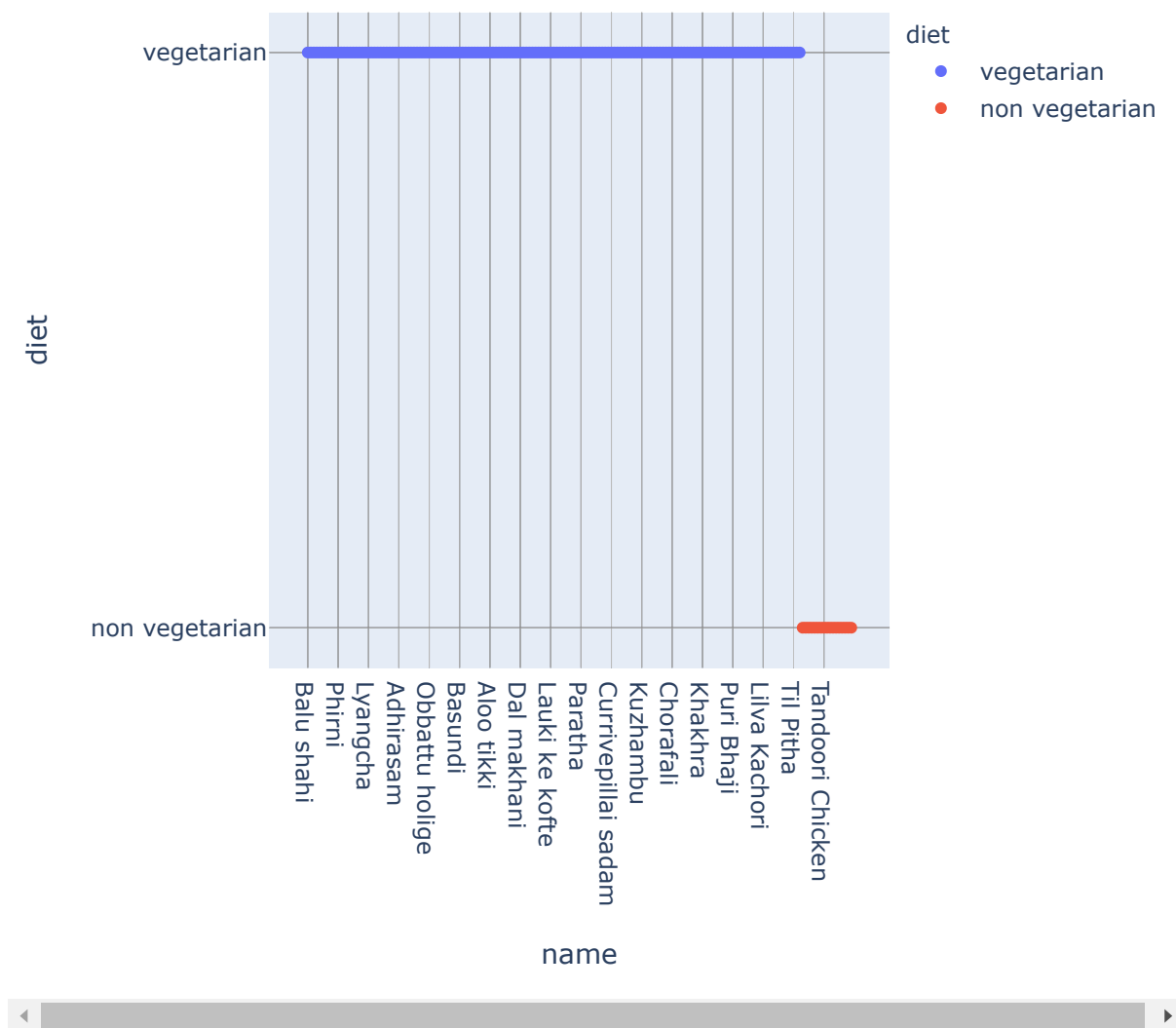


Above graph shows the distribution of dishes over flavour.

In [31]:

```
fig = px.scatter(df, x="name", y="diet", color='diet', title="Diet V/S Name",  
width=600,  
height=600)  
fig.show()
```

Diet V/S Name



Above graph shows the distribution of dishes over diet.

In [32]:

```
fig = px.scatter(df, x="state", y="diet", color='diet', title="Diet V/S State",  
width=600,  
height=600)  
fig.show()
```

Diet V/S State



Above graph shows the distribution of diet over states.

In [33]:

```
fig = px.scatter(df, x="region", y="diet", color='diet', title="Diet V/S Region",  
                width=600,  
                height=600)  
fig.show()
```

Diet V/S Region

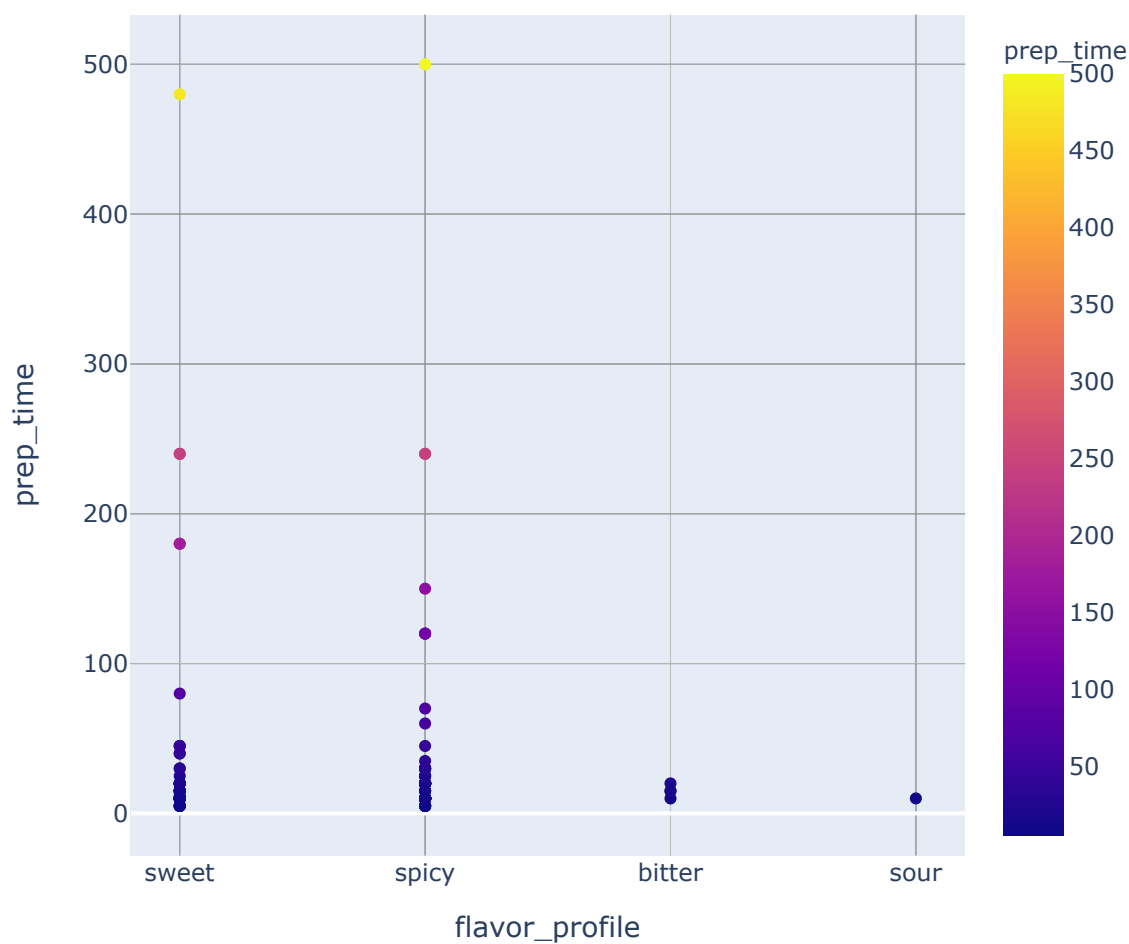


Above graph shows the distribution of diet over region.

In [34]:

```
fig = px.scatter(df, x="flavor_profile", y="prep_time", color='prep_time', title="Prep_time V/S Flavour",  
                width=600,  
                height=600)  
fig.show()
```

Prep_time V/S Flavour

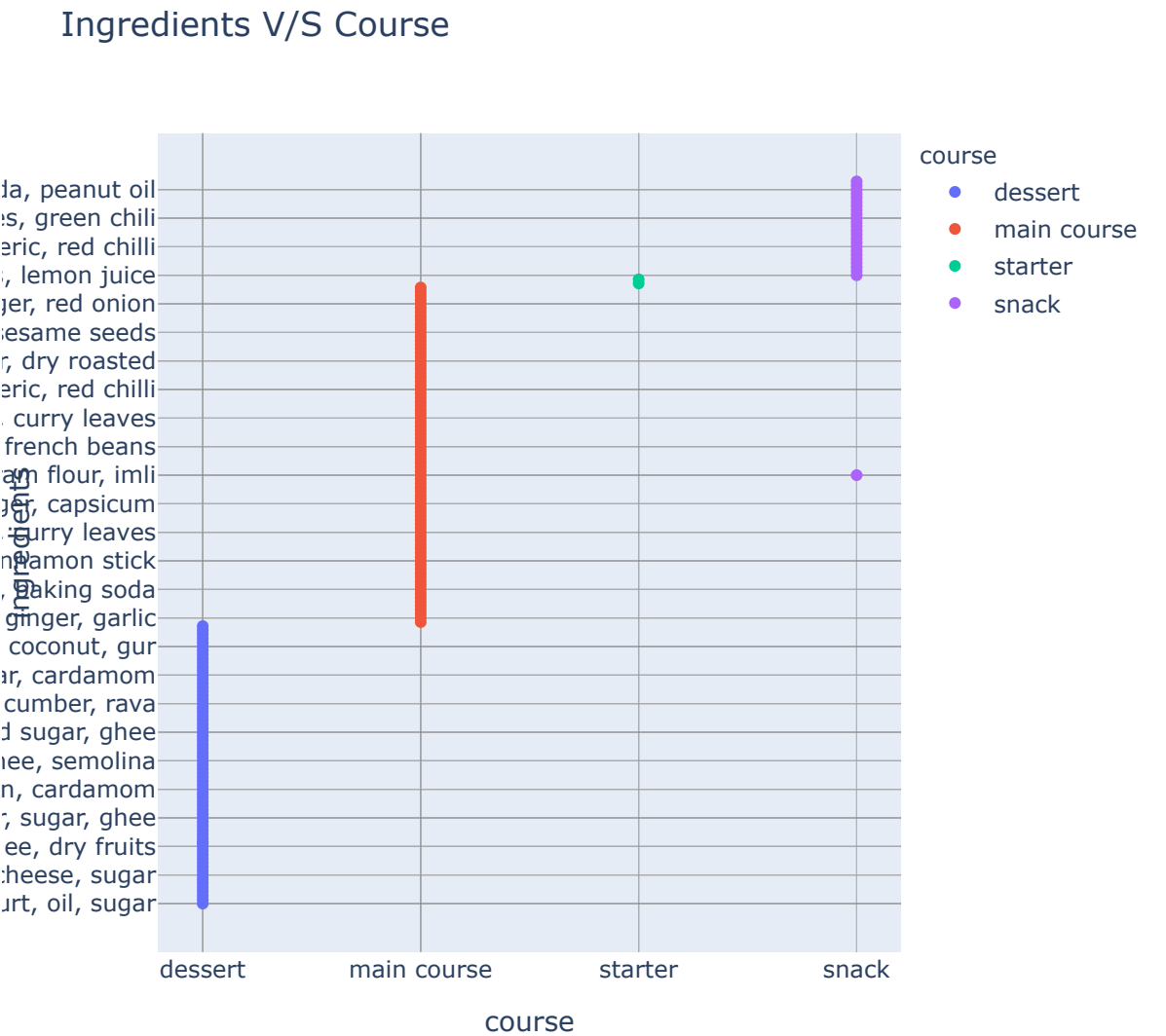


Above graph gives idea about which flavour takes how much time for preparation.

In [35]:

```
fig = px.scatter(df, x="course", y="ingredients", color='course',title="Ingredients V/S Course",
width=600,
height=600)

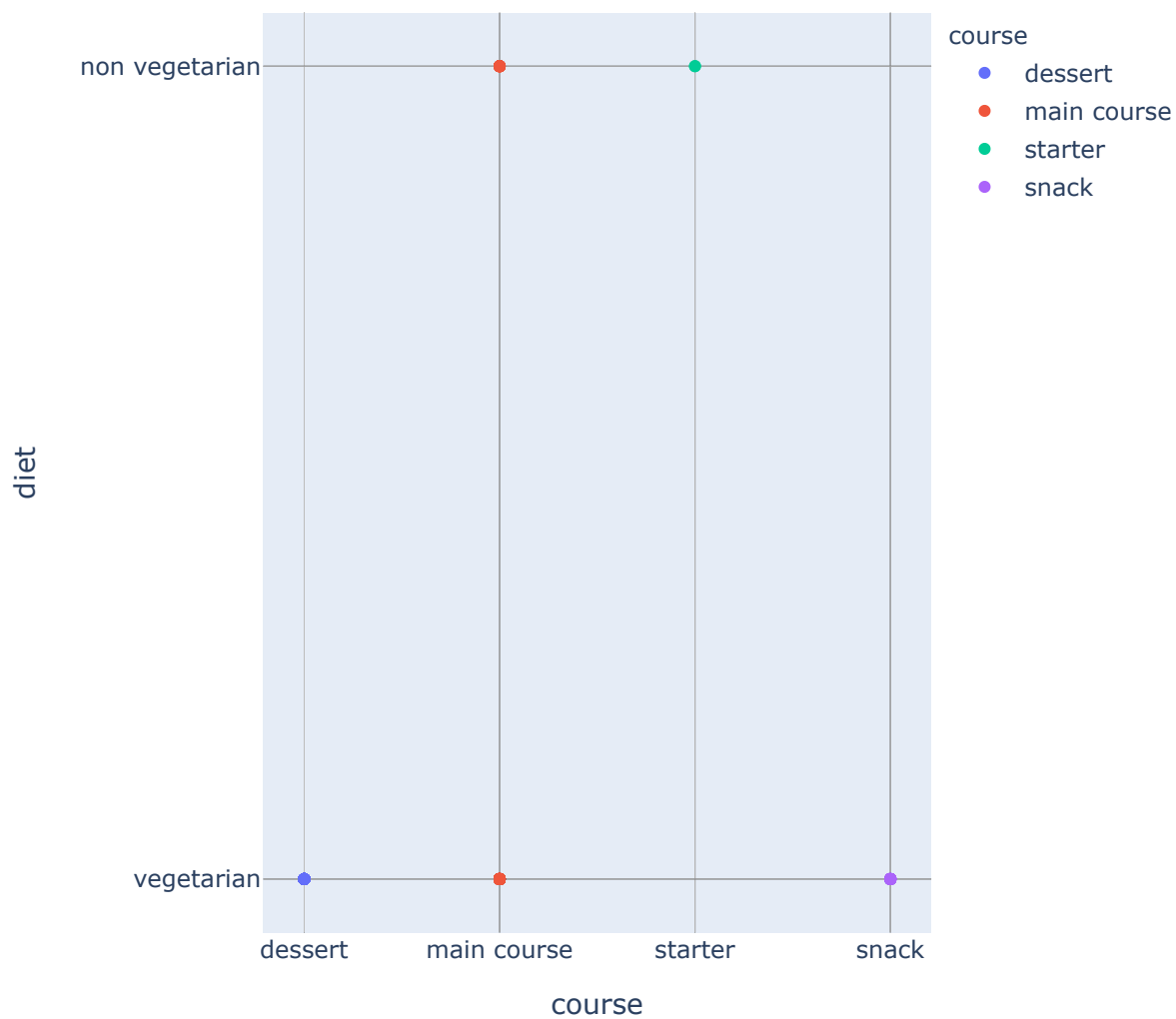
fig.show()
```



Above graph gives idea about which food requires which indegridents.

In [36]:

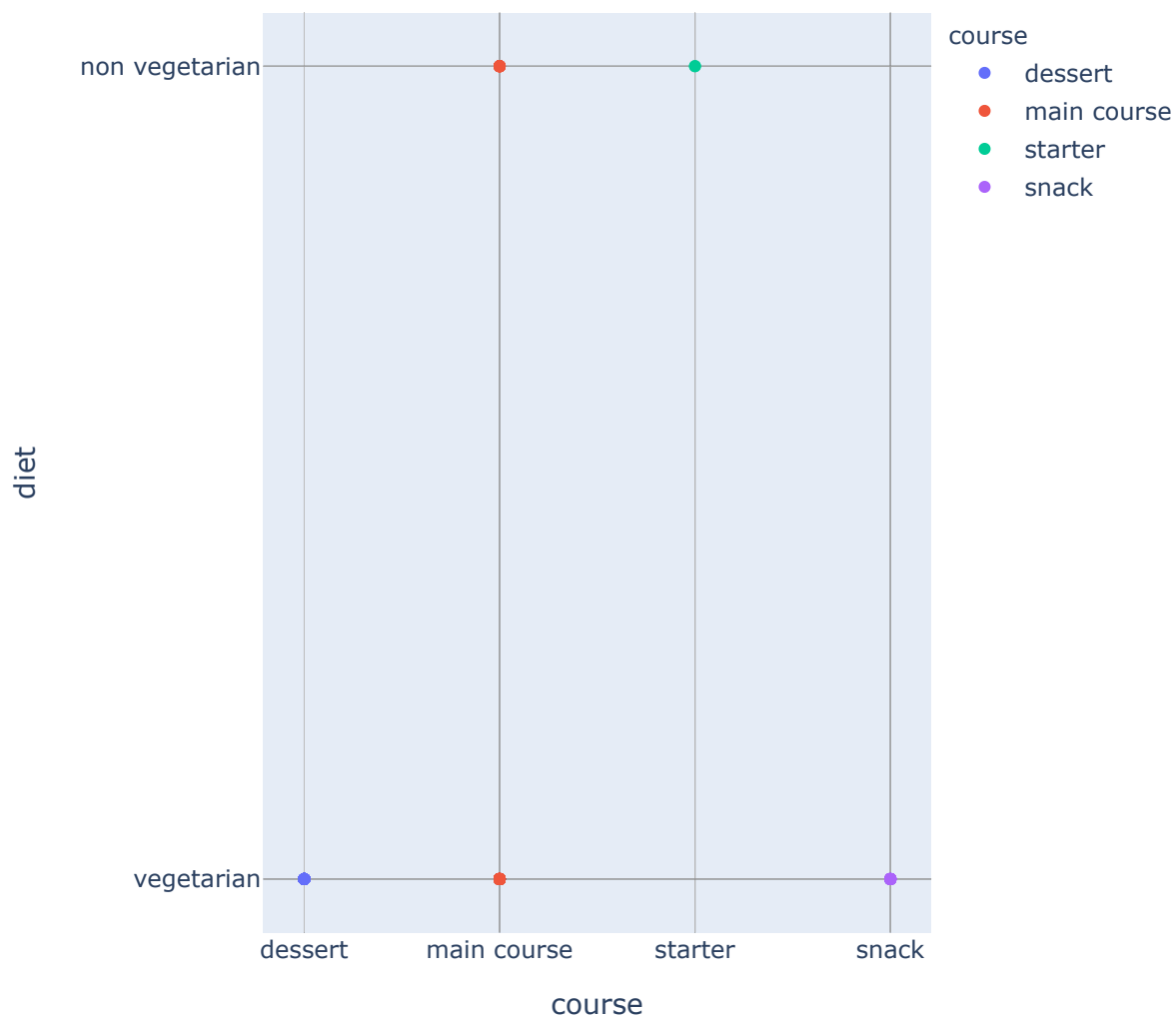
```
fig = px.scatter(df, x="course", y="diet", color="course", hover_data=['prep_time'], size_max=50,
                width=600,
                height=600)
fig.show()
```



Above graph gives idea about preparation time , diet type and food type.

In [37]:

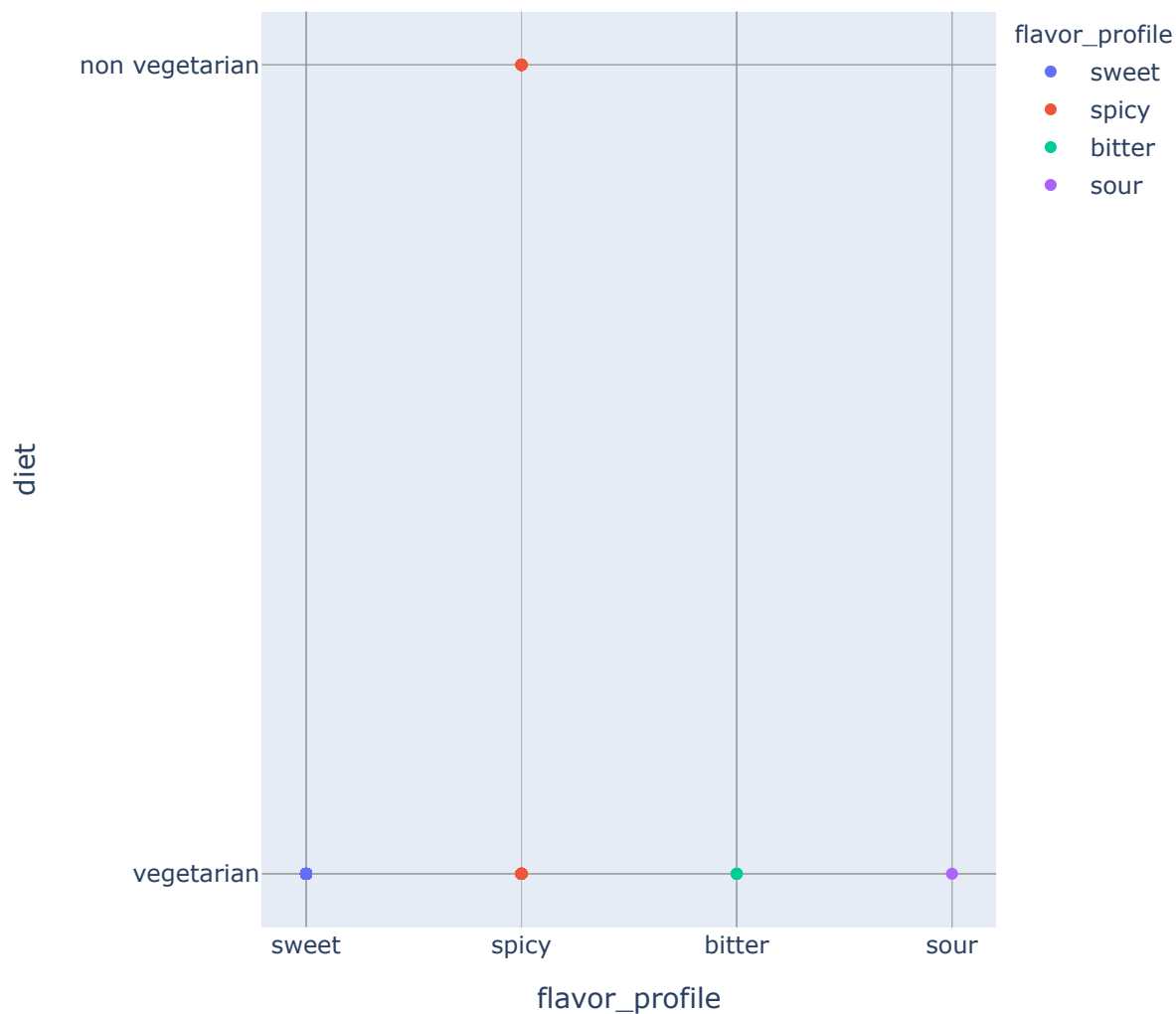
```
fig = px.scatter(df, x="course", y="diet", color="course", hover_data=['cook_time'], size_max=50,  
width=600,  
height=600)  
fig.show()
```



Above graph gives idea about cook time , diet type and food type.

In [38]:

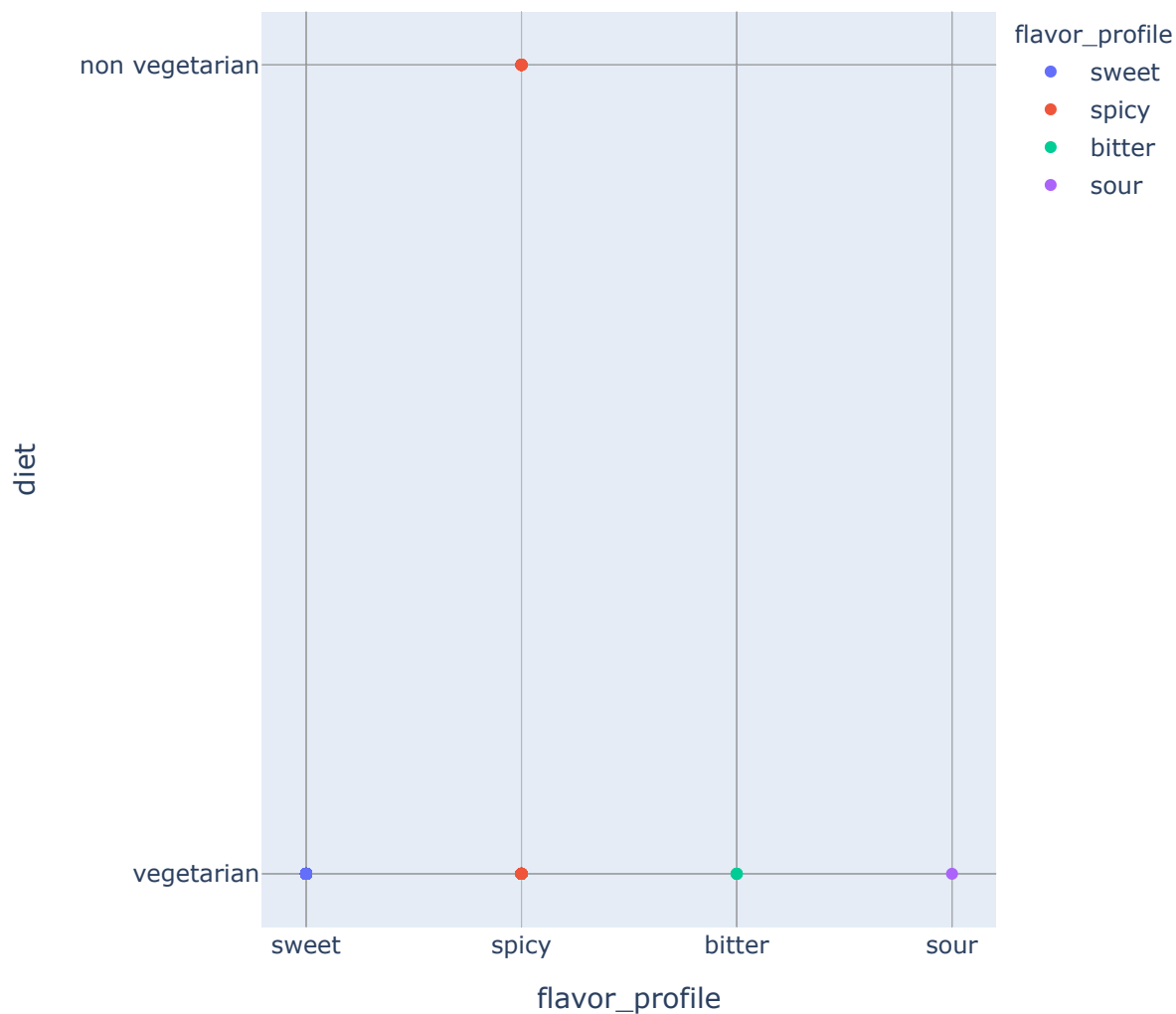
```
fig = px.scatter(df, x="flavor_profile", y="diet", color="flavor_profile",  
                 hover_data=['prep_time'], size_max=50,  
                 width=600,  
                 height=600)  
fig.show()
```



Above graph gives idea about preparation time , diet type and flavour.

In [39]:

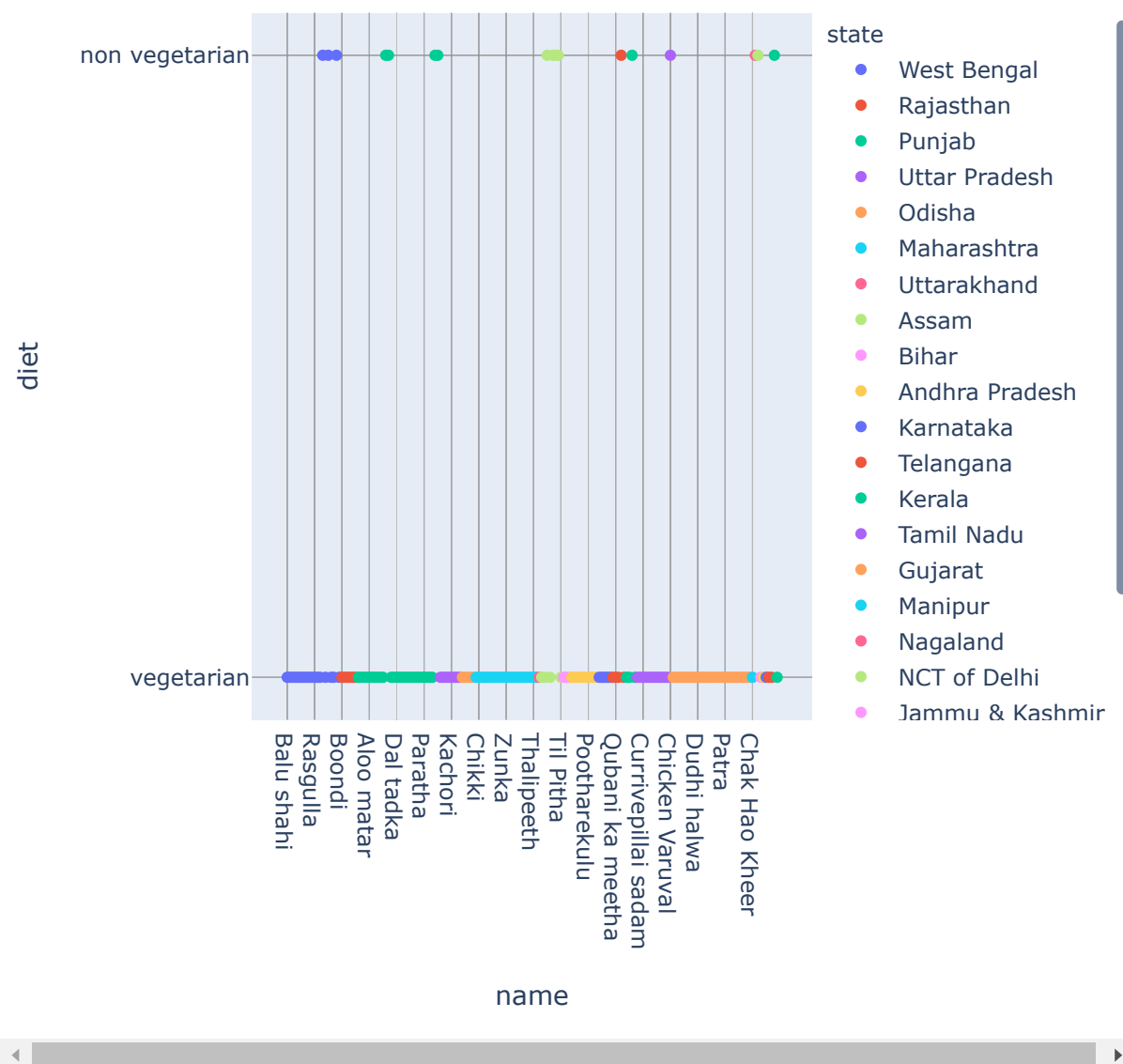
```
fig = px.scatter(df, x="flavor_profile", y="diet", color="flavor_profile",  
                 hover_data=['cook_time'], size_max=50,  
                 width=600,  
                 height=600)  
fig.show()
```



Above graph gives idea about cook time , diet type and flavour.

In [40]:

```
fig = px.scatter(df, x="name", y="diet", color="state",  
                hover_data=['state'], size_max=40,  
                width=600,  
                height=600)  
fig.show()
```

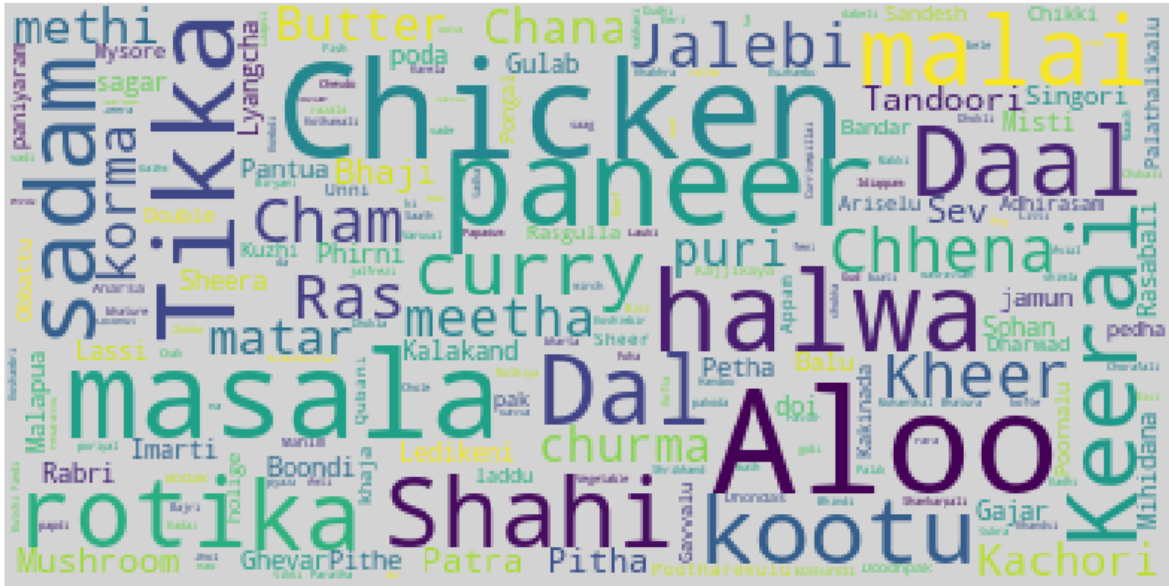


Above graph give us idea about name of the food , state and its diet type.

Dishes Word art

In [41]:

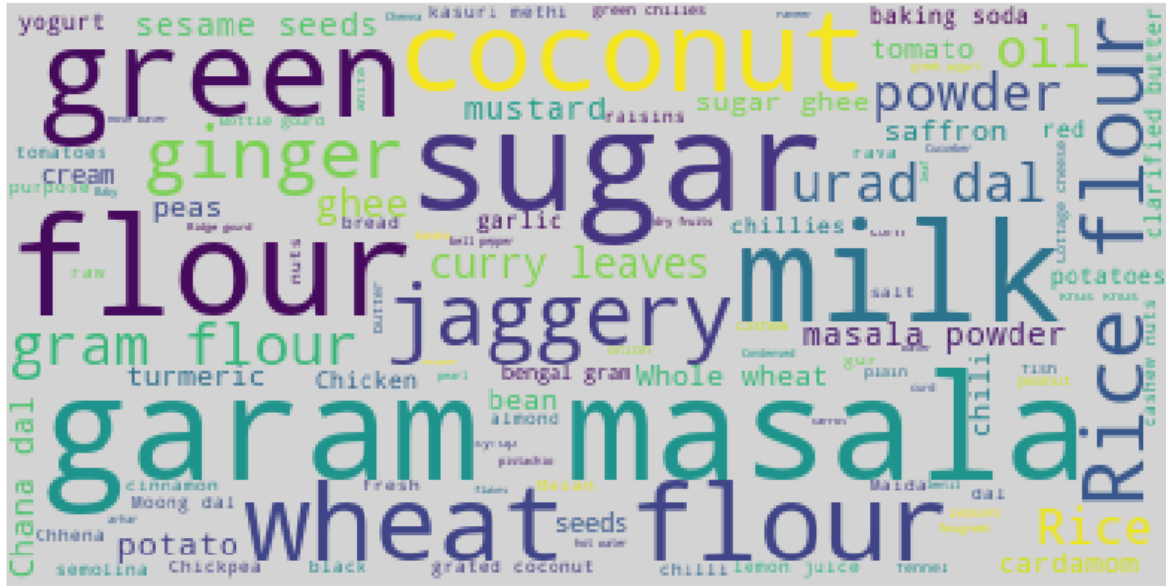
```
wordCloud = WordCloud(
    background_color='lightgrey',
    max_font_size = 50).generate(' '.join(df['name']))
plt.figure(figsize=(14,10))
plt.axis('off')
plt.imshow(wordCloud)
plt.show()
```



Ingredients word art

In [42]:

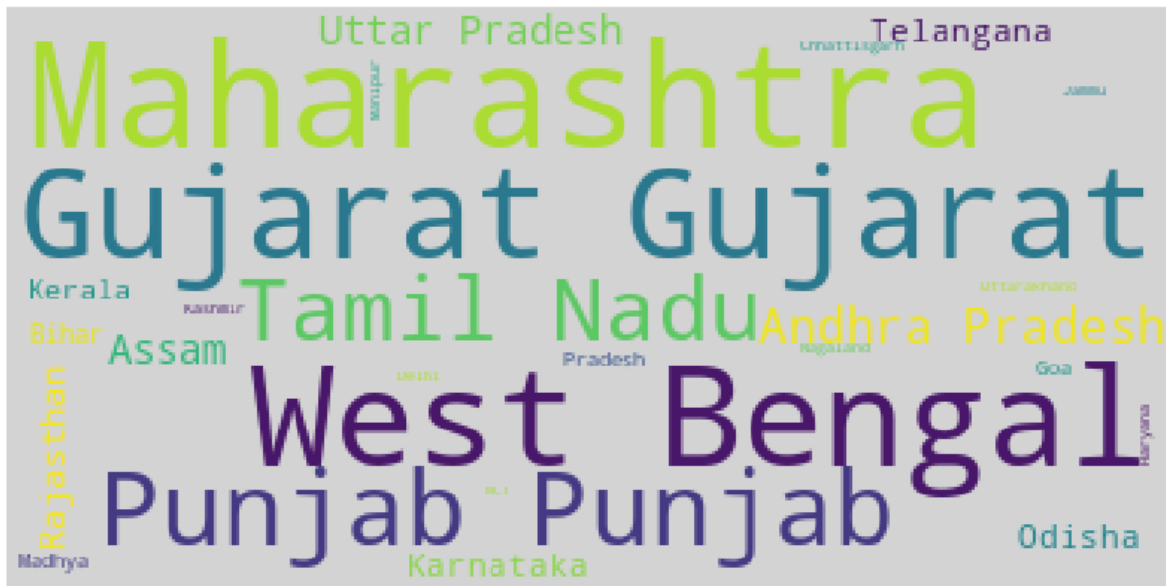
```
wordCloud = WordCloud(
    background_color='lightgrey',
    max_font_size = 50).generate(' '.join(df['ingredients']))
plt.figure(figsize=(14,10))
plt.axis('off')
plt.imshow(wordCloud)
plt.show()
```



State word art

In [43]:

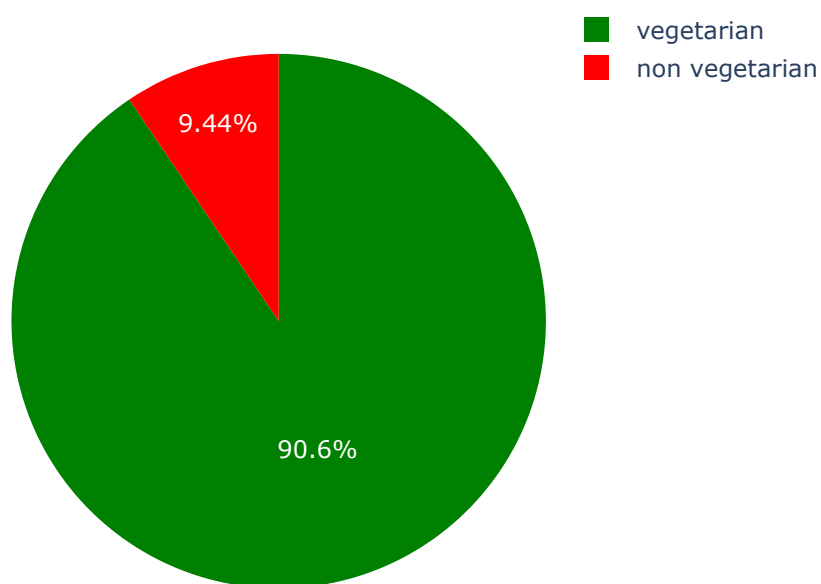
```
wordCloud = WordCloud(
    background_color='lightgrey',
    max_font_size = 50).generate(' '.join(df['state']))
plt.figure(figsize=(14,10))
plt.axis('off')
plt.imshow(wordCloud)
plt.show()
```



In [44]:

```
pie_df = df.diet.value_counts().reset_index()
pie_df.columns = ['diet', 'count']
fig = px.pie(pie_df, values='count', names='diet', title='Proportion of Vegetarian and Non-Vegetarian dishes',
             color_discrete_sequence=['green', 'red'],
             width=500,
             height=500)
fig.show()
```

Proportion of Vegetarian and Non-Vegetarian dishes



Above pie chart shows that number of vegetarian is more than non vegetarian.