



Indian Food and Its Recipes

Importing necessary libraries

```
In [260... import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

import re
from IPython.display import display, HTML
import matplotlib.patches as mpatches
```

```
In [261... df = pd.read_csv('cuisines.csv')
df
```

Out[261...

	name	image_url	description	cuisine	course
0	Thayir Semiya Recipe (Curd Semiya)	https://www.archanaskitchen.com/images/archana...	Thayir Semiya or Curd Vermicelli is a quick di...	Indian	Lunch
1	Chettinad Style Kara Kuzhambu Recipe with Pota...	https://www.archanaskitchen.com/images/archana...	Chettinad Style Kara Kuzhambu Recipe with Pot...	South Indian Recipes	Lunch
2	Goan Chana Ros Recipe (White Peas Curry)	https://www.archanaskitchen.com/images/archana...	Goan Chana Ros is a vegetarian dish from Goan ...	Goan Recipes	Lunch
3	Minced Meat And Egg Croquettes Recipe	https://www.archanaskitchen.com/images/archana...	The croquette is usually cigar shaped or cylin...	North Indian Recipes	Appetizer
4	Thekera Tenga Recipe	https://www.archanaskitchen.com/images/archana...	Thekera Tenga is a Sour curry from Assamese cu...	Assamese	Dinner
...
4231	Stir Fry Green beans and Tofu with Panch Phoro...	https://www.archanaskitchen.com/images/archana...	Stir Fry Green beans and Tofu with Panch Phoro...	Bengali Recipes	Side Dish
4232	Dhuska Recipe (Rice Fried Bread)	https://www.archanaskitchen.com/images/archana...	Dhuska is a traditional recipe from Jharkhand...	Jharkhand	Snack
4233	Khatta Meetha Petha Recipe (Yellow Pumpkin Sabzi)	https://www.archanaskitchen.com/images/archana...	Khatta Meetha Petha or sweet and sour yellow p...	Indian	Lunch
4234	Patta Gobi Matar Nu Shaak	https://www.archanaskitchen.com/images/archana...	Patta Gobi Matar Nu Shaak	North Indian Recipes	Lunch

	name	image_url	description	cuisine	course
	Recipe (Cabbage & Pe...		Recipe (Cabbage & Pe...		
4235	Kerala Pumpkin Pachadi Recipe (Parangikai Pach...	https://www.archanaskitchen.com/images/archana...	The Kerala Pumpkin Pachadi Recipe, is an Ayurv...	Kerala Recipes	Lunch

4236 rows × 9 columns

Steps for performing EDA (Exploratory Data Analysis):

In [262... `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4236 entries, 0 to 4235
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   name            4236 non-null   object
1   image_url       4236 non-null   object
2   description     4236 non-null   object
3   cuisine         4230 non-null   object
4   course          4198 non-null   object
5   diet            4199 non-null   object
6   prep_time       4226 non-null   object
7   ingredients     4236 non-null   object
8   instructions    4236 non-null   object
dtypes: object(9)
memory usage: 298.0+ KB
```

1. Understand Data Structure

In [263... `numerical_features = df.select_dtypes(include=['int64','int32','float']).columns`
numerical_features

Out[263... `[]`

In [264... `categorical_features = df.select_dtypes(include=['object','category']).columns`
categorical_features

```
Out[264... ['name',
            'image_url',
            'description',
            'cuisine',
            'course',
            'diet',
            'prep_time',
            'ingredients',
            'instructions']
```

```
In [265... #text features
```

```
text_features=[]
categorical_features_cleaned=[]

for col in categorical_features:
    unique_ratio = df[col].nunique()/len(df)
    avg_length = df[col].astype(str).str.len().mean()

    if unique_ratio > 0.5 or avg_length > 20:
        text_features.append(col)
    else:
        categorical_features_cleaned.append(col)

print(f"::: Cleaned Categorical Features :::")
print(categorical_features_cleaned)
print()
print(f"::: Cleaned Text Features :::")
print(text_features)
```

```
::: Cleaned Categorical Features :::
['cuisine', 'course', 'diet', 'prep_time']
```

```
::: Cleaned Text Features :::
['name', 'image_url', 'description', 'ingredients', 'instructions']
```

2. Data Cleaning

```
In [266... # for val in df['prep_time'].values:
#         numerical = re.search("[0-9][0-9]",str(val))
#         if numerical:
#             print(numerical.group())
```

```
In [267... #extrcting only
df['prep_time']=df['prep_time'].astype(str).str.extract(r"(\d+)")
```

```
In [268... #removing extra spaces, tabs and newline characters
df['ingredients'] = df['ingredients'].astype(str).str.replace(r"\s+", " ", regex
```

```
In [269... df['diet'] = df['diet'].astype(str).str.strip().str.title().replace({
    "Non Vegeterian": "Non Vegetarian"
```

```
)  
  
df['diet'] = df['diet'].replace("Nan", "Unknown")
```

```
In [270... # Find duplicate rows  
duplicates = df[df.duplicated()]  
# print(duplicates)  
  
# Keep only the first occurrence  
df_cleaned = df.drop_duplicates()  
  
# OR keep the last occurrence  
# df = df.drop_duplicates(keep='last')
```

```
In [271... df_cleaned.shape
```

```
Out[271... (4226, 9)
```

```
In [272... df.shape
```

```
Out[272... (4236, 9)
```

```
In [273... duplicate_rows_dropped = df.shape[0]-df_cleaned.shape[0]  
print(f"Number of duplicates rows ::: {duplicate_rows_dropped}")
```

```
Number of duplicates rows ::: 10
```

```
In [274... df_cleaned
```

Out[274...

	name	image_url	description	cuisine	course
0	Thayir Semiya Recipe (Curd Semiya)	https://www.archanaskitchen.com/images/archana...	Thayir Semiya or Curd Vermicelli is a quick di...	Indian	Lunch
1	Chettinad Style Kara Kuzhambu Recipe with Pota...	https://www.archanaskitchen.com/images/archana...	Chettinad Style Kara Kuzhambu Recipe with Pot...	South Indian Recipes	Lunch
2	Goan Chana Ros Recipe (White Peas Curry)	https://www.archanaskitchen.com/images/archana...	Goan Chana Ros is a vegetarian dish from Goan ...	Goan Recipes	Lunch
3	Minced Meat And Egg Croquettes Recipe	https://www.archanaskitchen.com/images/archana...	The croquette is usually cigar shaped or cylin...	North Indian Recipes	Appetizer
4	Thekera Tenga Recipe	https://www.archanaskitchen.com/images/archana...	Thekera Tenga is a Sour curry from Assamese cu...	Assamese	Dinner
...
4231	Stir Fry Green beans and Tofu with Panch Phoro...	https://www.archanaskitchen.com/images/archana...	Stir Fry Green beans and Tofu with Panch Phoro...	Bengali Recipes	Side Dish
4232	Dhuska Recipe (Rice Fried Bread)	https://www.archanaskitchen.com/images/archana...	Dhuska is a traditional recipe from Jharkhand...	Jharkhand	Snack
4233	Khatta Meetha Petha Recipe (Yellow Pumpkin Sabzi)	https://www.archanaskitchen.com/images/archana...	Khatta Meetha Petha or sweet and sour yellow p...	Indian	Lunch
4234	Patta Gobi Matar Nu Shaak	https://www.archanaskitchen.com/images/archana...	Patta Gobi Matar Nu Shaak	North Indian Recipes	Lunch

	name	image_url	description	cuisine	course
	Recipe (Cabbage & Pe...		Recipe (Cabbage & Pe...		
4235	Kerala Pumpkin Pachadi Recipe (Parangikai Pach...	https://www.archanaskitchen.com/ images/archana...	The Kerala Pumpkin Pachadi Recipe, is an Ayurv...	Kerala Recipes	Lunch

4226 rows × 9 columns

3.Univariate Analysis

A. Distribution of cuisine (top cuisines).

```
In [275... #count cuisine according to cuisine.
data_cuisine = df.groupby('cuisine').count()['name']
print(f"Cuisine :")
print()
print(data_cuisine)
```

Cuisine :

cuisine	
Afghan	1
African	1
Andhra	116
Arab	1
Asian	8
Assamese	26
Awadhi	37
Bengali Recipes	164
Bihari	22
Chettinad	70
Chinese	2
Coastal Karnataka	14
Continental	5
Coorg	15
Fusion	16
Goan Recipes	83
Gujarati Recipes	124
Haryana	3
Himachal	17
Hyderabadi	29
Indian	809
Indo Chinese	5
Jharkhand	2
Karnataka	133
Kashmiri	59
Kerala Recipes	159
Kongunadu	3
Konkan	32
Lucknowi	9
Maharashtrian Recipes	142
Malabar	11
Malvani	9
Mangalorean	45
Middle Eastern	6
Mughlai	31
Nagaland	5
Nepalese	5
North East India Recipes	20
North Indian Recipes	827
North Karnataka	14
Oriya Recipes	29
Pakistani	11
Parsi Recipes	37
Punjabi	86
Rajasthani	117
Sichuan	1
Sindhi	32
South Indian Recipes	633
South Karnataka	8
Sri Lankan	8
Tamil Nadu	159

Thai	1
Udupi	13
Uttar Pradesh	11
Uttarakhand-North Kumaon	4

Name: name, dtype: int64

```
In [276... cuisine_names = data_cuisine.index.tolist()
cuisine_values = data_cuisine.values.tolist()

# Create a DataFrame
cuisine_df = pd.DataFrame({'cuisine': cuisine_names, 'count': cuisine_values})
print("Top cuisine DataFrame")
print()
cuisine_df
```

Top cuisine DataFrame

Out[276...

	cuisine	count
0	Afghan	1
1	African	1
2	Andhra	116
3	Arab	1
4	Asian	8
5	Assamese	26
6	Awadhi	37
7	Bengali Recipes	164
8	Bihari	22
9	Chettinad	70
10	Chinese	2
11	Coastal Karnataka	14
12	Continental	5
13	Coorg	15
14	Fusion	16
15	Goan Recipes	83
16	Gujarati Recipes	124
17	Haryana	3
18	Himachal	17
19	Hyderabadi	29
20	Indian	809
21	Indo Chinese	5
22	Jharkhand	2
23	Karnataka	133
24	Kashmiri	59
25	Kerala Recipes	159
26	Kongunadu	3
27	Konkan	32
28	Lucknowi	9
29	Maharashtrian Recipes	142
30	Malabar	11

	cuisine	count
31	Malvani	9
32	Mangalorean	45
33	Middle Eastern	6
34	Mughlai	31
35	Nagaland	5
36	Nepalese	5
37	North East India Recipes	20
38	North Indian Recipes	827
39	North Karnataka	14
40	Oriya Recipes	29
41	Pakistani	11
42	Parsi Recipes	37
43	Punjabi	86
44	Rajasthani	117
45	Sichuan	1
46	Sindhi	32
47	South Indian Recipes	633
48	South Karnataka	8
49	Sri Lankan	8
50	Tamil Nadu	159
51	Thai	1
52	Udupi	13
53	Uttar Pradesh	11
54	Uttarakhand-North Kumaon	4

```
In [277... # Sort in descending order
top_cuisines = cuisine_df.sort_values(by='count', ascending=False).reset_index

top_5 = 5

# Assign colors
colors = ['orange' if i < top_5 else 'skyblue' for i in range(len(top_cuisines)

plt.figure(figsize=(25,10))
# bars = plt.bar(cuisine_names,cuisine_values,color=colors)
```

```

bars = plt.bar(top_cuisines['cuisine'],top_cuisines['count'],color=colors)

plt.xticks(rotation=90, fontsize=15)

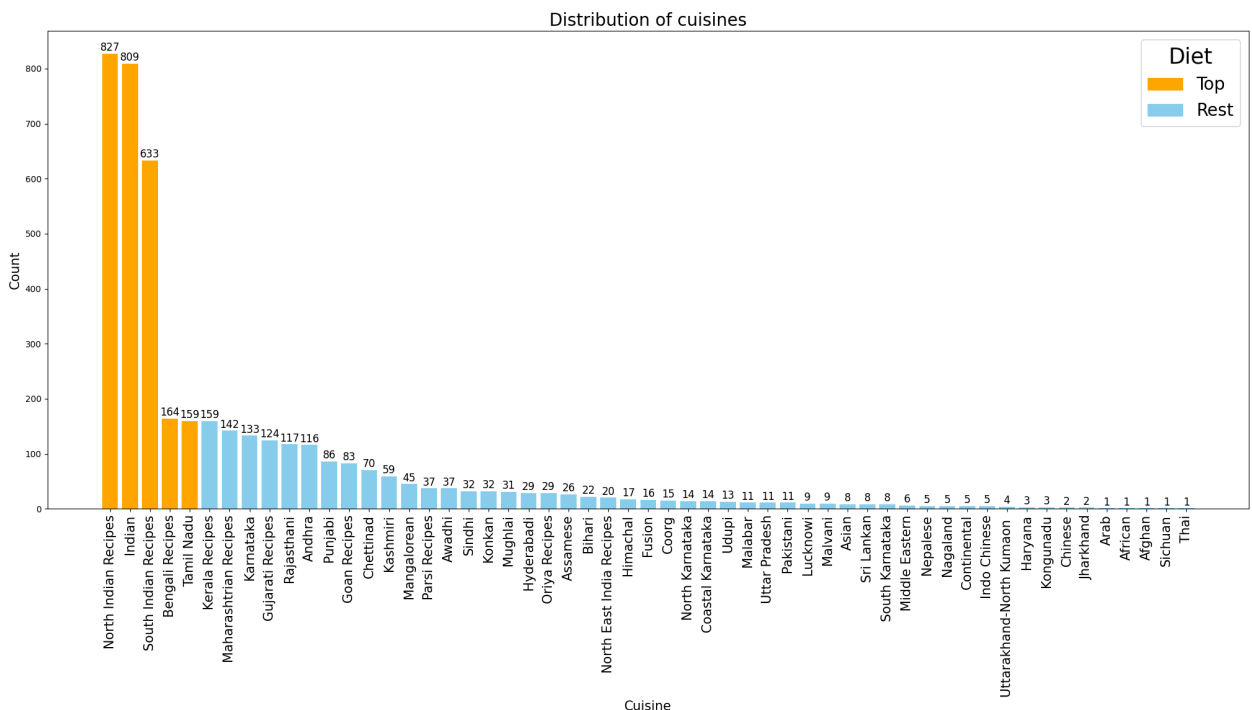
# Create legend manually
legend_handles = [
    mpatches.Patch(color="orange", label="Top"),
    mpatches.Patch(color="skyblue", label="Rest")
]

plt.legend(handles=legend_handles,
           title="Diet",
           title_fontsize=25, # title size
           fontsize=20,      # legend text size
           loc="upper right", # placement
           bbox_to_anchor=(1, 1)) # fine-tune position

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x()+bar.get_width()/2,height+1,str(height),ha="center",va="bottom")

plt.xlabel("Cuisine", fontsize=15)
plt.ylabel("Count", fontsize=15)
plt.title("Distribution of cuisines", fontsize=20)
# plt.savefig("1.Distribution of cuisines with top highlighted.png", bbox_inches="tight")
plt.show()

```



✓ Explanation

bar.get_height() → gets the value (height) of each bar.

`plt.text(x, y, text, ha='center', va='bottom')` → places the text on top of the bar:

- **x** → the center position of the bar
- **y** → a little above the bar (height + 1)
- **ha='center'** → horizontally centers the text
- **va='bottom'** → vertically places the text just above the bar

✓ Explanation:

- `bbox_inches='tight'` ensures that the saved figure includes all labels, titles, and margins.
- This is especially useful when the x-axis labels are rotated or very long.
- You can also adjust `figsize` if you need more horizontal space for the plot.

Insights from Cuisine Distribution

- **North Indian, South Indian, and Indian recipes** dominate with over **600-800 recipes each**.
- **Bengali, Tamil Nadu, Kerala, and Maharashtrian** cuisines are moderately represented (100-160 recipes).
- Smaller cuisines like **Thai, Afghan, African** have very few recipes (1-5 only).
- The dataset is **highly skewed towards Indian regional cuisines**.

B.Distribution of course (most common meal type).

```
In [278... data_course = df.groupby('course').count()['name']  
print("Course with counts:")  
print()  
data_course
```

Course with counts:

```
Out[278...] course
Appetizer          91
Brunch             1
Dessert            389
Dinner             442
Indian Breakfast   77
Lunch             1638
Main Course        203
North Indian Breakfast 119
One Pot Dish       19
Side Dish          855
Snack              82
South Indian Breakfast 275
World Breakfast    7
Name: name, dtype: int64
```

```
In [279...] data_course_name = data_course.index.tolist()
data_course_count = data_course.values.tolist()

#create dataframe
df_course = pd.DataFrame({
    'name':data_course_name,'count':data_course_count
})
df_course
```

```
Out[279...]      name  count
0      Appetizer    91
1        Brunch     1
2        Dessert   389
3        Dinner   442
4  Indian Breakfast    77
5          Lunch  1638
6    Main Course   203
7  North Indian Breakfast 119
8    One Pot Dish    19
9        Side Dish   855
10         Snack    82
11  South Indian Breakfast 275
12    World Breakfast     7
```

```
In [280...] df_course_desc = df_course.sort_values(by='count', ascending=False).reset_index()
top_10_course = df_course_desc[:10]
```

```

colors = ['orange' if i < 10 else 'skyblue' for i in range(len(df_course_desc))

plt.figure(figsize=(25,10))
# plt.bar(df_course_desc['name'],df_course_desc['count'],color=colors)
bars = plt.bar(df_course_desc['name'],df_course_desc['count'],color=colors)

plt.xticks(rotation=90,fontsize=15)
plt.yticks(fontsize=15)

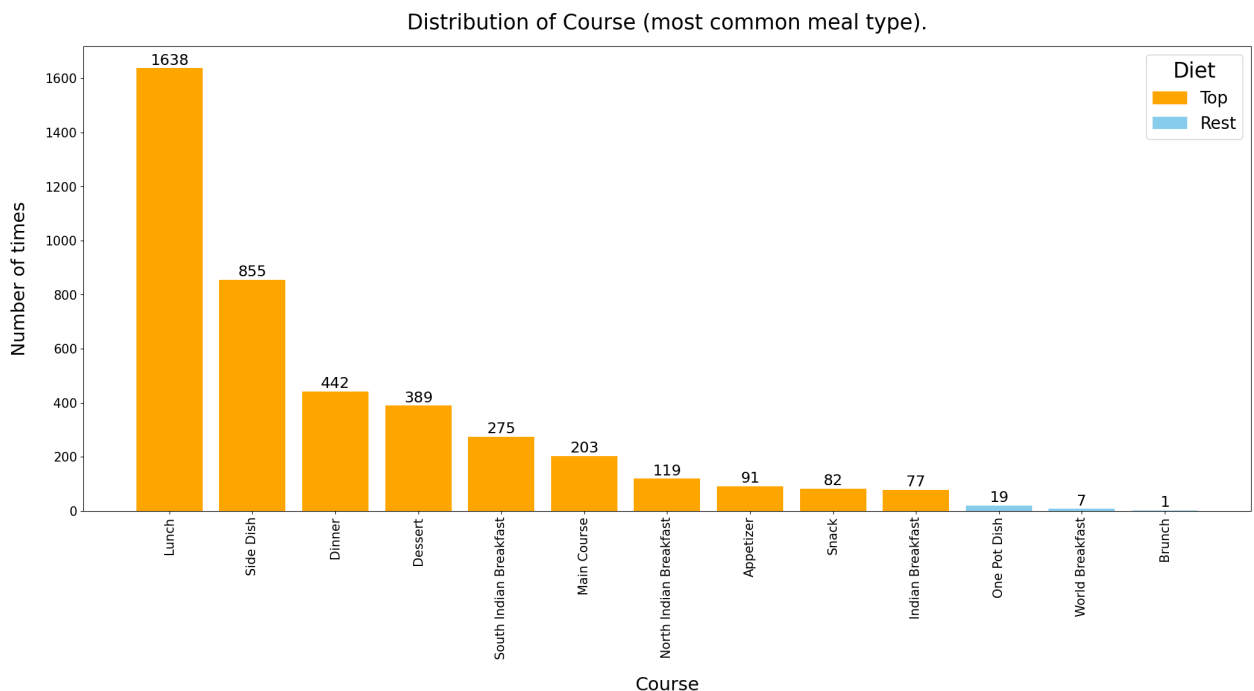
# Create legend manually
legend_handles = [
    mpatches.Patch(color="orange", label="Top"),
    mpatches.Patch(color="skyblue", label="Rest")
]

plt.legend(handles=legend_handles,
           title="Diet",
           title_fontsize=25, # title size
           fontsize=20,      # legend text size
           loc="upper right", # placement
           bbox_to_anchor=(1, 1)) # fine-tune position

# showing values at the top of the bar
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x()+bar.get_width()/2,height+2,str(height),ha="center",va="bottom")

plt.title("Distribution of Course (most common meal type).",fontsize=25,pad=20)
plt.xlabel("Course",fontsize=22, labelpad=20)
plt.ylabel("Number of times",fontsize=22, labelpad=20)
# plt.savefig("2.Distribution of course (most common meal type).png", bbox_inches='tight')
plt.show()

```



Insights from Course Distribution (Meal Types)

- **Lunch (1638 recipes)** is by far the most common meal type.
- **Side dishes (855)** and **Dinner (442)** are also very popular.
- **Desserts (389)** and **Breakfasts** (South Indian – 275, North Indian – 119) have a moderate share.
- **Brunch (1)** and **World Breakfast (7)** are extremely rare.
- The dataset mainly focuses on **lunch and dinner recipes**.

C. Count of diet.

```
In [281... data_diet = df.groupby('diet').count()['name']
print(":::Diet with counts:::")
print()
data_diet
```

:::Diet with counts:::

```
Out[281... diet
Diabetic Friendly          236
Eggetarian                 95
Gluten Free                36
High Protein Non Vegetarian 139
High Protein Vegetarian    514
No Onion No Garlic (Sattvic) 60
Non Vegetarian            240
Unknown                   37
Vegan                     21
Vegetarian                2858
Name: name, dtype: int64
```

```
In [282... #create DataFrame
df_diet = pd.DataFrame({
    'name':data_diet.index.tolist(), 'count':data_diet.values.tolist()
})
df_diet
```


Out[282...

	name	count
0	Diabetic Friendly	236
1	Eggetarian	95
2	Gluten Free	36
3	High Protein Non Vegetarian	139
4	High Protein Vegetarian	514
5	No Onion No Garlic (Sattvic)	60
6	Non Vegetarian	240
7	Unknown	37
8	Vegan	21
9	Vegetarian	2858

In [283...

```
df_diet_desc = df_diet.sort_values(by='count',ascending=False).reset_index(drop=True)
df_diet_desc
```

Out[283...

	name	count
0	Vegetarian	2858
1	High Protein Vegetarian	514
2	Non Vegetarian	240
3	Diabetic Friendly	236
4	High Protein Non Vegetarian	139
5	Eggetarian	95
6	No Onion No Garlic (Sattvic)	60
7	Unknown	37
8	Gluten Free	36
9	Vegan	21

In [284...

```
colors = ['orange' if i < 5 else 'skyblue' for i in range(len(df_diet_desc))]

plt.figure(figsize=(25,10))
bars = plt.bar(df_diet_desc['name'],df_diet_desc['count'],color=colors)

plt.xticks(rotation=90,fontsize=20)
plt.yticks(fontsize=20)

# Create legend manually
legend_handles = [
    mpatches.Patch(color="orange", label="Top"),
```

```

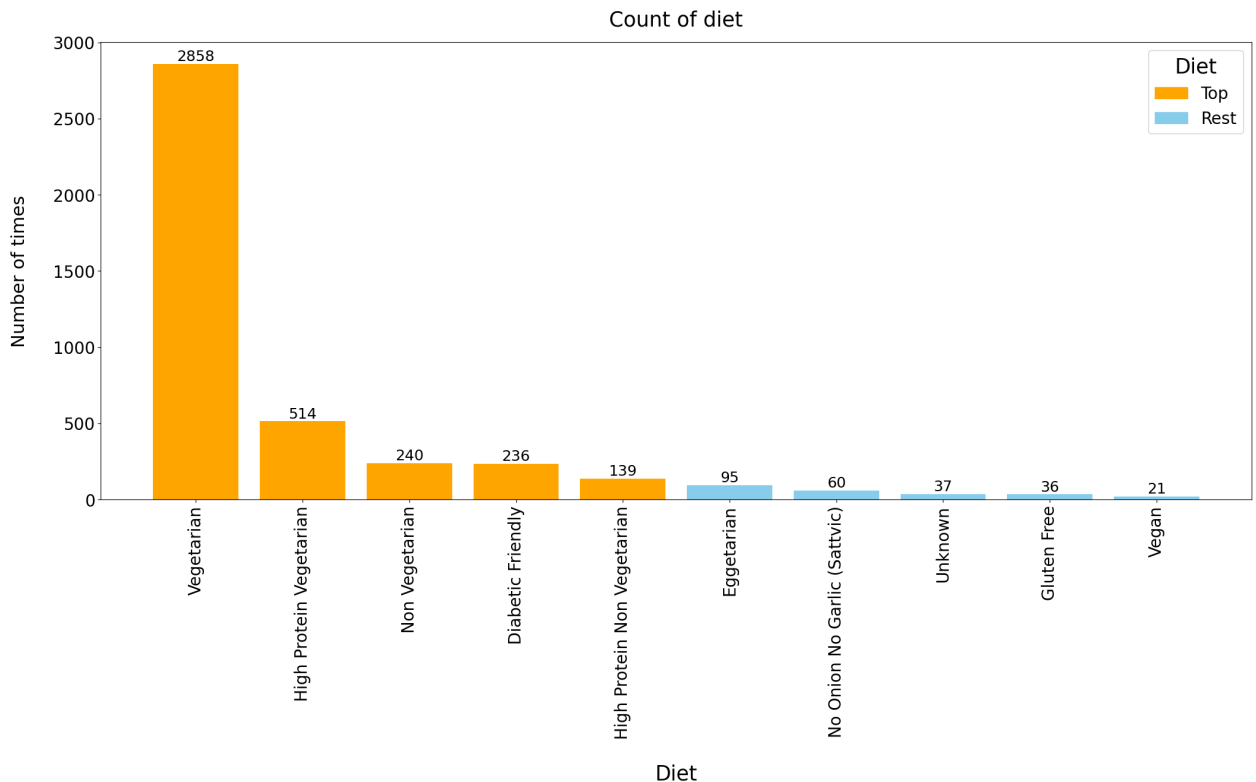
mpatches.Patch(color="skyblue", label="Rest")
]

plt.legend(handles=legend_handles,
           title="Diet",
           title_fontsize=25, # title size
           fontsize=20,      # legend text size
           loc="upper right", # placement
           bbox_to_anchor=(1, 1)) # fine-tune position

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x()+bar.get_width()/2,height,str(height),ha="center",va="

plt.title("Count of diet",fontsize=25,pad=20)
plt.xlabel("Diet",fontsize=25, labelpad=30)
plt.ylabel("Number of times",fontsize=22, labelpad=30)
# plt.savefig("3.Count of diet.png", bbox_inches='tight')
plt.show()

```



Insights from Diet Distribution

- The dataset is overwhelmingly **Vegetarian (2858 recipes)**.
- **High Protein Vegetarian (514)** also has a strong presence.
- Non-vegetarian options are limited: **Non-Veg (240)** and **High Protein Non-Veg (139)**.
- Special diets like **Diabetic Friendly (236)** and **Eggetarian (95)** exist

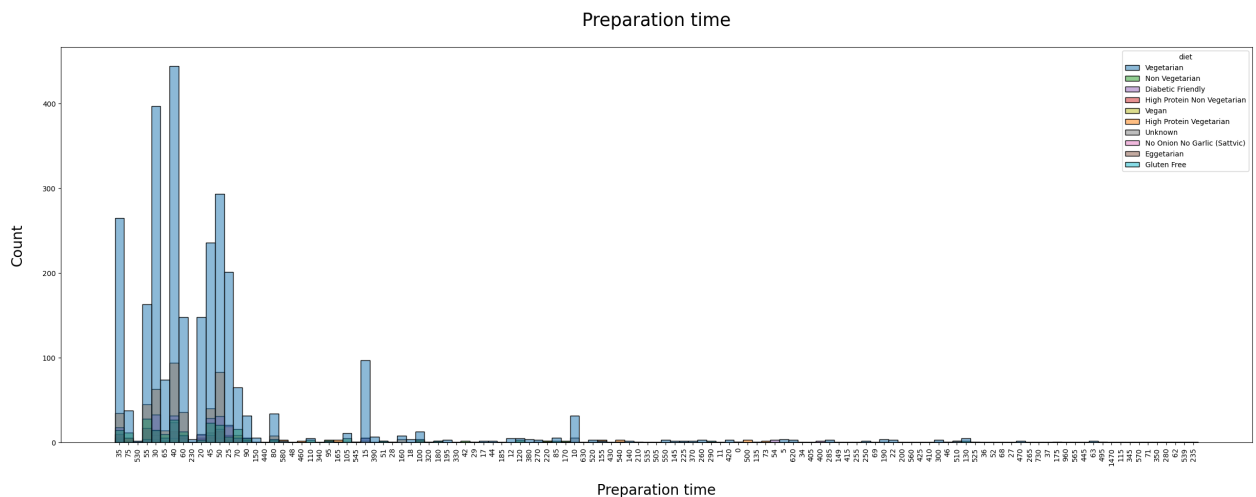
in smaller proportions.

- Niche categories like **Gluten Free (36)** and **Vegan (21)** are very limited.
- Overall, the dataset is **heavily vegetarian-biased**.

D. Distribution of preparation times (histogram).

```
In [285... custom_palette = {
    "Vegetarian": "#1f77b4",          # Blue
    "High Protein Vegetarian": "#ff7f0e", # Orange
    "Non Vegetarian": "#2ca02c",       # Green
    "High Protein Non Vegetarian": "#d62728", # Red
    "Diabetic Friendly": "#9467bd",    # Purple
    "Eggetarian": "#8c564b",          # Brown
    "No Onion No Garlic (Sattvic)": "#e377c2", # Pink
    "Unknown": "#7f7f7f",            # Gray
    "Vegan": "#bcbd22",               # Olive
    "Gluten Free": "#17becf"         # Cyan
}

plt.figure(figsize=(30,10))
plt.xticks(rotation=90,fontsize=10)
plt.title("Preparation time",fontsize=25,pad=30)
sns.histplot(data=df,x='prep_time',hue='diet', palette=custom_palette)
plt.xlabel("Preparation time",fontsize=20,labelpad=25)
plt.ylabel("Count",fontsize=20,labelpad=25)
plt.savefig("4.Preparation time.png", bbox_inches='tight')
plt.show()
```



Insights from Preparation Time Distribution

- Most recipes fall within **20-70 minutes** of preparation time.
- The highest concentration of recipes is around **35-55 minutes**.
- Recipes with preparation times longer than **100 minutes** are much less

common.

- Very long preparation times (**200+ minutes**) are rare outliers.
- All diet types (Vegetarian, Non-Vegetarian, Vegan, etc.) follow a similar pattern, clustering in the shorter preparation times.

In [286... *# Univariate Analysis Report*

```
html_code = """
<h1 style="text-align:center;">Food Dataset Insights</h1>

<!-- Cuisine Distribution -->
<h2 style="color:#ff6600;">1. Distribution of Cuisines</h2>

  - <b>North Indian, South Indian, and Indian recipes</b> dominate with over
  - These three categories alone form the majority of recipes.<br>
  - <b>Bengali, Tamil Nadu, Kerala, and Maharashtrian</b> cuisines are moder
  - <b>Thai, Afghan, African</b> cuisines are rare (1–5 recipes).<br>
  - Overall, the dataset is highly skewed towards Indian regional cuisines.
</p>

<!-- Course Distribution -->
<h2 style="color:#ff6600;">2. Distribution of Courses (Meal Types)</h2>

  - <b>Lunch (1638 recipes)</b> is the most common meal type.<br>
  - <b>Side dishes (855)</b> and <b>Dinner (442)</b> are also popular.<br>
  - <b>Desserts (389)</b> and <b>Breakfasts</b> (South Indian – 275, North I
  - Rare types: <b>Brunch (1)</b>, <b>World Breakfast (7)</b>.<br>
  - The dataset mainly focuses on <b>lunch and dinner</b>.
</p>

<!-- Diet Distribution -->
<h2 style="color:#ff6600;">3. Distribution of Diet Types</h2>

  - Majority are <b>Vegetarian (2858 recipes)</b>.<br>
  - <b>High Protein Vegetarian (514)</b> is also significant.<br>
  - Non-vegetarian: <b>Non-Veg (240)</b>, <b>High Protein Non-Veg (139)</b>.
  - Special diets: <b>Diabetic Friendly (236)</b>, <b>Eggetarian (95)</b>.<br>
  - Rare: <b>Gluten Free (36)</b>, <b>Vegan (21)</b>.<br>
  - Overall, the dataset is <b>heavily vegetarian-biased</b>.
</p>

<!-- Diet Distribution -->
<h2 style="color:#ff6600;">Insights from Preparation Time Distribution</h2>

  - Majority are <b>Vegetarian (2858 recipes)</b>.<br>
  - <b>High Protein Vegetarian (514)</b> is also significant.<br>
  - Non-vegetarian: <b>Non-Veg (240)</b>, <b>High Protein Non-Veg (139)</b>.
```

```

- Special diets: <b>Diabetic Friendly (236)</b>, <b>Eggetarian (95)</b>.<b>
- Rare: <b>Gluten Free (36)</b>, <b>Vegan (21)</b>.<br>
- Overall, the dataset is <b>heavily vegetarian-biased</b>.
</p>

<!-- Preparation Time Distribution -->
<h2 style="color:#ff6600;">Insights from Preparation Time Distribution</h2>
<img src="http://localhost:8888/files/4.Preparation%20time.png?_xsrf=2%7Cb1051
<p>
- Most recipes fall within <b>20–70 minutes</b> of preparation time.<br>
- The highest concentration of recipes is around <b>35–55 minutes</b>.<br>
- Recipes longer than <b>100 minutes</b> are much less common.<br>
- Very long preparation times (<b>200+ minutes</b>) are rare outliers.<br>
- All diet types (Vegetarian, Non-Vegetarian, Vegan, etc.) follow a simila
</p>

"""

# display(HTML(html_code))

```

4. Bivariate Analysis

A. Compare average preparation time across cuisines.

```
In [287... # df.head(5)
```

```
In [288... df['prep_time'] = df['prep_time'].astype('float32') #convert prep_time to float
```

```
In [289... data_cuisine_average = df.groupby('cuisine').agg({
    "prep_time": "mean"
})

avg_prep_time=[]

# data_cuisine_average.values.tolist()
for val in data_cuisine_average.values.tolist():
    for v in val:
        avg_prep_time.append(round(v,2))

data_cuisine = pd.DataFrame({
    "name" : data_cuisine_average.index.tolist(),
    "avg_time" : avg_prep_time
})

data_cuisine_desc = data_cuisine.sort_values(by='avg_time',ascending=False).re
colors = ['orange' if i < 10 else 'skyblue' for i in range(len(data_cuisine_de
plt.figure(figsize=(30,10))

```

```

# Create legend manually
legend_handles = [
    mpatches.Patch(color="orange", label="Top"),
    mpatches.Patch(color="skyblue", label="Rest")
]

plt.legend(handles=legend_handles,
           title="Avg Preparation Time",
           title_fontsize=25, # title size
           fontsize=20,      # legend text size
           loc="upper right", # placement
           bbox_to_anchor=(1, 1)) # fine-tune position

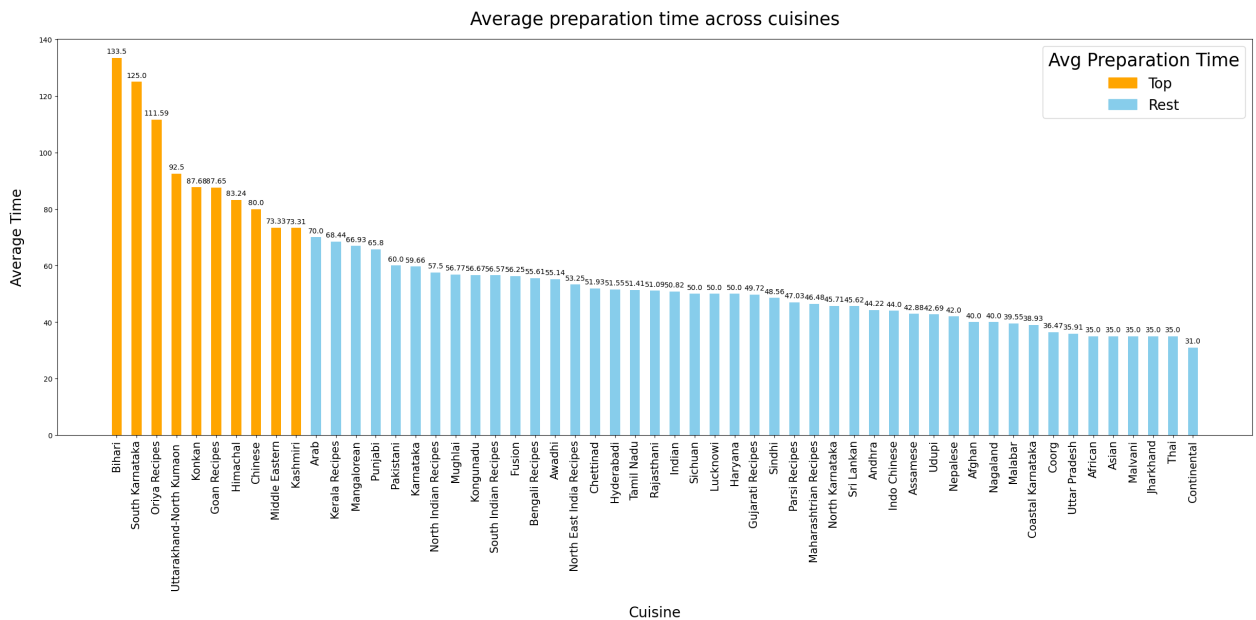
plt.xticks(rotation=90, fontsize=15)

bars = plt.bar(data_cuisine_desc['name'], data_cuisine_desc['avg_time'], color=c

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x()+bar.get_width()/2, height+1, str(height), ha="center", va

plt.title("Average preparation time across cuisines", fontsize=25, pad=20)
plt.xlabel("Cuisine", fontsize=20, labelpad=20)
plt.ylabel("Average Time", fontsize=20, labelpad=20)
plt.savefig("Bivariate Analysis/1. Compare average preparation time across cui
plt.show()

```



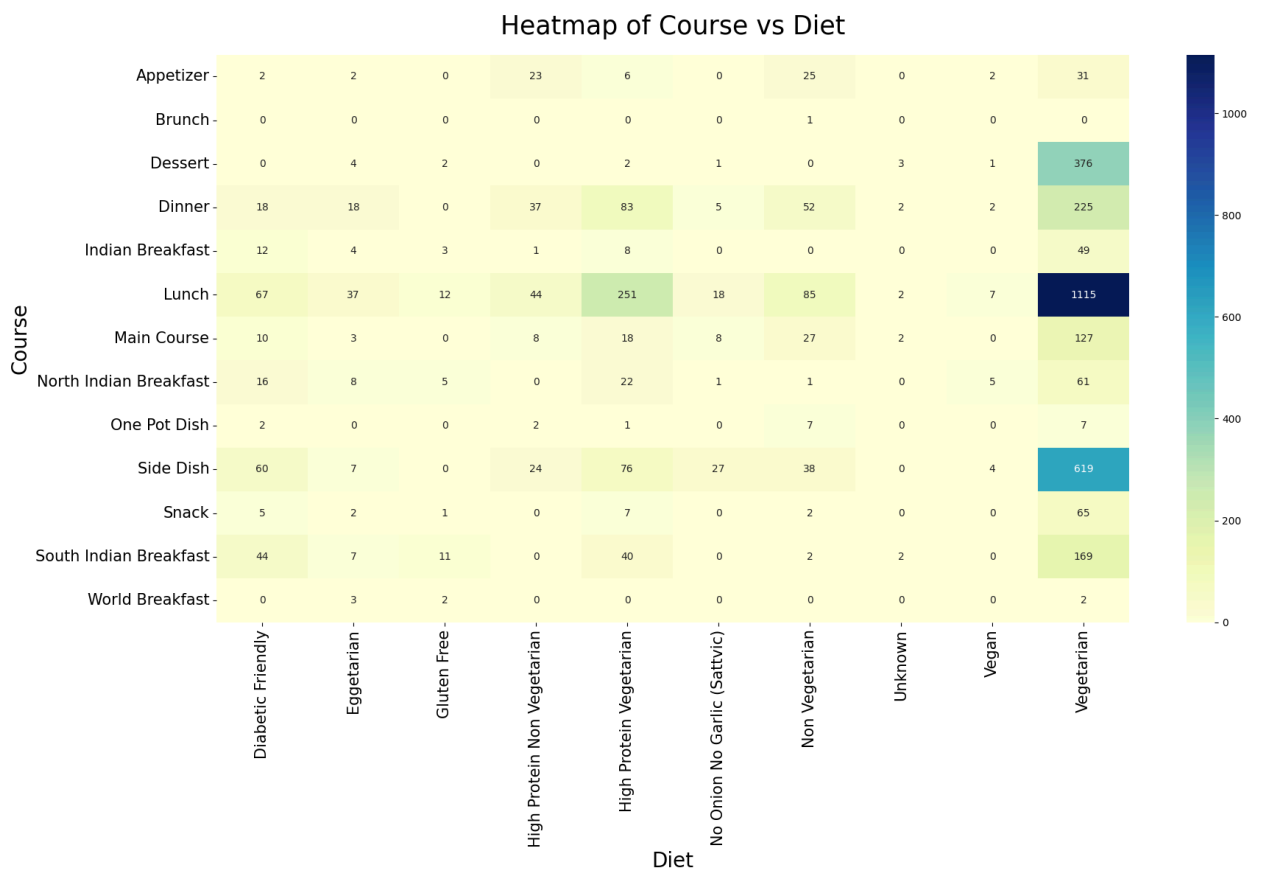
B. Check relationship between course and diet.

```

In [290... relationship = pd.crosstab(df['course'], df['diet'])
# print(relationship)

plt.figure(figsize=(25,10))

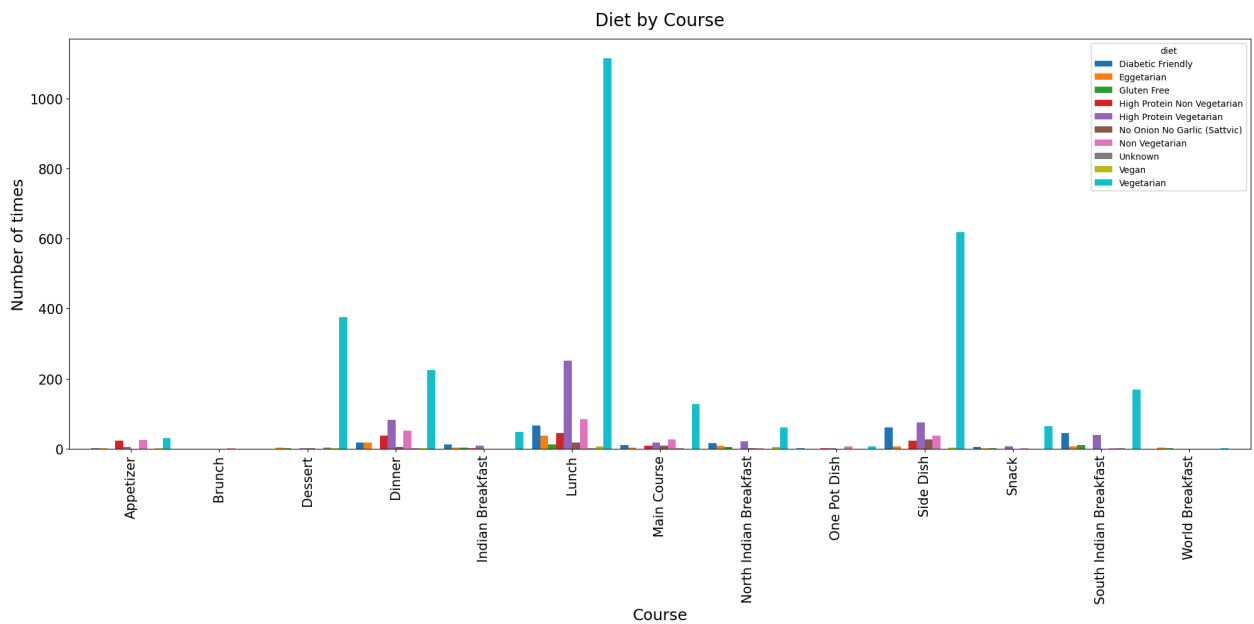
```

```
In [291... fig, ax = plt.subplots(figsize=(20, 10), dpi=100)
relationship.plot(kind='bar', stacked=False, ax=ax, width=0.9)

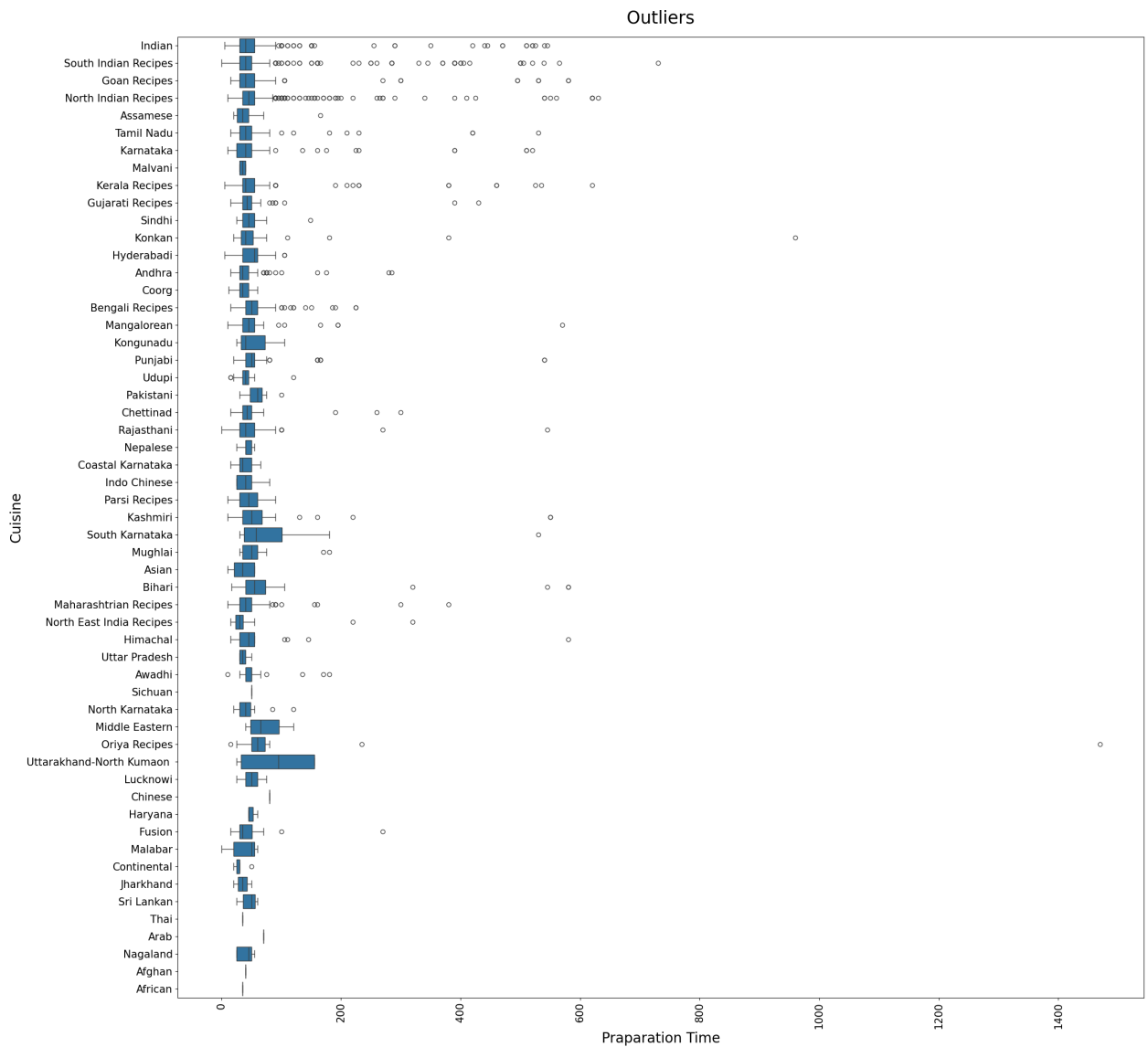
ax.set_title('Diet by Course', fontsize=20, pad=15)
ax.set_xlabel('Course', fontsize=18)
ax.set_ylabel('Number of times', fontsize=18)
ax.tick_params(axis='x', rotation=90, labelsize=15)
ax.tick_params(axis='y', labelsize=15)

plt.tight_layout()
# plt.savefig("Bivariate Analysis/2.2. Diet by Course.png", bbox_inches="tight")
plt.show()
```

5. Outlier & Anomaly Detection

```
In [292... plt.figure(figsize=(25,25))
plt.xticks(rotation=90, fontsize=15)
plt.yticks(fontsize=15)
plt.title('Outliers', fontsize=25,pad=20)
sns.boxplot(df,x='prep_time',y='cuisine')
plt.xlabel("Praparation Time", fontsize=20)
plt.ylabel("Cuisine", fontsize=20)
# plt.savefig("1. Outliiers.png",bbox_inches="tight")
plt.show()
```



In [293... *# Removing outliers*

```
Q1 = df['prep_time'].quantile(0.25)
```

```
Q3 = df['prep_time'].quantile(0.75)
```

```
IQR = Q3 - Q1
```

```
lower_bound = Q1 - 1.5 * IQR
```

```
upper_bound = Q3 + 1.5 * IQR
```

```
df_no_outliers = df[(df['prep_time'] >= lower_bound)&(df['prep_time'] <= upper_bound)]
```

```
print("\n1. Original Shape:",df.shape[0])
```

```
print("\n2. No Outliers Shape:",df_no_outliers.shape[0])
```

```
print("\n3. Number of outliers:",df.shape[0]-df_no_outliers.shape[0])
```

```
print("\nAfter Removing Outliers:\n")
```

```
df_no_outliers
```

1. Original Shape: 4236
2. No Outliers Shape: 3962
3. Number of outliers: 274

After Removing Outliers:

Out[293...

	name	image_url	description	cuisine	course
0	Thayir Semiya Recipe (Curd Semiya)	https://www.archanaskitchen.com/images/archana...	Thayir Semiya or Curd Vermicelli is a quick di...	Indian	Lunch
1	Chettinad Style Kara Kuzhambu Recipe with Pota...	https://www.archanaskitchen.com/images/archana...	Chettinad Style Kara Kuzhambu Recipe with Pot...	South Indian Recipes	Lunch
3	Minced Meat And Egg Croquettes Recipe	https://www.archanaskitchen.com/images/archana...	The croquette is usually cigar shaped or cylin...	North Indian Recipes	Appetizer
4	Thekera Tenga Recipe	https://www.archanaskitchen.com/images/archana...	Thekera Tenga is a Sour curry from Assamese cu...	Assamese	Dinner
5	Spicy Cabbage Rice Recipe (South Indian Style ...	https://www.archanaskitchen.com/images/archana...	Spicy Cabbage Rice takes very few ingredients ...	Indian	Lunch
...
4231	Stir Fry Green beans and Tofu with Panch Phoro...	https://www.archanaskitchen.com/images/archana...	Stir Fry Green beans and Tofu with Panch Phoro...	Bengali Recipes	Side Dish
4232	Dhuska Recipe (Rice Fried Bread)	https://www.archanaskitchen.com/images/archana...	Dhuska is a traditional recipe from Jharkhand...	Jharkhand	Snack
4233	Khatta Meetha Petha Recipe (Yellow Pumpkin Sabzi)	https://www.archanaskitchen.com/images/archana...	Khatta Meetha Petha or sweet and sour yellow p...	Indian	Lunch
4234	Patta Gobi Matar Nu	https://www.archanaskitchen.com/images/archana...	Patta Gobi Matar Nu	North Indian	Lunch

	name	image_url	description	cuisine	course
	Shaak Recipe (Cabbage & Pe...		Shaak Recipe (Cabbage & Pe...	Recipes	
4235	Kerala Pumpkin Pachadi Recipe (Parangikai Pach...	https://www.archanaskitchen.com/images/archana...	The Kerala Pumpkin Pachadi Recipe, is an Ayurv...	Kerala Recipes	Lunch

3962 rows × 9 columns

6. Visualizations

```
In [294... categorical_columns = ['cuisine', 'course', 'diet']

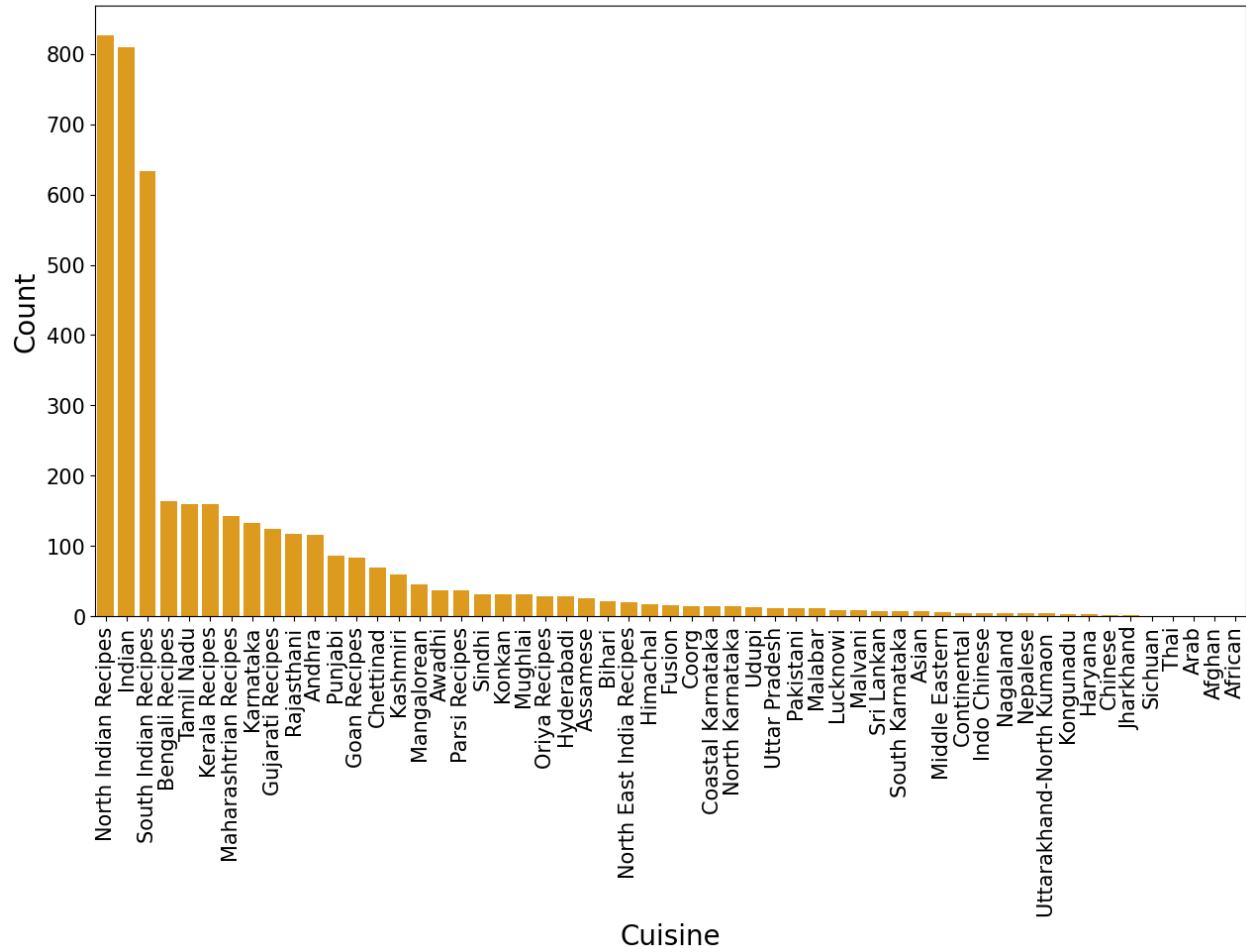
for col in categorical_columns:
    plt.figure(figsize=(15,8))

    ax = sns.countplot(
        data=df, x=col, order=df[col].value_counts().index, color='orange',
        hue=None, legend=False
    )

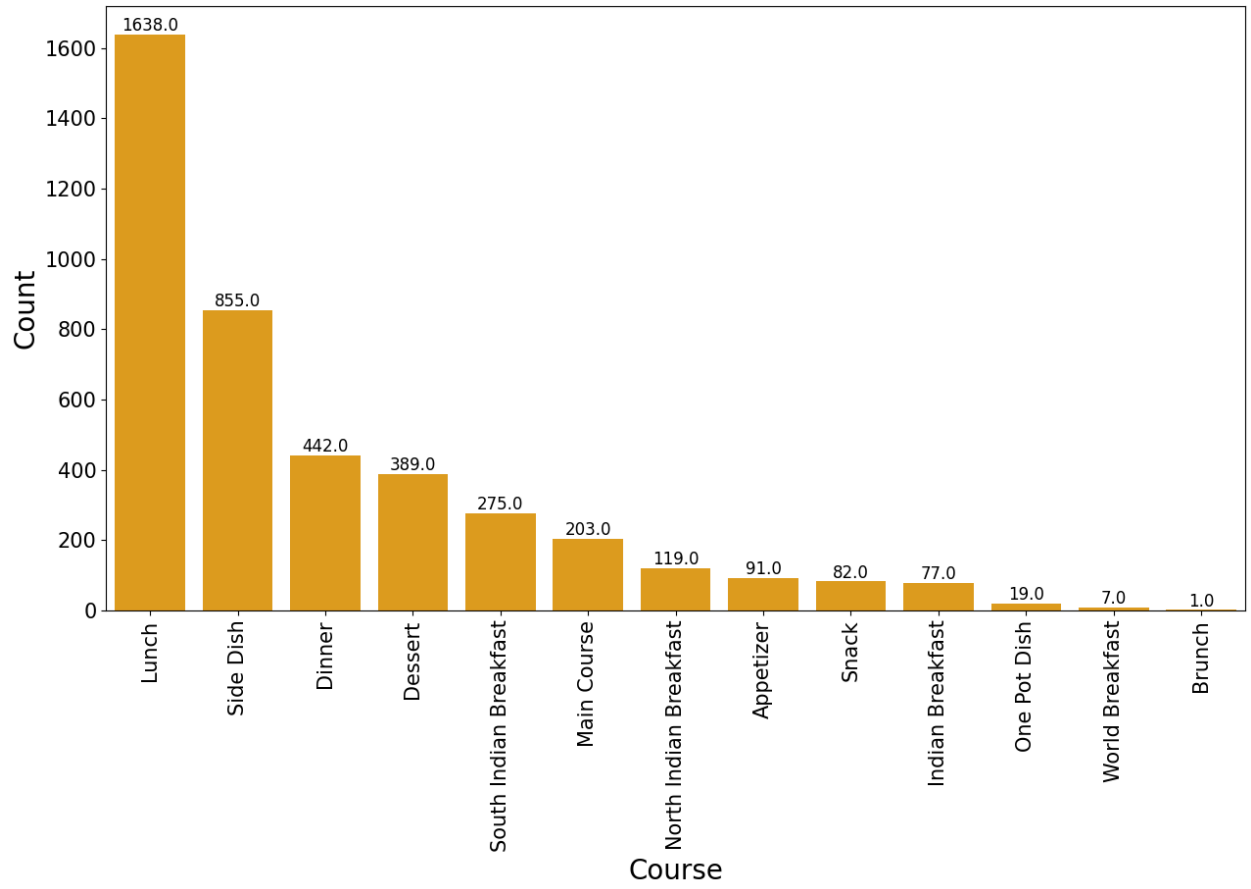
    if col in ['course', 'diet']:
        for p in ax.patches:
            height = p.get_height()
            ax.annotate(str(height), (p.get_x() + p.get_width()/2., height), ha="

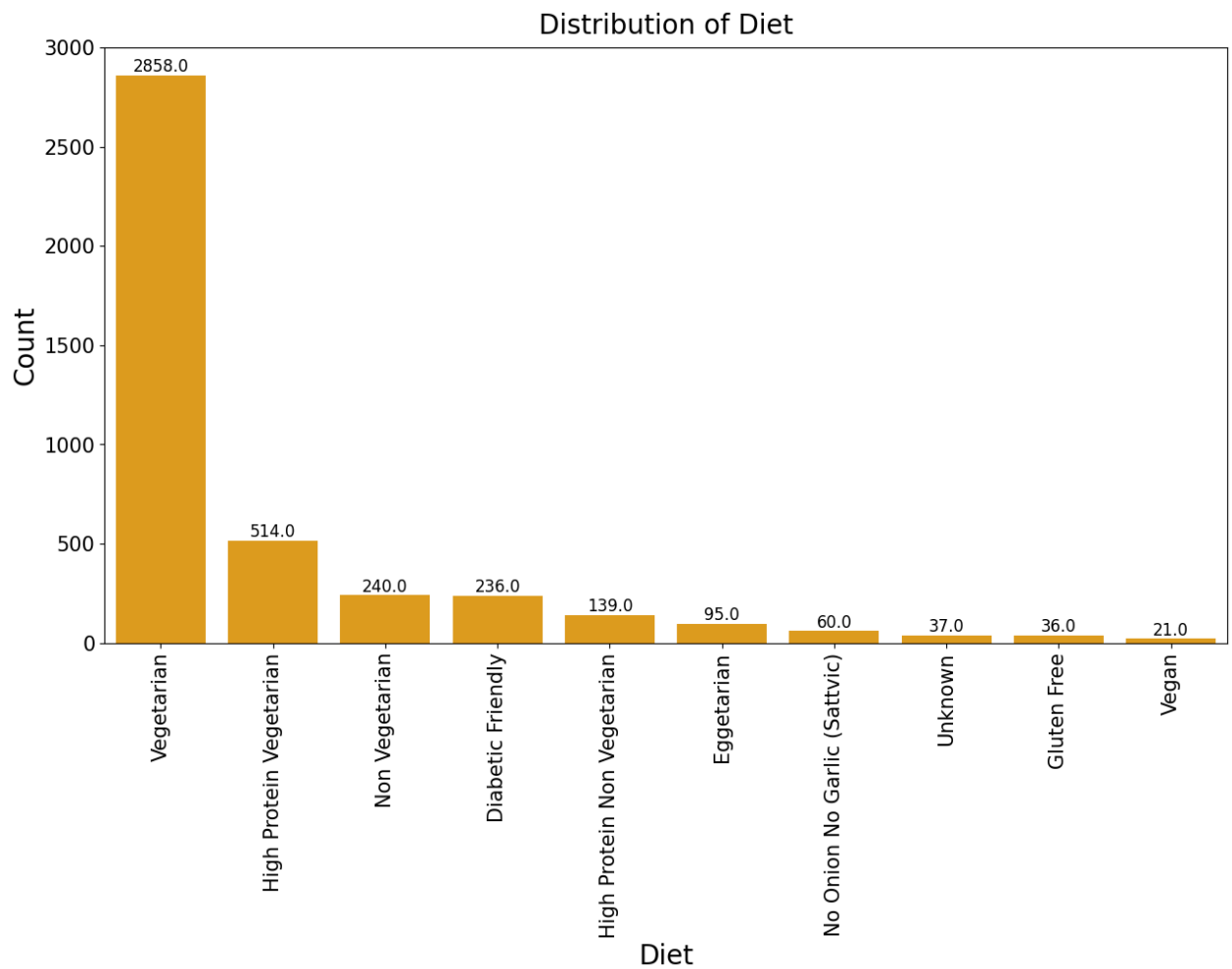
plt.title(f"Distribution of {col.capitalize()}", fontsize=20, pad=10)
plt.xlabel(col.capitalize(), fontsize=20)
plt.ylabel("Count", fontsize=20)
plt.xticks(rotation=90, fontsize=15)
plt.yticks(fontsize=15)
# plt.savefig(f"1. Distribution of {col.capitalize()}.png", bbox_inches="ti
plt.show()
```

Distribution of Cuisine



Distribution of Course





7. Insights & Summary

Key Findings

- North Indian, South Indian, and Indian recipes dominate with over 600–800 entries each.
- Vegetarian dishes are more common at the Lunch course.
- Most recipes fall within 20–70 minutes of preparation time.

END