Assignment 5: Complete data visualization on Mushroom dataset using Matplotlib and Seaborn libraries

In [2]:
```python
# Data Manipulation
import numpy as np
import pandas as pd

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```python
# Load the dataset

headers = ['Class', 'CapShape',
           'CapSurface', 'CapColor',
           'Bruises', 'Odor',
           'GillAttachment','GillSpacing',
           'GillSize','GillColor',
           'StalkShape', 'StalkRoot',
           'StalkAboveRing', 'StalkBelowRing',
           'ColorAboveRing', 'ColorBelowRing',
           'VeilType', 'VeilColor',
           'RingNumber', 'RingType',
           'SporeColor','Population',
           'Habitat']

raw_data = pd.read_csv("agaricus-lepiota.data",
                       header = None,
                       names = headers,
                       sep = ',',
                       na_values=["?"],
                       engine='python')
raw_data.head()
```

Out[5]:

	Class	CapShape	CapSurface	CapColor	Bruises	Odor	GillAttachment	GillSpacing	GillSize	GillColor	...	StalkBelowRing
0	p	x	s	n	t	p	f	c	n	k	...	s
1	e	x	s	y	t	a	f	c	b	k	...	s
2	e	b	s	w	t	l	f	c	b	n	...	s
3	p	x	y	w	t	p	f	c	n	n	...	s
4	e	x	s	g	f	n	f	w	b	k	...	s

5 rows × 23 columns

In [25]:
```python
raw_data.shape
```

Out[25]:
```
(8124, 23)
```

In [26]:
```python
raw_data.dtypes
```

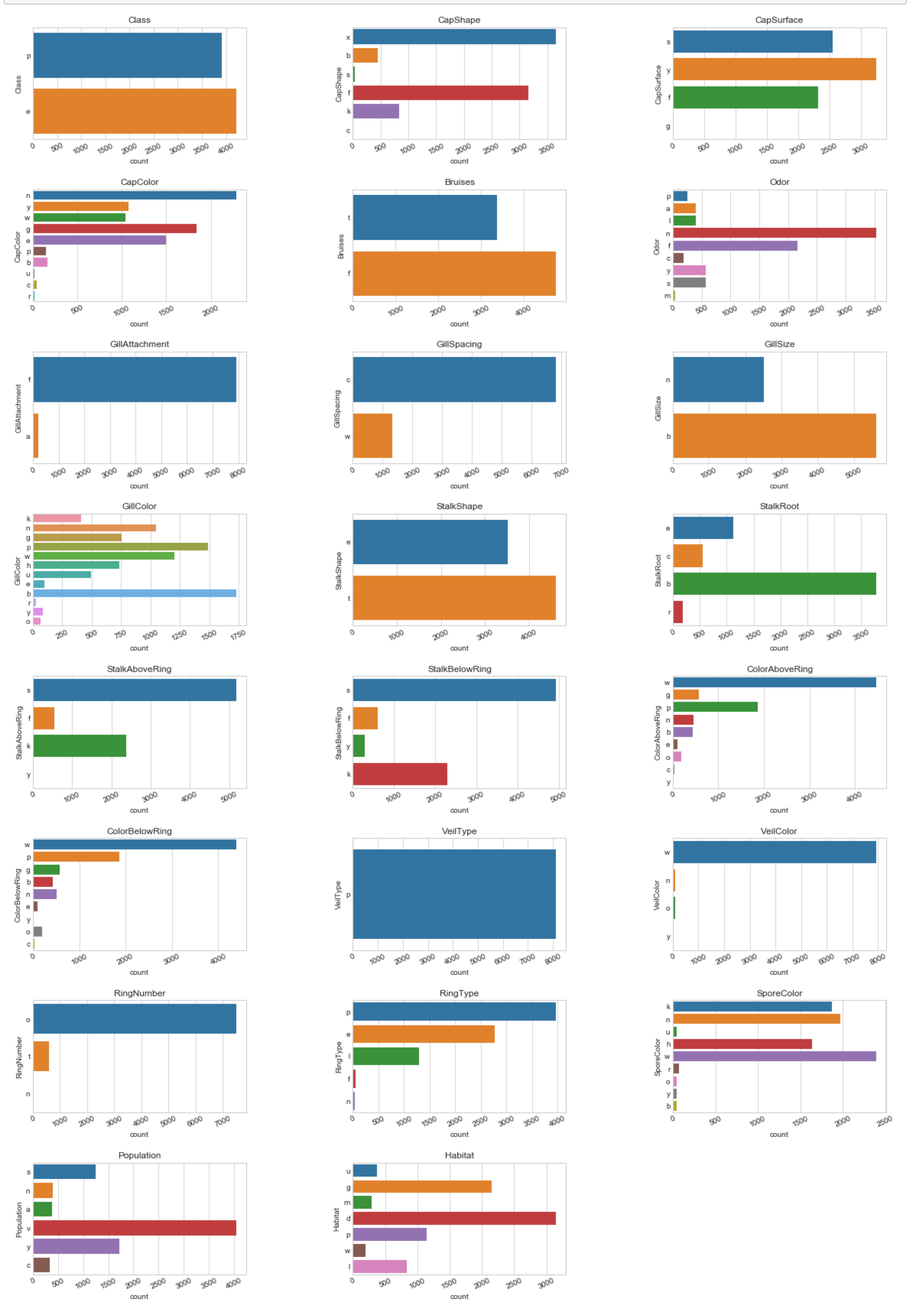
Out[26]:
```
Class             object
CapShape          object
CapSurface        object
CapColor          object
Bruises           object
Odor              object
GillAttachment    object
GillSpacing       object
GillSize          object
GillColor         object
StalkShape        object
StalkRoot         object
StalkAboveRing    object
StalkBelowRing    object
ColorAboveRing    object
ColorBelowRing    object
VeilType          object
VeilColor         object
RingNumber        object
RingType          object
SporeColor        object
Population        object
Habitat           object
dtype: object
```

In [7]:
```python
raw_data['Class'].value_counts()
```

Out[7]:
```
e    4208
p    3916
Name: Class, dtype: int64
```

In [8]:
```python
plt.figure()
pd.Series(raw_data['Class']).value_counts().sort_index().plot(kind = 'bar')
plt.ylabel("Count")
plt.xlabel("Class")
plt.title('Number of poisonous/edible mushrooms (0=edible, 1=poisonous)');
```

In [12]:
```python
import math
# Let's plot the distribution of each feature
def plot_distribution(dataset, cols=5, width=20, height=15, hspace=0.2, wspace=0.5):
    plt.style.use('seaborn-whitegrid')
    fig = plt.figure(figsize=(width,height))
    fig.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=wspace, hspace=
hspace)
    rows = math.ceil(float(dataset.shape[1]) / cols)
    for i, column in enumerate(dataset.columns):
        ax = fig.add_subplot(rows, cols, i + 1)
        ax.set_title(column)
        if dataset.dtypes[column] == np.object:
            g = sns.countplot(y=column, data=dataset)
            substrings = [s.get_text()[:18] for s in g.get_yticklabels()]
            g.set(yticklabels=substrings)
            plt.xticks(rotation=25)
        else:
            g = sns.distplot(dataset[column])
            plt.xticks(rotation=25)

plot_distribution(raw_data, cols=3, width=20, height=30, hspace=0.45, wspace=0.5)
```

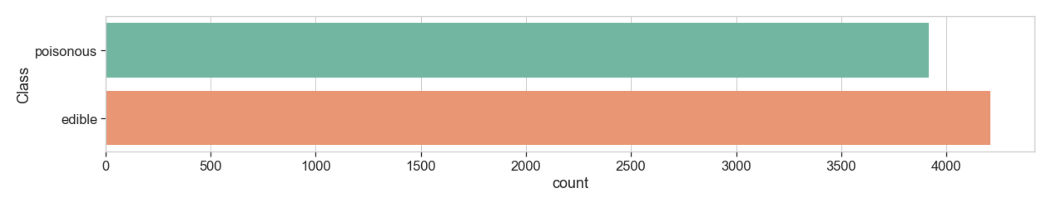


Feature: Class

In [16]:
```python
# Let's fix the Class Feature
raw_data.loc[raw_data['Class'] == 'e', 'Class'] = 1
raw_data.loc[raw_data['Class'] == 'p', 'Class'] = 0

raw_data['Class'] = raw_data['Class']
```

In [20]:
```python
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,3))
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="Class", data=raw_data, palette="Set2");
g.set_yticklabels(['poisonous','edible'])
```

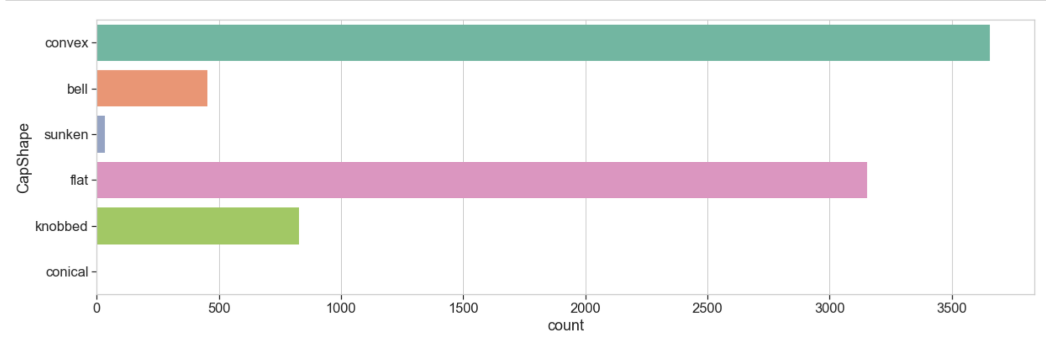
Out[20]:
```
[Text(0, 0, 'poisonous'), Text(0, 0, 'edible')]
```



Feature: CapShape bell=b,conical=c,convex=x,flat=f,knobbed=k,sunken=s

In [21]:
```python
raw_data.loc[raw_data['CapShape'] == 'b', 'CapShape'] = 'bell'
raw_data.loc[raw_data['CapShape'] == 'c', 'CapShape'] = 'conical'
raw_data.loc[raw_data['CapShape'] == 'x', 'CapShape'] = 'convex'
raw_data.loc[raw_data['CapShape'] == 'f', 'CapShape'] = 'flat'
raw_data.loc[raw_data['CapShape'] == 'k', 'CapShape'] = 'knobbed'
raw_data.loc[raw_data['CapShape'] == 's', 'CapShape'] = 'sunken'
```

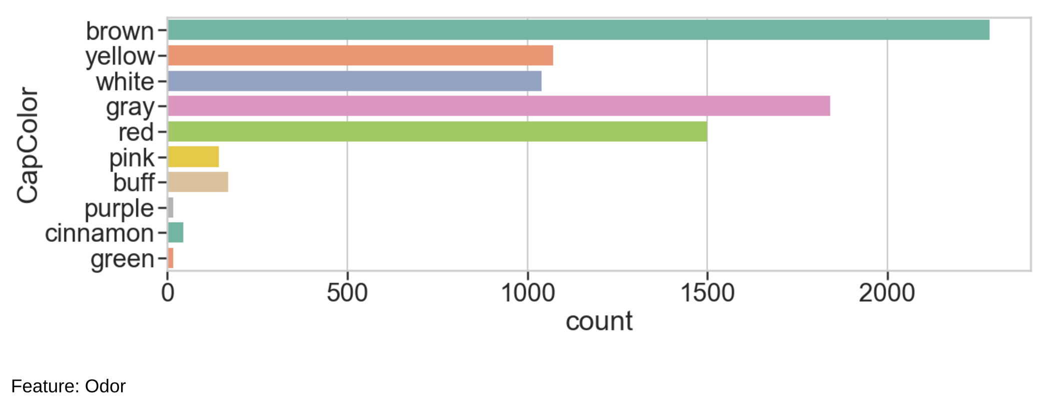
In [23]:
```python
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,6))
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="CapShape", data=raw_data, palette="Set2");
```



Feature: CapColor brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y

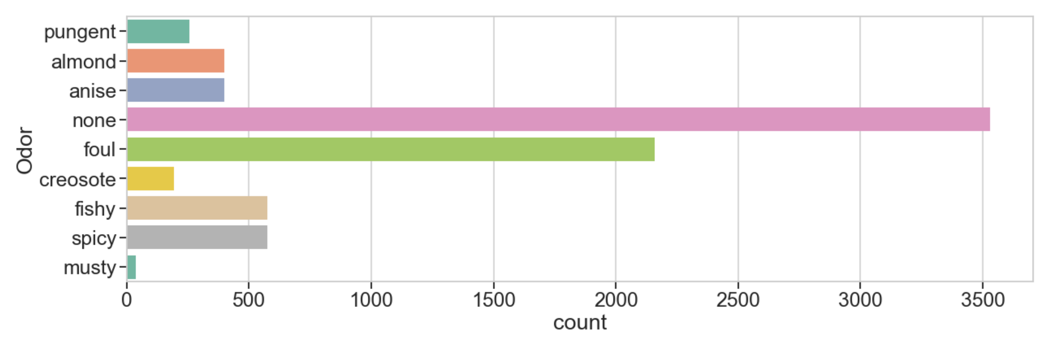
In [24]:
```python
raw_data.loc[raw_data['CapColor'] == 'n', 'CapColor'] = 'brown'
raw_data.loc[raw_data['CapColor'] == 'b', 'CapColor'] = 'buff'
raw_data.loc[raw_data['CapColor'] == 'c', 'CapColor'] = 'cinnamon'
raw_data.loc[raw_data['CapColor'] == 'g', 'CapColor'] = 'gray'
raw_data.loc[raw_data['CapColor'] == 'r', 'CapColor'] = 'green'
raw_data.loc[raw_data['CapColor'] == 'p', 'CapColor'] = 'pink'
raw_data.loc[raw_data['CapColor'] == 'u', 'CapColor'] = 'purple'
raw_data.loc[raw_data['CapColor'] == 'e', 'CapColor'] = 'red'
raw_data.loc[raw_data['CapColor'] == 'w', 'CapColor'] = 'white'
raw_data.loc[raw_data['CapColor'] == 'y', 'CapColor'] = 'yellow'
```

In [25]:
```python
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,6))
sns.set_context("poster", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="CapColor", data=raw_data, palette="Set2");
```



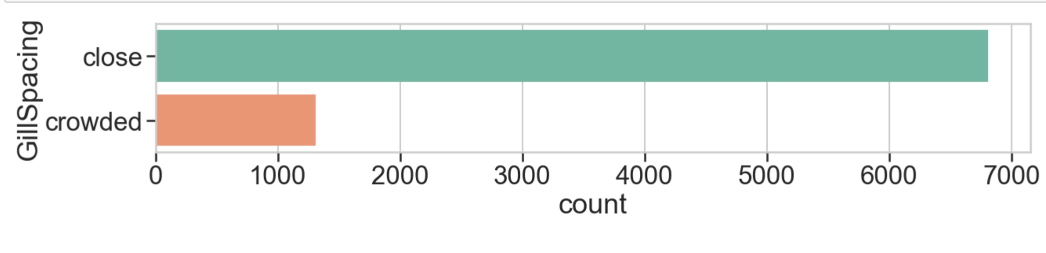
Feature: Odor

In [26]:
```python
raw_data.loc[raw_data['Odor'] == 'a', 'Odor'] = 'almond'
raw_data.loc[raw_data['Odor'] == 'l', 'Odor'] = 'anise'
raw_data.loc[raw_data['Odor'] == 'c', 'Odor'] = 'creosote'
raw_data.loc[raw_data['Odor'] == 'y', 'Odor'] = 'fishy'
raw_data.loc[raw_data['Odor'] == 'f', 'Odor'] = 'foul'
raw_data.loc[raw_data['Odor'] == 'm', 'Odor'] = 'musty'
raw_data.loc[raw_data['Odor'] == 'n', 'Odor'] = 'none'
raw_data.loc[raw_data['Odor'] == 'p', 'Odor'] = 'pungent'
raw_data.loc[raw_data['Odor'] == 's', 'Odor'] = 'spicy'


plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,6))
sns.set_context("talk", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="Odor", data=raw_data, palette="Set2");
```



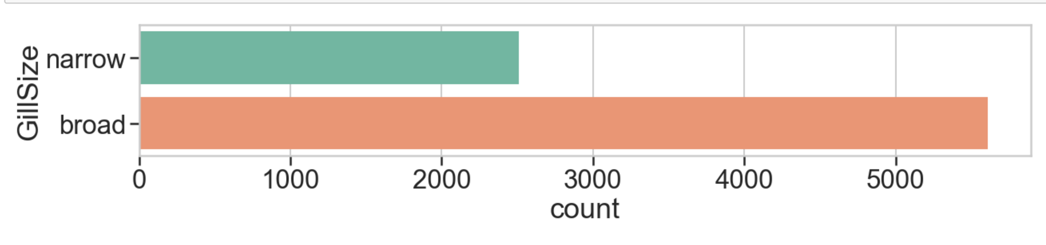
Feature: GillAttachment attached=a,descending=d,free=f,notched=n

In [27]:
```python
raw_data.loc[raw_data['GillSpacing'] == 'c', 'GillSpacing'] = 'close'
raw_data.loc[raw_data['GillSpacing'] == 'w', 'GillSpacing'] = 'crowded'
raw_data.loc[raw_data['GillSpacing'] == 'd', 'GillSpacing'] = 'distant'
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,3))
sns.set_context("poster", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="GillSpacing", data=raw_data, palette="Set2");
```



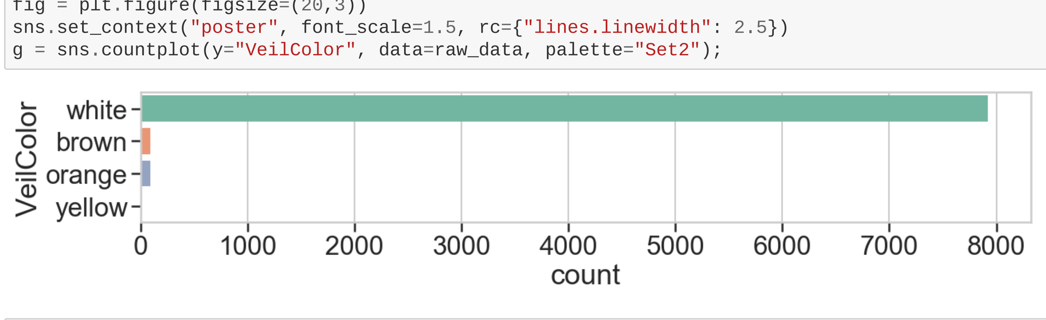
Feature: GillSize broad=b,narrow=n

In [28]:
```python

raw_data.loc[raw_data['GillSize'] == 'b', 'GillSize'] = 'broad'
raw_data.loc[raw_data['GillSize'] == 'n', 'GillSize'] = 'narrow'
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,3))
sns.set_context("poster", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="GillSize", data=raw_data, palette="Set2");
```



Feature: VeilColor brown=n,orange=o,white=w,yellow=y

In [29]:
```python

raw_data.loc[raw_data['VeilColor'] == 'n', 'VeilColor'] = 'brown'
raw_data.loc[raw_data['VeilColor'] == 'o', 'VeilColor'] = 'orange'
raw_data.loc[raw_data['VeilColor'] == 'w', 'VeilColor'] = 'white'
raw_data.loc[raw_data['VeilColor'] == 'y', 'VeilColor'] = 'yellow'
plt.style.use('seaborn-whitegrid')
fig = plt.figure(figsize=(20,3))
sns.set_context("poster", font_scale=1.5, rc={"lines.linewidth": 2.5})
g = sns.countplot(y="VeilColor", data=raw_data, palette="Set2");
```



In [ ]: