Producing publication-ready plots with seaborn

View solution HTML

In this project, we will practice making creating report-ready plots with seaborn by performing exploratory data analysis (EDA) on a large public health dataset. The Behavioral Risk Factor Surveillance System (BRFSS) is the largest long-term, ongoing health survey system in the world. Administered by the the CDC, the BRFSS conducts 400,000 telephone interviews with American adults each year. The survey asks participants a number of yes or no questions as well as a few questions that require a numerical answer. In this EDA we will explore relationships between just a few of the variables: mental health, physical health, sleep, and alcohol use. Feel free to explore any of the other variables you find interesting! This version of the data set was downloaded from Kaggle and has been cleaned and subset for this project.

Import packages and load data

1. Import `pandas`, `seaborn`, `matplotlib`, and `numpy`.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

To make sure that the plots will display properly in the notebook, run the provided code for `%matplotlib inline` and `plt.rcParams` settings.

In [2]:
```
%matplotlib inline
plt.rcParams['figure.dpi']=72
```

2. Load the BRFSS dataset stored in **BRFSS_sample.csv** and save it to the variable `health`. Next, view the first five lines of the data frame to get a better understanding of the shape of the dataset and of the different variables.

In [3]:
```
# load the dataset
health = pd.read_csv('BRFSS_sample.csv')
# print the first 5 rows
health.head()
```

Out[3]:

	HeartDisease	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWa
0	No	Yes	No	No	0	0	
1	No	No	No	No	3	0	
2	No	No	No	No	10	23	
3	No	No	No	No	0	0	
4	No	Yes	No	No	0	0	

Looking at the head of this dataframe, we can see that there are many 'yes' or 'no' survey questions about the participants' behaviors and health status. There are also a few questions with categorical responses, such as `AgeCategory` and `GenHealth`. Importantly, there three questions with numerical responses:

- `MentalHealth`: The number of days in the last month that a participant experienced negative mental health symptoms
- `PhysicalHealth`: The number of days in the last month that a participant experienced physical illness or injury
- `SleepTime`: The average number of hours participants report sleeping each night

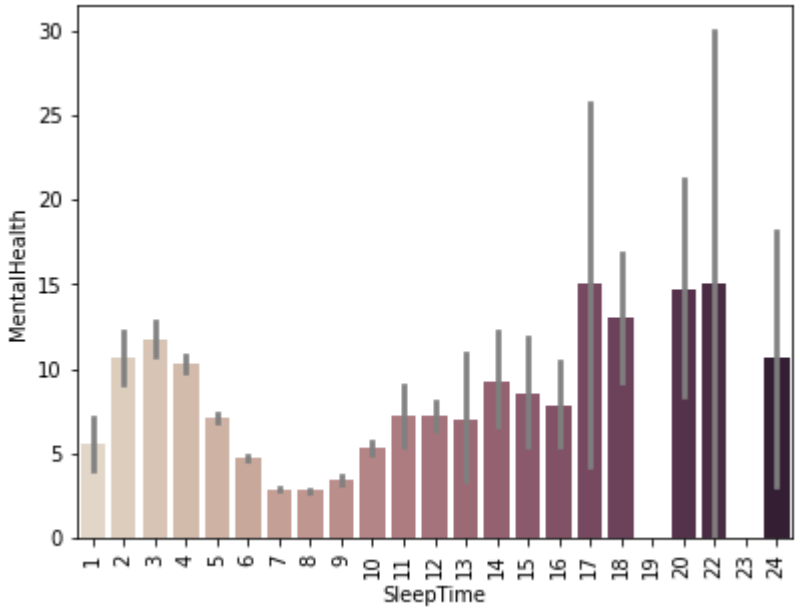
We will look at how these three variables relate to one another and then consider how `AlcoholDrinking` further affects `MentalHealth`. In this survey, `AlcoholDrinking` refers to heavy alcohol use defined as 14 or more drinks per week for adult men and 7 or more drinks per week for adult women.

Explore the relationship between sleep and mental and physical health

3. First, let's explore the relationship between sleep and mental health in the BRFSS dataset. Make a bar plot with `SleepTime` on the x-axis and `MentalHealth` on the y-axis. Name this plot `M_health`. Set the color palette to `ch:.25` and make the error bars `grey`.

Note: There is an additional line of code in the cell for you that will rotate the labels on the x-axis by 90°. This will keep the labels from overlapping with each other.

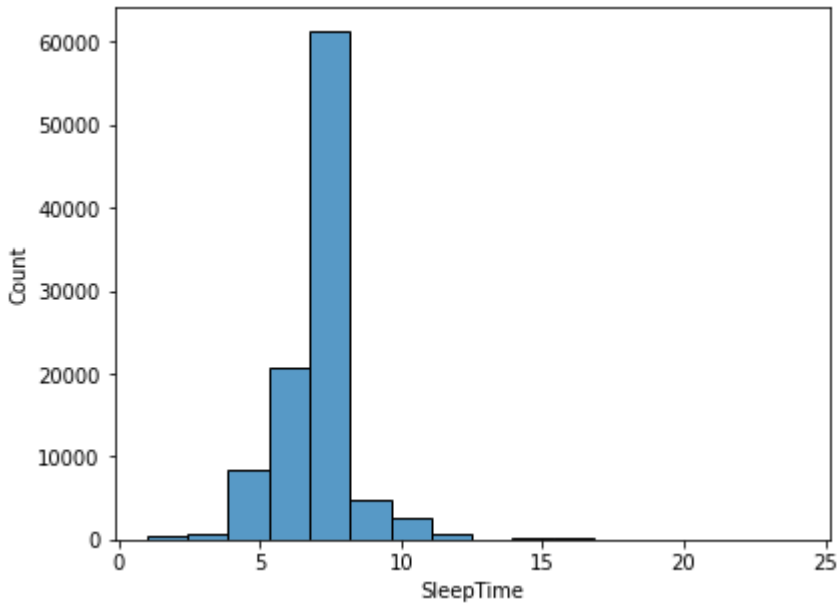
In [4]:
```
# bar plot of average mental health by sleep hours
M_health = sns.barplot(data=health, x='SleepTime', y='MentalHealth', pale
M_health.set_xticklabels(M_health.get_xticklabels(), rotation=90)
plt.show()
```



In this barplot, we can see that among people that reported sleeping 3 to 15 hours a night, those that slept 7 or 8 hours a night reported experiencing negative mental health symptoms the fewest number of days out of the month. For those that slept 16 or more hours per day, the error bars are much larger. Large error bars can be an indication that there is not enough data. Let's next plot a histogram of `SleepTime` to see how the data are distributed.

4. Since we have uneven error bars (there is much more error at the far end of the sleep data), we need to check the data distribution. Make a histogram of `SleepTime` from the `health` dataset called `sleep_hist`. Divide the data into 16 bins.

In [5]:
```
sleep_hist = sns.histplot(data=health, x='SleepTime', bins=16)
plt.show()
```

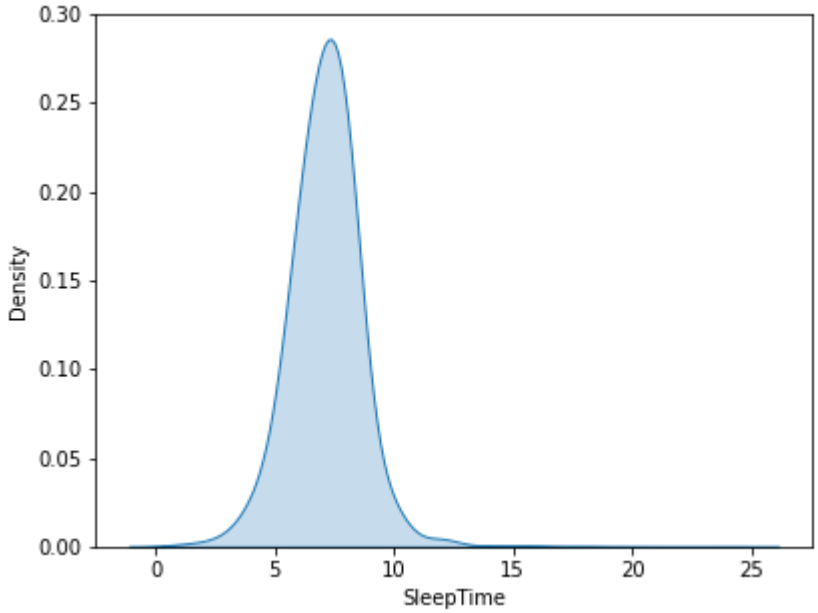


In the histogram, we can see that the vast majority of survey participants reported sleeping 5 to 10 hours a day.

5. A KDE plot could also be used to examine the data distribution. Create a KDE plot of the `health` data with `SleepTime` on the x-axis called `sleep_kde`. Set `fill` to `True` and set `bw_adjust=5`.

NOTE: By setting `bw_adjust`, we will smooth out the curve. Otherwise, we would see peaks at each whole number since participants only reported full hours of sleep.

In [6]:
```
sleep_kde = sns.kdeplot(data=health, x='SleepTime', fill=True, bw_adjust=
plt.show()
```



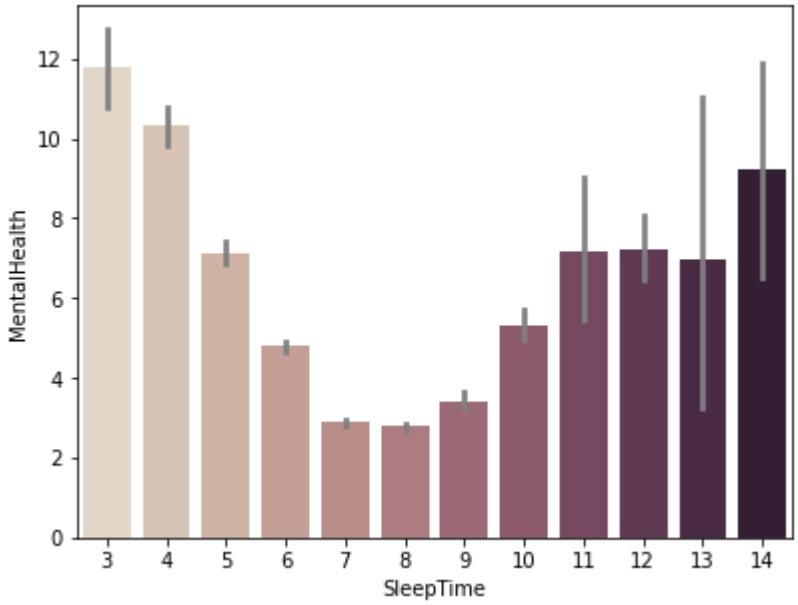
As in the histogram, we can clearly see that most participants sleep 4 to 10 hours a day. Those that sleep much more than this or much less are outliers. We can exclude these data to make our plots cleaner, easier to look at, and more accurate.

We will subset the `health` dataframe to a new dataframe called `health_sub` that only contains data from participants that sleep more than 2 or less than 15 hours a day.

6. There are only a few data points at the far ends of the sleep distribution. Run the provided code to subset the data and remove those observations.

Then, using the subset data frame `health_sub`, create a new bar plot called `M_health` with `SleepTime` on the x-axis and `MentalHealth` on the y-axis. Again, set the palette to `ch:.25` and make the error bars `grey`.

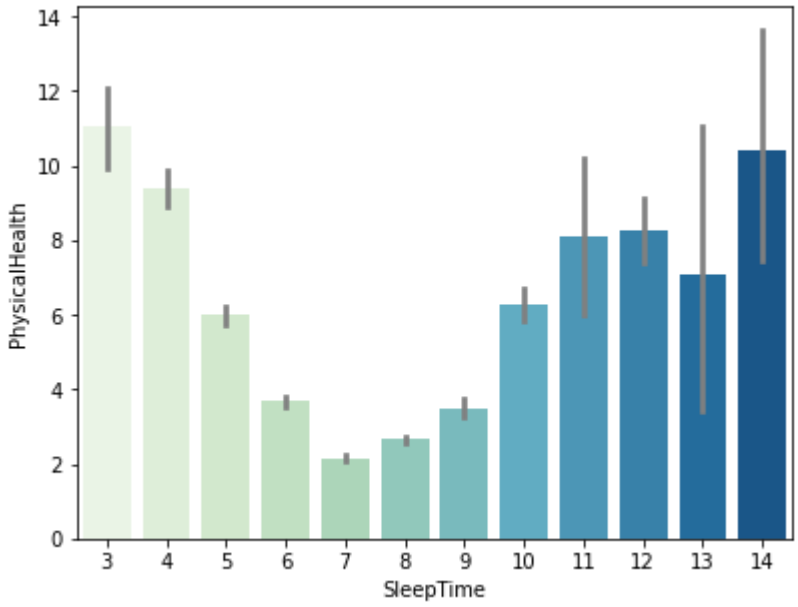
In [7]:
```
# subset to 3-14 hours of sleep
health_sub = health.loc[(health["SleepTime"] < 15) & (health["SleepTime"]

# plot the bar plot for health_sub data
M_health = sns.barplot(data=health_sub, x='SleepTime', y='MentalHealth',
plt.show()
```



By removing the outliers, the new bar plot is much cleaner and easier to interpret. Now, we can see that participants that sleep 8 hours a day generally report the least poor mental health days.

7. Next, let's check out the relationship between physical health and sleep. Using the dataframe subset `health_sub`, create a new bar plot called `P_health` with `SleepTime` on the x-axis and `PhysicalHealth` on the y-axis. For this plot, set the palette to `GnBu` and make the error bars `grey`.

In [8]:
```
P_health = sns.barplot(data=health_sub, x='SleepTime', y='PhysicalHealth'
plt.show()
```



Very interesting! Participants that sleep 7 hours a day report the least number of days with physical illness or injury, with those that sleep 8 hours a day as a close second.

8. If we want to share these results with others, we will want to put the two plots side by side, update the axis labels to make them more meaningful, and add

titles to each plot. We will create two subplots to place the plots side by side. The formatting code is already provided in the cell for you.

Below the included formatting code, create a bar plot called `M_health` from the `health_sub` dataframe with `SleepTime` on the x-axis and `MentalHealth` on the y-axis. For this plot, set:

- the palette to `ch:.25`
- the error bars to `grey`
- `ax=axes[0]`

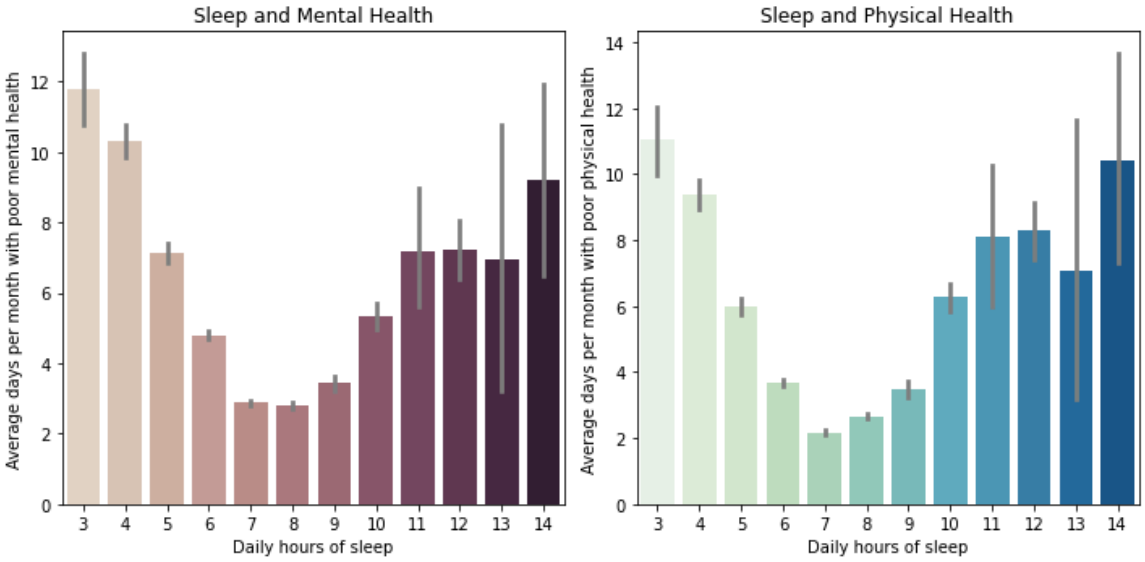
Then use `.set()` to set the title to "Sleep and Mental Health", the x label to "Daily hours of sleep", and the y label to "Average days per month with poor mental health".

Directly below the `M_health` plot, similarly create a plot `P_health` from the `health_sub` dataframe with `SleepTime` on the x-axis and `PhysicalHealth` on the y-axis. For this plot, set:

- the palette to `GnBu`
- the error bars `grey`
- `ax=axes[1]`

Then use `.set()` to set the title to "Sleep and Physical Health", the x label to "Daily hours of sleep", and the y label to "Average days per month with poor physical health".

In [9]:
```
# formatting code
plt.rcParams["figure.autolayout"] = True
plt.rcParams["figure.figsize"] = [10.00, 5.00]
f, axes = plt.subplots(1, 2)

# mental health plot
M_health = sns.barplot(data=health_sub, x='SleepTime', y='MentalHealth',
M_health.set(title='Sleep and Mental Health', xlabel='Daily hours of slee

# physical health plot
P_health = sns.barplot(data=health_sub, x='SleepTime', y='PhysicalHealth'
P_health.set(title='Sleep and Physical Health', xlabel='Daily hours of sl

plt.show()
```



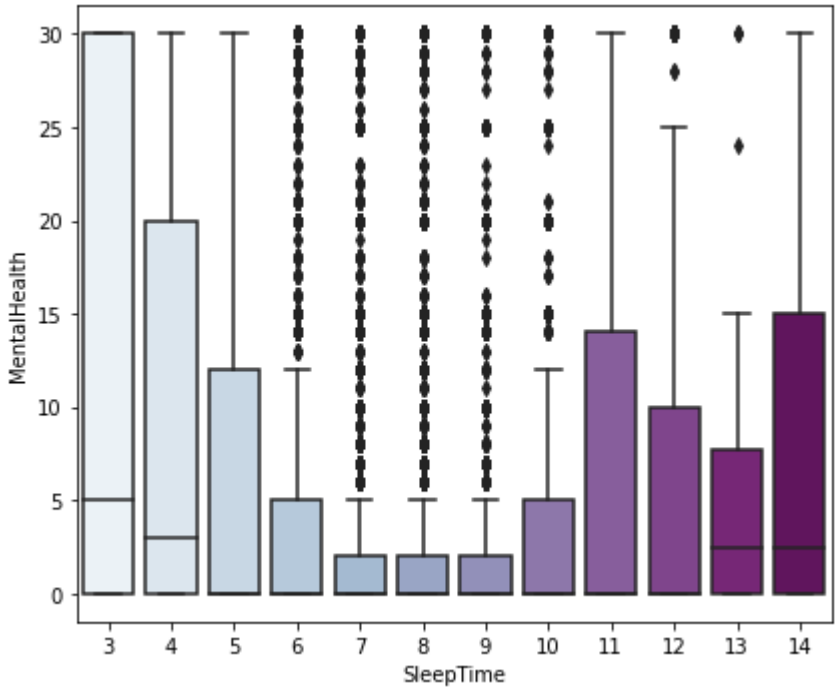
Awesome! Now that is a publishable plot!

Looking at the interaction between sleep, mental health, and alcohol consumption

9. Instead of bar plots, we could also display the mental health data with box plots, which will contain more information in a single plot.

Try creating a box plot called `M_boxplot` from the `health_sub` dataframe with `SleepTime` on the x-axis and `MentalHealth` on the y-axis. Choose any palette you would like.

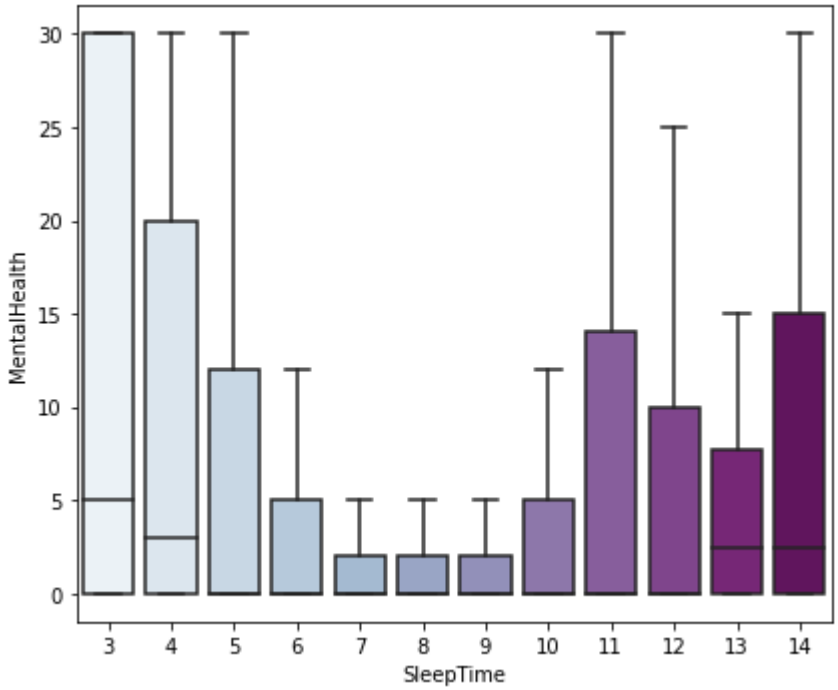
In [10]:
```
# reset the plot size for single plot
plt.rcParams["figure.figsize"] = [6.00, 5.00]

# mental health box plot
M_boxplot = sns.boxplot(data=health_sub, x='SleepTime', y='MentalHealth',
plt.show()
```



10. We can see that there is a larger range and more spread out percentiles at the high and low end of sleep where there is less data. We can also see that for people that sleep 6 to 10 hours a day, the boxes are smaller but there are a lot of outliers. These outliers make it more difficult to see the trends in the data.

Remake the boxplot, but set `showfliers=False` to remove the outliers.

In [11]:
```
M_boxplot = sns.boxplot(data=health_sub, x='SleepTime', y='MentalHealth',
plt.show()
```



11. Since alcohol consumption is linked to lower-quality and short-duration sleep, we could split our plot between participants that reported that they were heavy alcohol drinkers (7 or more drinks a week for women; 14 or more drinks a week

for men). Make a publication-ready plot of the same variables on the x and y axis but for two different subsets of the dataframe.

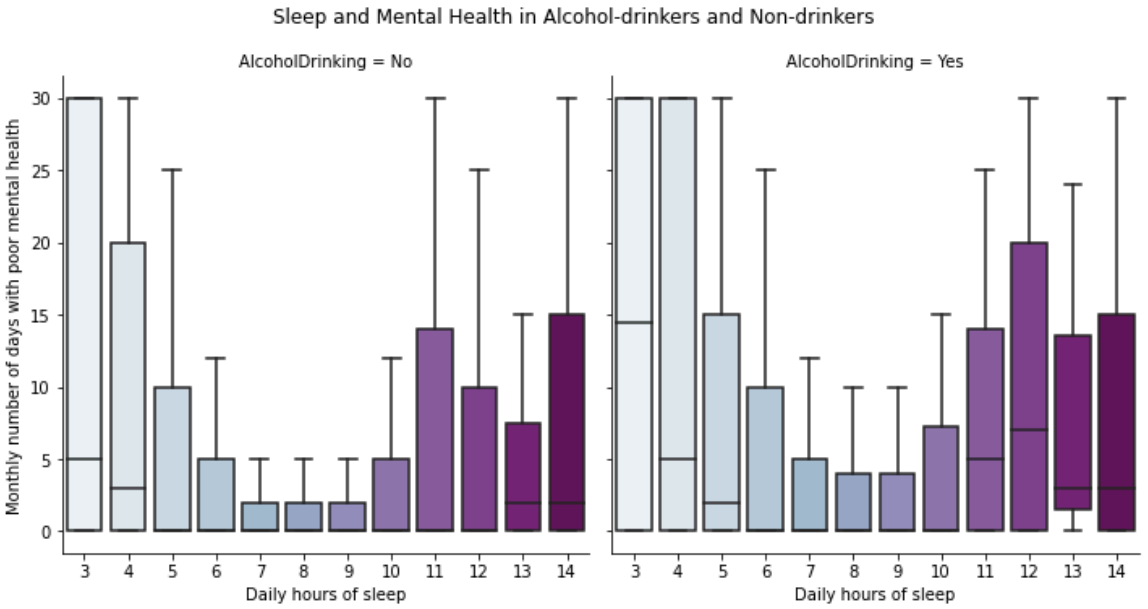
Use `sns.catplot()` to make a plot from `health_sub` called `M_A_boxplot` with the following parameters:

- `kind` set to `box`
- `col` set to `AlcoholDrinking`
- `SleepTime` on the x-axis
- `MentalHealth` on the y-axis
- `showfliers` set to `False`
- palette of your choice

Set the x label to 'Daily hours of sleep' and y label to 'Monthly number of days with poor mental health'.

Add an overall title to the plot with `M_A_boxplot.fig.suptitle('Sleep and Mental Health in Alcohol-drinkers and non-drinkers', y=1.05)`.

In [12]:
```
# create M_A_boxplot
M_A_boxplot = sns.catplot(data=health_sub, kind='box', col='AlcoholDrinki

# set the x and y labels
M_A_boxplot.set(xlabel='Daily hours of sleep', ylabel='Monthly number of

# set the overall title
M_A_boxplot.fig.suptitle('Sleep and Mental Health in Alcohol-drinkers and

plt.show()
```

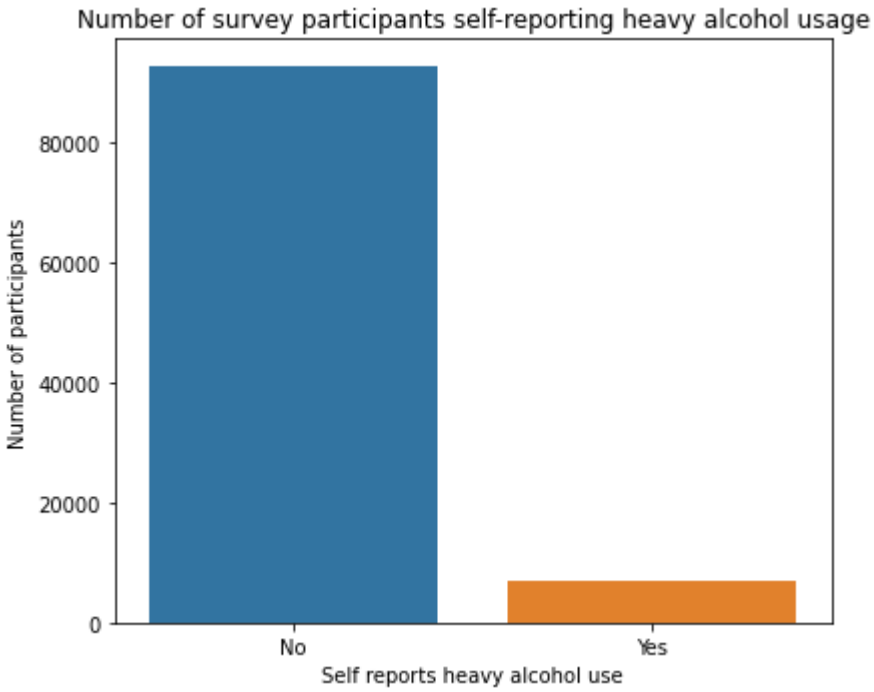


Great job! Now we have a publication-ready plot showing the distribution of poor mental health days by the number of hours slept each day for alcohol drinkers and non-drinkers. By comparing these two plots side by side, we can see an uptick in the number of poor mental health days for alcohol drinkers.

12. Before finishing, we should consider that there may be a difference in the number of participants that reported drinking more than recommended and those that reported drinking less. Let's make one last plot to check out this possibility. We can use `sns.countplot()` to make a count plot for categorical frequencies just as we would make a histogram for numerical frequencies.

Make a count plot with the `health_suc` dataframe called `A_counts` with `AlcoholDrinking` on the x-axis. Add a title and label the axes.

In [13]:
```
# create A_counts plot
A_counts = sns.countplot(data=health_sub, x='AlcoholDrinking')

# set title and axes labels
A_counts.set(title='Number of survey participants self-reporting heavy al

plt.show()
```



13. As expected, many more people did not report being heavy alcohol-drinkers than did. If we were reporting our results somewhere, we would want to include this information.

Nice work! We created multiple publication-ready plots that showed interesting relationships between variables in a large public-health dataset. However, there is much more that can be done with this dataset, so feel free to try a few more plot types on some of the other variables.

In [14]:
```
# create box plot
P_S_boxplot = sns.catplot(data=health_sub, kind='box', col='Smoking', x='

# set the x and y labels
P_S_boxplot.set(xlabel='Daily hours of sleep', ylabel='Monthly number of
```

```
# set the overall title
P_S_boxplot.fig.suptitle('Sleep and Physical Health in Smokers and Non-sm

plt.show()
```