# The Relationship Between Sleep and Select Demographic Factors in the United States

Yashkirat Singh - CS 2316

# Purpose - Why this Topic?

- Neuroscience major + college student
- Topic often comes up in my classes
- What we've been taught: sleep is closely related to physical and mental health
- Importance of sleep habits
- Expected to find
    - Correlations between sleep duration and demographic factors to match what I've learned in other classes
    - Using real data

# Data Collection Process

- Datasets
  - Downloaded Dataset Source: https://www.cdc.gov/brfss/annual_data/2022/files/LLCP2022ASC.zip
  - Web Collection #1 Source: https://www.cdc.gov/brfss/annual_data/2022/zip/codebook22_llcp-v2-508.zip
  - Web Collection #2 Source: https://data.cdc.gov/resource/hksd-2xuw.json
- How I found the datasets
  - Started with CDC website - found that they conducted this BRFSS survey
  - Needed codebook to be able to read and understand survey data
  - JSON API focused on state-level responses

# Data Cleaning Process

```python
import requests
import pandas as pd

def web_parser2():
    url = "https://data.cdc.gov/api/views/hksd-2xuw/rows.json"
    response = requests.get(url)
    raw_data = response.json()

    columns = [col["fieldName"] for col in raw_data["meta"]["view"]["columns"]]
    data = raw_data["data"]

    df = pd.DataFrame(data, columns=columns)

    topics = ["Short sleep duration among adults", "Depression among adults", "Obesity among adults", "Average mentally unhealthy days among adults"]
    df = df[df["question"].isin(topics)]

    df = df[(df["yearstart"] == "2022") & (df["stratification1"] == "Overall")]

    df = df[["yearstart", "locationabbr", "locationdesc", "topic", "question", "datavalue"]]

    df = df.dropna(subset=["datavalue"])
    df["datavalue"] = pd.to_numeric(df["datavalue"])

    df.to_csv("cleaned_json_data.csv", index=False)

    return df
```

```python
import pandas as pd
import numpy as np

def data_parser():
    colspecs = [(113, 115), (1900, 1901), (1997, 2001), (185, 187)
    names = ["SLEPTIM1", "_MENT14D", "_BMI5", "INCOME3", "EDUCA"]

    df = pd.read_fwf("LLCP2022.txt", colspecs=colspecs, names=name

    invalid_codes = {
        "SLEPTIM1": [77, 99],
        "_MENT14D": [9],
        "INCOME3": [77, 99],
        "EDUCA": [9]
    }

    for col, codes in invalid_codes.items():
        df[col] = df[col].replace(codes, np.nan)

    df = df.astype(float)

    df = df[
        ((df["SLEPTIM1"].between(1, 24)) | df["SLEPTIM1"].isna()) &
        ((df["_MENT14D"].isin([1, 2, 3])) | df["_MENT14D"].isna()) &
        ((df["INCOME3"].between(1, 11)) | df["INCOME3"].isna()) &
        ((df["EDUCA"].between(1, 6)) | df["EDUCA"].isna()) &
        ((df["_BMI5"] >= 1) | df["_BMI5"].isna())
    ]
```

```python
import pandas as pd
from bs4 import BeautifulSoup

def web_parser1():
    with open('USCODE22_LLCP_102523 4.html', 'r', encoding='windows-1252') as f:
        html = f.read()

    soup = BeautifulSoup(html, 'html.parser')

    variable_blocks = soup.find_all('td', class_='l m linecontent')

    all_variables = []

    for variable in variable_blocks:
        text = variable.get_text(separator="\n")

        variables_dict = {}
        for line in text.split('\n'):
            line = line.strip()
            if ':' in line:
                key, value = line.split(':', 1)
                key_final = ' '.join(key.split())
                variables_dict[key_final] = value.strip()

        variable_name = variables_dict.get('SAS Variable Name', 'N/A')
        variable_label = variables_dict.get('Label', 'N/A')

        table = variable.find_parent('table')
        rows = table.find_all('tr')[2:]

        values = []
        for row in rows:
            cells = row.find_all('td')
            if len(cells) >= 2:
                code = cells[0].get_text(strip=True)
                value_label = cells[1].get_text(strip=True)
                if code.lower() != 'blank' and code != '':
                    values.append((code, value_label))

        all_variables.append({
            'Variable': variable_name,
            'Label': variable_label,
            'Values': values
        })

    df_rows = []
    for var in all_variables:
        if var['Values']:
            for val_code, val_label in var['Values']:
                df_rows.append({
                    'Variable': var['Variable'],
                    'Label': var['Label'],
                    'Value Code': val_code,
                    'Value Label': val_label
                })
        else:
            df_rows.append({
                'Variable': var['Variable'],
                'Label': var['Label'],
                'Value Code': None,
                'Value Label': None
            })

    df = pd.DataFrame(df_rows)

    variables_of_interest = ["SLEPTIM1", "_MENT14D", "_BMI5", "INCOME3", "EDUCA"]

    df = df[df["Variable"].isin(variables_of_interest)]

    df.to_csv('parsed_codebook_data.csv', index=False)

    return df
```

- Used pandas to read and sort survey data based on specific variables
- Used BeautifulSoup to parse the codebook
- Used requests module to filter JSON data for specific questions and variable data
- Inconsistencies
    - Survey data numeric codes for invalid responses changed to NaN
    - converted the datavalue column of JSON to numeric types to prevent potential errors when cleaning
    - Divided BMI values by a factor to be in the standard BMI scale

# Data Analysis, Insights and Visuals - Insights Code

```python
import pandas as pd

def insight1():
    sleep_hours_bins = [0, 4, 6, 8, 24]
    sleep_hours_labels = ["0–4 hours", "5–6 hours", "7–8 hours", "8+ hours"]

    df = pd.read_csv("cleaned_brfss_data.csv")
    df["Sleep_Duration"] = pd.cut(df["SLEPTIM1"], bins=sleep_hours_bins, labels=sleep_hours_labels, right=True)

    output = df.groupby("Sleep_Duration", observed=True)["_MENT14D"].mean().reset_index()
    output = output.rename(columns={"_MENT14D": "Mentally Unhealthy Days Over the Past Two Weeks", "Sleep_Duration": "Sleep Duration Per Night"})
    output = output.style.hide(axis="index").format({"Mentally Unhealthy Days Over the Past Two Weeks": "{:.3f}"})

    return output
```

```python
import pandas as pd

def insight4():
    df = pd.read_csv("cleaned_brfss_data.csv")

    df = df.dropna(subset=["_BMI5", "SLEPTIM1"])

    bmi_bins = [0, 18.5, 24.9, 29.9, 100]
    bmi_labels = ["Underweight", "Normal weight", "Overweight", "Obese"]

    df["Bmi_Category"] = pd.cut(df["_BMI5"], bins=bmi_bins, labels=bmi_labels)

    output = df.groupby("Bmi_Category", observed=True)["SLEPTIM1"].mean().reset_index()
    output = output.rename(columns={"SLEPTIM1": "Average Nightly Sleep Duration in Hours", "Bmi_Category": "BMI Category"})
    output = output.style.hide(axis="index").format({"Average Nightly Sleep Duration in Hours": "{:.3f}"})

    return output
```

```python
import pandas as pd

def insight3():
    df = pd.read_csv("cleaned_brfss_data.csv")

    df = df.dropna(subset=["INCOME3", "SLEPTIM1"])

    dict_of_income_amount_labels = {
        1: "$0 – $10,000",
        2: "$10,000 – $14,999",
        3: "$15,000 – $19,999",
        4: "$20,000 – $24,999",
        5: "$25,000 – $34,999",
        6: "$35,000 – $49,999",
        7: "$50,000 – $74,999",
        8: "$75,000 or more",
    }

    df["Income_Level"] = df["INCOME3"].map(dict_of_income_amount_labels)

    output = df.groupby("Income_Level")["SLEPTIM1"].mean().reset_index()
    output = output.rename(columns={"SLEPTIM1": "Average Nightly Sleep Duration in Hours", "Income_Level": "Income Level"})
    output = output.style.hide(axis="index").format({"Average Nightly Sleep Duration in Hours": "{:.3f}"})

    return output
```

```python
import pandas as pd

def insight2():
    df = pd.read_csv("cleaned_brfss_data.csv")

    df = df.dropna(subset=["EDUCA"])

    dict_of_educa_responses = {
        1: "Never attended school or only kindergarten",
        2: "Grades 1 through 8 (Elementary)",
        3: "Grades 9 through 11 (Some high school)",
        4: "Grade 12 or GED (High school graduate)",
        5: "College 1 year to 3 years (Some college or technical school)",
        6: "College 4 years or more (College graduate)"
    }

    df["Education_Level"] = df["EDUCA"].astype(int).map(dict_of_educa_responses)

    median_amount_of_sleep = df.groupby("Education_Level")["SLEPTIM1"].median().round(3)

    df["short_sleep"] = df["SLEPTIM1"] < 6

    proportion_of_short_sleep = df.groupby("Education_Level")["short_sleep"].mean().round(3)

    output = pd.DataFrame({
        "Median Sleep Duration in Hours per Night": median_amount_of_sleep,
        "Proportion of Short Sleep (<6 Hours)": proportion_of_short_sleep
    })

    output = output.reset_index()
    output = output.rename(columns={"Education_Level": "Highest Level of Education"})
    output = output.style.hide(axis="index").format({"Median Sleep Duration in Hours per Night": "{:.3f}", "Proportion of Short Sleep (<6 Hours)": "{:.3f}"})
    return output
```

```python
import pandas as pd

def insight5():
    df = pd.read_csv("cleaned_json_data.csv")

    question = [
        "Short sleep duration among adults",
    ]
    df = df[df["question"].isin(question)]

    states = [
        "Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
        "Connecticut", "Delaware", "District of Columbia", "Florida", "Georgia",
        "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas", "Kentucky",
        "Louisiana", "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota",
        "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada", "New Hampshire",
        "New Jersey", "New Mexico", "New York", "North Carolina", "North Dakota",
        "Ohio", "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina",
        "South Dakota", "Tennessee", "Texas", "Utah", "Vermont", "Virginia",
        "Washington", "West Virginia", "Wisconsin", "Wyoming"
    ]
    df = df[df["locationdesc"].isin(states)]

    output_table = df.pivot_table(
        index="locationdesc",
        columns="question",
        values="datavalue"
    )

    normalize = [
        "Short sleep duration among adults",
    ]
    for column in normalize:
        minimum = output_table[column].min()
        maximum = output_table[column].max()
        normalized_column_name = f"{column} (normalized)"
        output_table[normalized_column_name] = (output_table[column] - minimum) / (maximum - minimum)

    output_table = output_table.reset_index()
    output_table = output_table.sort_values("Short sleep duration among adults (normalized)", ascending=False)
    output_table = output_table.rename(columns={"locationdesc": "State", "Short sleep duration among adults": "Percentage of Short Sleep", "Short sleep duration among adults (normalized)": "Percentage of Short Sleep (normalized value)"})
    output_table = output_table.style.hide(axis="index").format({"Percentage of Short Sleep": "{:.3f}", "Percentage of Short Sleep (normalized value)": "{:.3f}"})

    return output_table
```

# Data Analysis, Insights and Visuals - Insights Explained

- People who sleep 7–8 hours report the fewest mentally unhealthy days
- Short sleep is most common among those with some high school education and least common among college graduates
- Sleep duration is fairly consistent across income levels but is slightly shorter among lower-income individuals
- Individuals with higher BMI tend to sleep slightly less than those with normal or underweight BMI
- States like Hawaii and West Virginia show the highest rates of short sleep

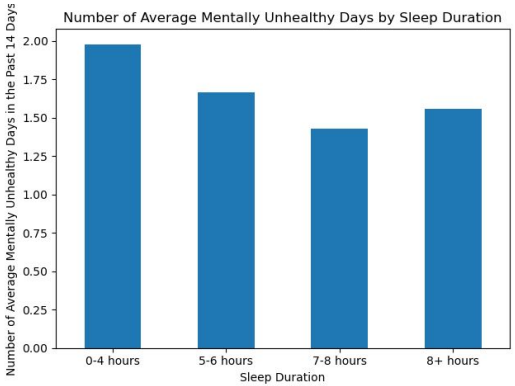
# Data Analysis, Insights and Visuals - Visualizations Code

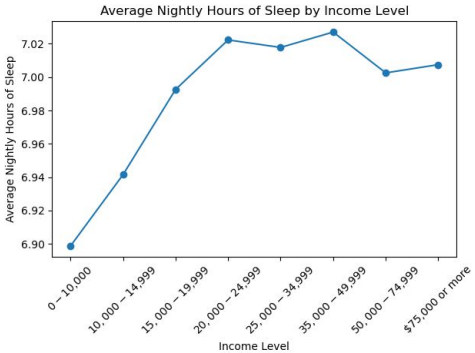
```python
import pandas as pd
import matplotlib.pyplot as plt

def visual1():
    sleep_hours_bins = [0, 4, 6, 8, 24]
    sleep_hours_labels = ["0-4 hours", "5-6 hours", "7-8 hours", "8+ hours"]

    df = pd.read_csv("cleaned_brfss_data.csv")
    df["Sleep_Duration"] = pd.cut(df["SLEPTIM1"], bins=sleep_hours_bins, labels=sleep_hours_labels, right=True)

    output = df.groupby("Sleep_Duration", observed=True)["_MENT14D"].mean()

    output.plot(kind="bar")
    plt.xlabel("Sleep Duration")
    plt.ylabel("Number of Average Mentally Unhealthy Days in the Past 14 Days")
    plt.title("Number of Average Mentally Unhealthy Days by Sleep Duration")
    plt.xticks(rotation=0)
    plt.tight_layout()
    plt.show()
```

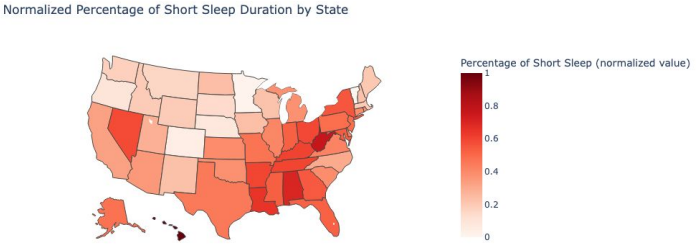
```python
import pandas as pd
import matplotlib.pyplot as plt

def visual2():
    df = pd.read_csv("cleaned_brfss_data.csv")
    df = df.dropna(subset=["INCOME3", "SLEPTIM1"])

    dict_of_income_amount_labels = {
        1: "$0 - $10,000",
        2: "$10,000 - $14,999",
        3: "$15,000 - $19,999",
        4: "$20,000 - $24,999",
        5: "$25,000 - $34,999",
        6: "$35,000 - $49,999",
        7: "$50,000 - $74,999",
        8: "$75,000 or more",
    }

    df["Income_Level"] = df["INCOME3"].map(dict_of_income_amount_labels)
    df["income_question_code"] = df["INCOME3"]

    output = df.groupby(["income_question_code", "Income_Level"])["SLEPTIM1"].mean()
    grouped = output.sort_index().reset_index()


    plt.plot(grouped["Income_Level"], grouped["SLEPTIM1"], marker="o", linestyle="-")
    plt.title("Average Nightly Hours of Sleep by Income Level")
    plt.xlabel("Income Level")
    plt.ylabel("Average Nightly Hours of Sleep")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```

```python
import pandas as pd
import plotly.express as px

def visual3():
    df = pd.read_csv("cleaned_json_data.csv")

    question = {
        "Short sleep duration among adults",
    }
    df = df[df["question"].isin(question)]

    states = [
        "Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
        "Connecticut", "Delaware", "District of Columbia", "Florida", "Georgia",
        "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas", "Kentucky",
        "Louisiana", "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota",
        "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada", "New Hampshire",
        "New Jersey", "New Mexico", "New York", "North Carolina", "North Dakota",
        "Ohio", "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina",
        "South Dakota", "Tennessee", "Texas", "Utah", "Vermont", "Virginia",
        "Washington", "West Virginia", "Wisconsin", "Wyoming"
    ]
    df = df[df["locationdesc"].isin(states)]

    output_table = df.pivot_table(
        index="locationdesc",
        columns="question",
        values="datavalue"
    )

    normalize = {
        "Short sleep duration among adults",
    }
    for column in normalize:
        minimum = output_table[column].min()
        maximum = output_table[column].max()
        normalized_column_name = f"{column} (normalized)"
        output_table[normalized_column_name] = (output_table[column] - minimum) / (maximum - minimum)

    output_table = output_table.reset_index()
    output_table = output_table.rename(columns={"locationdesc": "State", "Short sleep duration among adults": "Percentage of Short Sleep", "Short sleep duration among adults (normalized)": "Percentage of Short Sleep (normalized value)"})


    state_abbrev = {
      "Alabama": "AL", "Alaska": "AK", "Arizona": "AZ", "Arkansas": "AR", "California": "CA",
      "Colorado": "CO", "Connecticut": "CT", "Delaware": "DE", "District of Columbia": "DC",
      "Florida": "FL", "Georgia": "GA", "Hawaii": "HI", "Idaho": "ID", "Illinois": "IL",
      "Indiana": "IN", "Iowa": "IA", "Kansas": "KS", "Kentucky": "KY", "Louisiana": "LA",
      "Maine": "ME", "Maryland": "MD", "Massachusetts": "MA", "Michigan": "MI",
      "Minnesota": "MN", "Mississippi": "MS", "Missouri": "MO", "Montana": "MT",
      "Nebraska": "NE", "Nevada": "NV", "New Hampshire": "NH", "New Jersey": "NJ",
      "New Mexico": "NM", "New York": "NY", "North Carolina": "NC", "North Dakota": "ND",
      "Ohio": "OH", "Oklahoma": "OK", "Oregon": "OR", "Pennsylvania": "PA",
      "Rhode Island": "RI", "South Carolina": "SC", "South Dakota": "SD", "Tennessee": "TN",
      "Texas": "TX", "Utah": "UT", "Vermont": "VT", "Virginia": "VA", "Washington": "WA",
      "West Virginia": "WV", "Wisconsin": "WI", "Wyoming": "WY"
    }

    output_table["state_abbrev"] = output_table["State"].map(state_abbrev)

    fig = px.choropleth(
        output_table,
        locations="state_abbrev",
        locationmode="USA-states",
        color="Percentage of Short Sleep (normalized value)",
        scope="usa",
        color_continuous_scale="Reds",
        labels={"Normalized Percentage": "Normalized % Short Sleep"},
        title="Normalized Percentage of Short Sleep Duration by State"
    )

    fig.show()
```

# Data Analysis, Insights and Visuals - Visualizations



# Overall Results and Conclusion - Code Highlights

- Parsing fixed-width files using pandas.read_fwf
- BeautifulSoup
- Requests
- NaNs
- Normalization
- Plotly + Matplotlib
- Pandas cut function
- Pandas pivot_table function

# Overall Results and Conclusion

- Sleep duration is related to demographic factors
- Can target specific demographics and populations to increase awareness
- Further research and expansion is needed to explore year-over-year data