Happy US

Nicholas Grove, William Jones, Fangyi Xu

Data Sources

We chose these data sources because we were interested in the relationship between state happiness and their various health measures.

1. annual_report.csv

   United Health Foundation: America's Health Rankings
   https://www.americashealthrankings.org/explore/annual

2. happiness.csv

   Wallet Hub: Happiest states in America https://wallethub.com/edu/happiest-states/6959/

Data Transformation I

We first reduced the health measurement dataframe from the annual_report .csv by selecting specific columns.

TRANSFORM: Only keep columns of interest

In [3]:
```
# Only keep 6 columns: state name, rank, measure name, score, source, source year
health_df = health_df[['State Name', 'Rank', 'Measure Name', 'Score', 'Source', 'Source Year']].copy()
health_df.head()
```

Out[3]:

	State Name	Rank	Measure Name	Score	Source	Source Year
0	Alaska	34.0	Adverse Childhood Experiences	0.96	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
1	Alabama	46.0	Adverse Childhood Experiences	1.56	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
2	United States	NaN	Adverse Childhood Experiences	NaN	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
3	Arkansas	47.0	Adverse Childhood Experiences	1.76	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
4	Arizona	48.0	Adverse Childhood Experiences	1.82	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017

Data Transformation II

Of the original 628 health measures in the health measurement dataframe, we filtered through the dataframe to select only three measurements of interest (i.e. mental illness, insufficient sleep, and air pollution).

TRANSFORM: Isolate each Health Measure Name of interest

- mental illness
- insufficient sleep
- air_pollution

In [4]:
```
# Create mental illness dataframe
health_mental_df = health_df.loc[health_df['Measure Name']=='Mental illness', :]
health_mental_df.head()
```

Out[4]:

	State Name	Rank	Measure Name	Score	Source	Source Year
23598	Alaska	45.0	Mental illness	1.94	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
23599	Alabama	14.0	Mental illness	0.05	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
23600	United States	NaN	Mental illness	NaN	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
23601	Arkansas	41.0	Mental illness	1.59	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017
23602	Arizona	28.0	Mental illness	0.45	U.S. HHS, HRSA, Maternal and Child Health Bure...	2016-2017

Data Transformation III

Finally, we defined a function to sort each dataframe by state and rename columns to match the column names in the SQL table.

TRANSFORM: Clean each dataframe.

Define a function to perform the following:

- Sort each dataframe by state
- Rename columns to match the SQL table column names
- Reset index for each dataframe.

In [7]:
```
# Function to clean dataframe
def clean_df(df, col_append):
    df = df.sort_values(by='State Name', ascending=False)
    df = df.rename(columns={"State Name":"state", 'Measure Name': 'measure_name', 'Source Year': 'source_year', "Rank":"rank", '
    df = df.reset_index(drop=True)
    df.columns = ['{}_'.format(col_append)+col_name for col_name in df.columns]
    return df
```

Database Loading

Next, we loaded the dataframes to the SQL database. Data was loaded to the happy_US database in pgAdmin 4. Dataframes are loaded to tables mental_illness, insufficient_sleep, air_pollution, and happiness_score respectively.

LOAD: Load dataframes to SQL database

In [14]:
```
rds_connection_string = "postgres:postgres@localhost:5432/happy_US"
engine = create_engine(f'postgresql://{rds_connection_string}')
```

In [15]:
```
engine.table_names()
```

Out[15]:
```
['mental_illness', 'insufficient_sleep', 'air_pollution', 'happiness_score']
```

In [16]:
```
health_mental_df.to_sql(name='mental_illness', con=engine, if_exists='append', index=False)
health_sleep_df.to_sql(name='insufficient_sleep', con=engine, if_exists='append', index=False)
health_airpollution_df.to_sql(name='air_pollution', con=engine, if_exists='append', index=False)
happiness_df.to_sql(name='happiness_score', con=engine, if_exists='append', index=False)
```

Final Result

- We joined the tables for state happiness, insufficient sleep, air pollution, and mental illness rankings to see the relationship between overall state happiness and the selected health indicators. There were some obvious outcomes, such as a visibly strong inverse relationship between state happiness and mental illness. However, insufficient sleep seems to have a somewhat weak connection with the happiest state also being to most sleep deprived state.