```
import time
import pandas as pd
import numpy as np

CITY_DATA = {'chicago': 'chicago.csv',
        'new york city': 'new_york_city.csv',
        'washington': 'washington.csv'}


def get_filters():
        """
        Asks user to specify a city, month, and day to analyze.

        Returns:
        (str) city - name of the city to analyze
        (str) month - name of the month to filter by, or "all" to apply no month filter
        (str) day - name of the day of week to filter by, or "all" to apply no day filter
        """
        print('Hello! Let\'s explore some US bikeshare data!')
        # TO DO: get user input for the city (chicago, new york city, washington). HINT: Use a
while loop to handle invalid inputs
        print
        city = ""
        while city not in ('chicago', 'new york city', 'washington'):
           city = input("Which City would you like to filter by?").lower()
           if city not in ('chicago', 'new york city', 'washington'):
        print("Sorry, I didn't get that. Please try again.")

        # TO DO: get user input for month (all, january, february, ... , june)
        print
        month = ""
        while month not in ('january', 'february', 'march', 'april', 'may', 'june', 'all'):
           month = input("Are you looking for a specific month? If so, please enter the month as
follows: january, february, march, april, may, june, july, august, september, october, november,
december or type 'all' if you do not have any preference.") .lower()
        if month not in ('january', 'february', 'march', 'april', 'may', 'june', 'all'):
           print("Sorry, I didn't get that. Please try again.")

        # TO DO: get user input for day of week (all, monday, tuesday, ... sunday)
        print
        day = ""
        while day not in ('sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'all'):
```

```
            day = input("Are you looking for a specific day? If so, please enter the day as follows: sunday, monday, tuesday, wednesday, thursday, friday, saturday or type 'all' if you do not have any preference.").lower()
        if day not in ('sunday', 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'all'):
            print("Sorry, I didn't get that. Please try again.")

        print('-' * 40)

        print('-' * 40)
        return city, month, day


def load_data(city, month, day):
        df = pd.read_csv(CITY_DATA[city])
        print(df)

        # convert the Start Time column to datetime
        df['Start Time'] = pd.to_datetime(df['Start Time'])
        print(df)['Start Time']
        # extract month and day of week from Start Time to create new columns
        df['month'] = df['Start Time'].dt.month
        df['day_of_week'] = df['Start Time'].dt.weekday_name

        # filter by month if applicable
        if month != 'all':
        # use the index of the months list to get the corresponding int months = ['january', 'february', 'march', 'april', 'may', 'june', 'all']
        month = months.index(month) + 1

        # filter by month to create the new dataframe
        df = df[df['month'] == month]

        # filter by day of week if applicable
        if day != 'all':
        # filter by day of week to create the new dataframe
            df = df[df['day_of_week'] == day.title()]

        return df


def time_stats(df):
        """Displays statistics on the most frequent times of travel."""
```

```
    print('\nCalculating The Most Frequent Times of Travel...\n')
    start_time = time.time()

    # display the most common month

    # display the most common day of week

    # display the most common start hour

    print("\nThis took %s seconds." % (time.time() - start_time))
    print('-' * 40)


def station_stats(df):
    """Displays statistics on the most popular stations and trip."""

    print('\nCalculating The Most Popular Stations and Trip...\n')
    start_time = time.time()

    # display most commonly used start station

    # display most commonly used end station

    # display most frequent combination of start station and end station trip

    print("\nThis took %s seconds." % (time.time() - start_time))
    print('-' * 40)


def trip_duration_stats(df):
    """Displays statistics on the total and average trip duration."""

    print('\nCalculating Trip Duration...\n')
    start_time = time.time()

    # display total travel time

    # display mean travel time

    print("\nThis took %s seconds." % (time.time() - start_time))
    print('-' * 40)


def user_stats(df):
```

```
    """Displays statistics on bikeshare users."""

    print('\nCalculating User Stats...\n')
    start_time = time.time()

    # Display counts of user types

    # Display counts of gender

    # Display earliest, most recent, and most common year of birth

    print("\nThis took %s seconds." % (time.time() - start_time))
    print('-' * 40)


def main():

    while True:
        city, month, day = get_filters()
        df = load_data(city, month, day)
        restart = input('\nWould you like to restart? Enter yes or no.\n')
        if restart.lower() != 'yes':
            break

        """
    time_stats(df)
    station_stats(df)
    trip_duration_stats(df)
    user_stats(df)
  """



if __name__ == "__main__":
    main()
```