Does Ohio have public transportation?

There are no **public transportation** services in 27 of **Ohio's** 88 counties; this represents about 9 percent of the state's total population. Without **public transportation**, only people participating in specific programs for the highest **need** individuals are able to reach essential services without a personal vehicle

What different types of transport are there?

Transport modes are the means of supporting the mobility of passengers and freight. They are mobile transport assets and fall into three basic types; land (road, rail, pipelines), **water** (shipping), and air.

The wait time (after a scheduled arrival time) in minutes for a train to arrive is Uniformly distributed over the interval [0,12]. You observe the wait time for the next 100 trains to arrive. Assume wait times are independent.

Part a) What is the approximate probability (to 2 decimal places) that the sum of the 100 wait times you observed is between 565 and 669 ?

Part b) What is the approximate probability (to 2 decimal places) that the average of the 100 wait times exceeds 6 minutes?

Part c) Find the probability (to 2 decimal places) that 97 or more of the 100 wait times exceed 1 minute. Please carry answers to at least 6 decimal places in intermediate steps.

Part d) Use the Normal approximation to the Binomial distribution (with continuity correction) to find the probability (to 2 decimal places) that 56 or more of the 100 wait times recorded exceed 5 minutes.

a)

expected sum =100\*(0+12)/2= 600

std deviation =(12-0)\*sqrt(100/12)=34.64

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| probability = | P(565<X<669) | = | P(-1.01<Z<1.99)= | 0.9768-0.1562= | **0.82** |

b) approximate probability  that the average of the 100 wait times exceeds 6 minutes =**0.5**

c)

P(X>=97)=P(Z>1.93)= **0.03**

(try 0.04 if this comes wrong)

d)

P(X >55.5)=P(Z>0.57)=**0.72**

Code:

Python Homework

Arrival Time Estimator  
Create a program that calculates the estimated hours and minutes for a trip. This should include an estimated date/time of departure and an estimated date/time of arrival.

Console  
Arrival Time Estimator Estimated date of departure (YYYY-MM-DD): 2016-11-23 Estimated time of departure (HH:MM AM/PM): 10:30 AM Enter miles: 200 Enter miles per hour: 65 Estimated travel time Hours: 3 Minutes: 5 Estimated date of arrival: 2016-11-23 Estimated time of arrival: 01:35 PM Continue? (y/n): y Estimated date of departure (YYYY-MM-DD): 2016-11-29 Estimated time of departure (HH:MM AM/PM): 11:15 PM Enter miles: 500 Enter miles per hour: 80 Estimated travel time Hours: 6 Minutes: 20 Estimated date of arrival: 2016-11-30 Estimated time of arrival: 05:35 AM Continue? (y/n): n

Specifications  
• For the date/time of departure and arrival, the program should use the YYYY-MM-DD format for dates and the HH:MM AM/PM format for times.  
• For the miles and miles per hour, the program should only accept integer entries like 200 and 65. • Assume that the user will enter valid data

Import the required packages.

import time

import datetime

from datetime import datetime

from datetime import timedelta

ch='y'

#Run the loop to get the

#input and perform

#the calculations.

while ch!='n':

#Define the try block

#to get the input and

#perform the calculations.

#Move to the except block if there

#is any error.

try:

    #Prompt the user to enter the input.

    est\_date\_dept = raw\_input('Estimated date of departure (YYYY-MM-DD):')

    est\_time\_dept = raw\_input('Estimated time of departure (HH:MM AM/PM):')

    dist = int(raw\_input('Enter Miles:'))

    speed = int(raw\_input('Enter miles per hour:'))

    #Calculate the hours to travel

    #the given distance at

    #the given speed.

    travel\_time = float(dist)/speed

    #Convert the calculated hours

    #into hours, minutes, and seconds.

    t=timedelta(hours=travel\_time)

    #Convert the obtained

    #time into the string format.

    t=str(t)

    #Split the string at ':'

    #to obtain the

    #hours, minutes, and seconds

    #at separate indices.

    t=t.split(':')

    #Check if the seconds

    #are greater than zero or not.

    if t[2] > '00':

      #If the seconds are greater than

      #zero then, increase the

      #number of minutes by 1.

      t[1]=int(t[1])+1

      #Increase the value of the

      #travel\_time by 0.01.

      travel\_time+=0.01

    #Convert the time entered into the recognizable format.

    time = datetime.strptime(est\_time\_dept, "%I:%M %p")

    #Convert the entered time into 24-hour

    #Format to perform the calculations.

    est\_time\_dept = datetime.strftime(time, "%H:%M")

    #Create a complete time by appending

    #the time at the end of the date.

    est\_date\_dept = est\_date\_dept+" "+est\_time\_dept

    #Calculate the time and date of arrival.

    arr\_date\_time = datetime.strptime(est\_date\_dept, "%Y-%m-%d %H:%M")

    arr\_date\_time += timedelta(hours=travel\_time)

    est\_arr = arr\_date\_time.strftime("%Y-%m-%d %I:%M %p")

    #Change the format of the

    #time obtained to the string format.

    est\_arr=str(est\_arr)

    #Split the obtained time at spaces

    #to obtain a list containing the

    #date,time and (AM/PM) seperately.

    est\_arr=est\_arr.split()

    #Print the time to travel

    #the given distance at

    #the given speed.

    print '\n\nEstimated travel time'

    print 'Hours:',t[0],'\nMinutes:', t[1]

    #Print the date of the arrival.

    print 'Estimated date of arrival:',est\_arr[0]

    #Print the time of the arrival.

    print 'Estimated time of arrival:', est\_arr[1], est\_arr[2]

    #Ask the user whether he/she

    #wishes to continue or not.

    ch = raw\_input('\nContinue? (y/n):')

#If there is any error in the input

#then the except block will execute.

except:

    #Print the error message.

    print "Invalid input"

    #Ask the user whether he/she

    #wishes to continue or not.

    ch = raw\_input("\nContinue? (y/n):")

