Mujtaba Pal
Case Studies in Math Modeling
28th October 2022

Solar Panels Project Report

Abstract:

Our project looked at the practicality of using solar panels to help fulfill Easton CAP's goal of achieving 100% renewable energy by 2030 and subsequently aiming for at least a 30% reduction in 2016-level emissions by 2030. We got production and consumption data from a single house in Allentown and used that to generate daily solar energy production and consumption in all houses in Easton by using a modifier. We found that having solar panels greatly reduced the amount of outside energy needed. We also looked at how using a solar battery improved the efficiency of our project by analyzing the cost and efficiency implications of installing solar panels with different solar battery sizes. We concluded that having a 15-20 kWh solar battery installed alongside the solar panels was the most eco-friendly solution but it took 25 years to pay off the initial installation costs for that while having no battery was the most cost effective solution and took 15 years to pay off. We also found that regardless of whichever solution we used, we always needed some alternative energy source because of the inability of batteries to store solar energy for long periods of time.

Introduction:

The city of Easton developed a CAP in 2016 that aimed for at least a 30% reduction in 2016-level emissions by 2030 and an 80% reduction by 2050, with a more ambitious goal of reducing emissions by 60% by 2030. One of the main targets to accomplish this goal was for 100% of Easton's electricity to come from renewable energy sources by 2030. Our project focused on how this energy could be generated solely from using solar panels while keeping in mind the cost and practicality limitations that solar panels might have. Our purpose was to find a solution that was both financially feasible and environmentally friendly to help the city of Easton achieve its goals efficiently and effectively.

Data:

We used production data that we found on the Solar Edge website from a house in Allentown, shown in Fig 1 [1]. We used daily data for every month of 2021 to come up with a mean and standard deviation for each month. This mean and standard deviation was then used to come up with a normal distribution for each specific month.

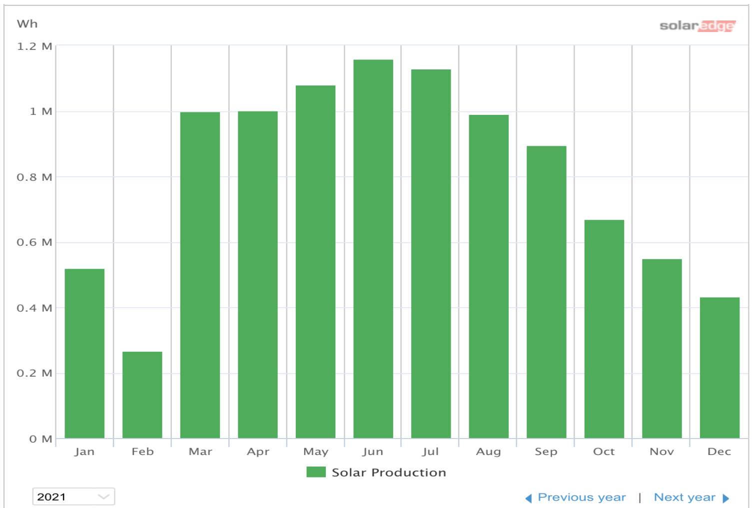


Fig 1. Production data for 2021

For our consumption data, we also used another house in Allentown that we found on the [Solar Edge](#) website [2]. We similarly used daily data for every month of 2018 to come up with a mean and standard deviation for each month which was then used to generate a normal distribution for each month. The graph for consumption in July 2018 is shown below in Fig 2.

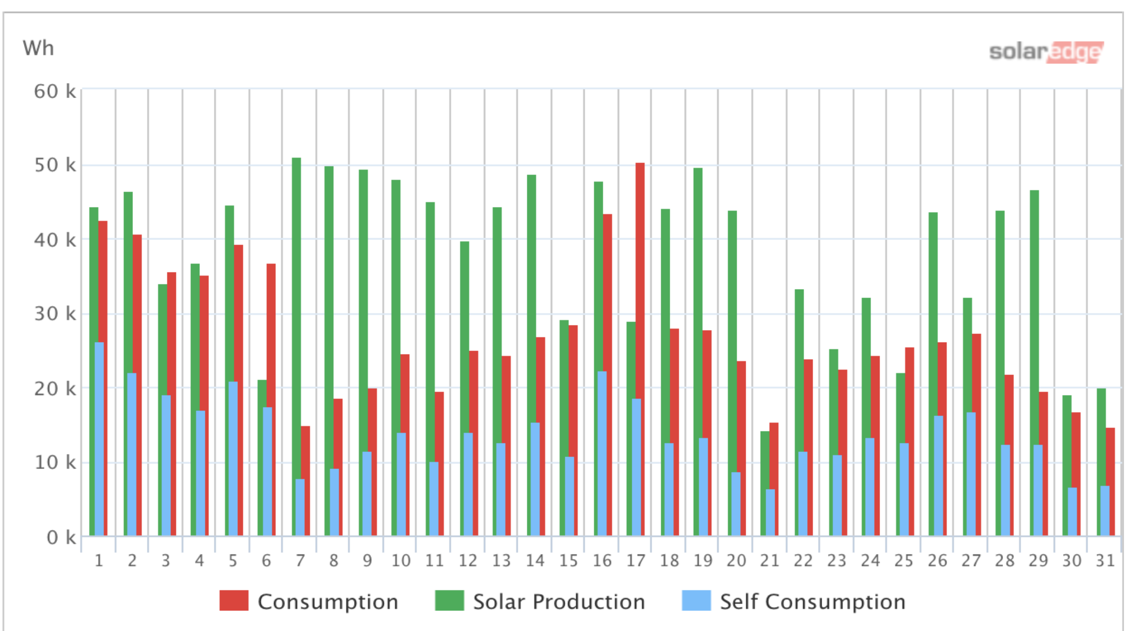


Fig 2. Consumption data for July 2018

Assumptions:

We found data on the Project Sunroof website about Easton which we used to come up with a modifier for scaling our production estimates from a single house to all houses in Easton. As solar systems are designed to be based on a house's consumption [3], we assumed that we could use the same modifier to scale consumption estimates from a single house to all houses in Easton. This assumption made sense as the larger the size of the home, the more solar energy it would require to carry out its daily operations. We also assumed that for the scope of this project we were only looking at houses in Easton that had the potential to generate solar energy according to Google Project Sunlight [4]. This was important later on because it would help us ensure that we only looked at the efficiency of solar systems and did not focus on buildings that used alternative sources of energy. We also incorporated a battery into our solar system and assumed that the energy stored in the battery would get utilized in the house's consumption before any new solar energy produced by the solar system could be used. This allowed us to ensure that the battery was being utilized to the fullest and making the system most efficient.

We used two different Allentown houses for our production and consumption data to come up with a distribution and assumed that they were representative of the production and consumption of a house in Easton. We were able to make this assumption because Allentown is close enough geographically for it to have almost the same weather and climate on a given day. Furthermore, it also has the same latitude as Easton which means it enjoys almost the same levels of sunlight. We also assumed that the data we got for production [1] and consumption [2] had already accounted for different weather conditions and optimal positioning of solar panels.

We found that the cost of buying energy from the grid was currently 14 cents/kWh [5]. Moreover, we found that the solar renewable energy credits were currently at $42/MWh [6] and the federal government was offering a tax credit of 30% on installation of solar panels [7]. Additionally, our research told us that the cost of installation of a solar battery system was $800/kWh [8]. Lastly, we deduced that the cost of installation of a solar system per watt decreased as the size of the solar system increased starting at $2.35/watt [3].

Simulation Description:

Our code consisted of four main loops which allowed us to simulate our solar system from a single home in Easton to all houses in Easton with solar potential. The innermost loop looked at consumption and production data from our Allentown houses and generated a normal monthly distribution based on each month's respective mean and standard deviation which was then used to generate a random number for daily consumption and production. A battery variable was also initialized and stored the excess energy if daily production exceeded daily consumption. However, each battery was only capable of storing energy less than or equal to its peak amount (which we varied in the code to account for different battery sizes). Thus, if there was still any energy left over after battery storage had been exceeded it was sold back to the grid. Our loop also accounted for and stored the number of days when energy production was less than energy consumption and energy had to be bought from the grid.

Our next outer loop basically followed the same process mentioned above but extended the simulation to an entire year for this single Easton house. Each month had its own production and consumption distribution which was referenced by calling the mean and standard deviation for each specific month using an array that had stored these values earlier. The number of days of each month had also been stored in an earlier array which was also utilized in this loop to make sure that each month had the data for the correct number of days. Outside the loop, we also stored important values such as total energy consumption, total energy production, total days energy was needed from the grid, total amount of energy needed from the grid, total excess energy generated etc.

The next outer loop extended our code from a single house in Easton to all homes in Easton with solar potential. The modifier was created by looking at production data found on Google Project Sunroof [4] which had a distribution of solar capacity of the houses in Easton. We used that distribution to come up with a modifier that was generated randomly but accounted for the randomness in the sizes and solar potential generation of the different houses in Easton. This loop also included the installation costs of the solar system which was scaled depending on the size of the system using a modifier. It also incorporated the 30% tax credit that the federal government offers on solar systems[7]. At the end of the loop we stored mean values of energy consumption, energy production, days energy was needed from the grid, amount of energy needed from the grid, excess energy generated for every specific house in Easton instead of saving them as total values for all houses in Easton.

Our final(outermost) loop extended our entire simulation to look at how different sized solar batteries would affect the efficiency of our solar systems. We looked at 10 different battery sizes from 0-45 kWh in increments of 5 kWh. We also had arrays to store our mean values for all of our aforementioned variables for each battery size. These arrays were then used to generate the subsequent graphs for our results.

Simulation Analysis:

We found that the use of solar energy led to a significant reduction in the days energy was needed from an outside source even when there was no battery being used. This could indicate that solar energy could be relied upon by the Easton CAP to help them achieve their goal of renewable energy. It is important to note that despite the fact that solar energy led to a large reduction in energy needed from the grid, there were still 149 days when energy was needed from the grid in the absence of a solar battery (Fig 3 below). However, even the usage of a 5kWh battery, significantly reduced the number of days outside energy needed from 149 days to 115 days only. We also saw in Fig 3 that the days grid energy was needed decreased as the size of the battery increased. While the decreases were bigger initially, the rate of decrease slowed down and leveled off at higher battery values which suggested decreasing marginal utility of batteries. Fig 4 also shows that the average yearly energy needed from the grid followed a similar trend.

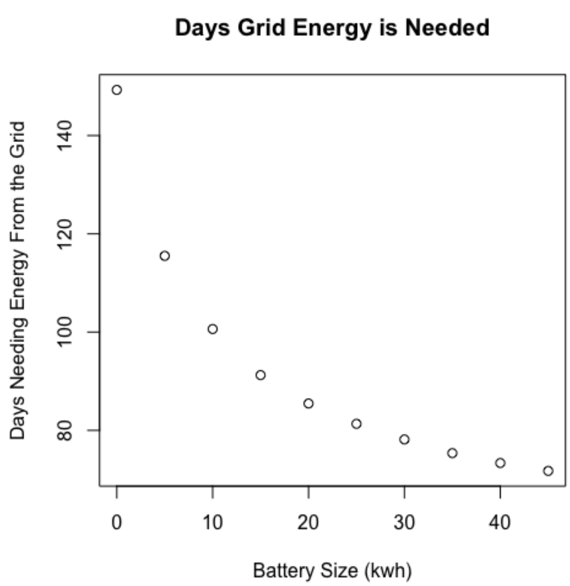


Fig 3. # Days Grid energy needed

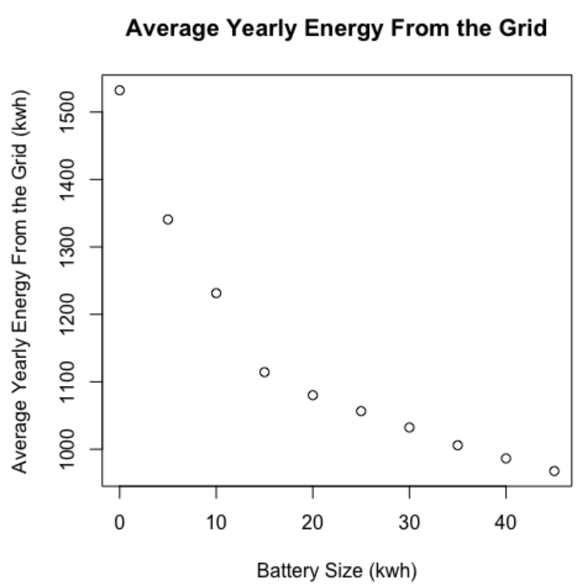


Fig 4. Avg yearly grid energy needed

Solar panels on average generated more energy than was needed for consumption which led to excess energy that could be sold back to the grid for profit. When there was no battery, there was about 2800 kWh of energy being sold back to the grid as shown in Fig 5. The use of a battery rapidly decreased the amount of energy that was being sold back to the grid as this energy was now being used to power the consumption of the house instead. As the battery size increased, the yearly energy sold back to the grid decreased and the energy stored in the battery increased as shown in Fig 5 and 6.

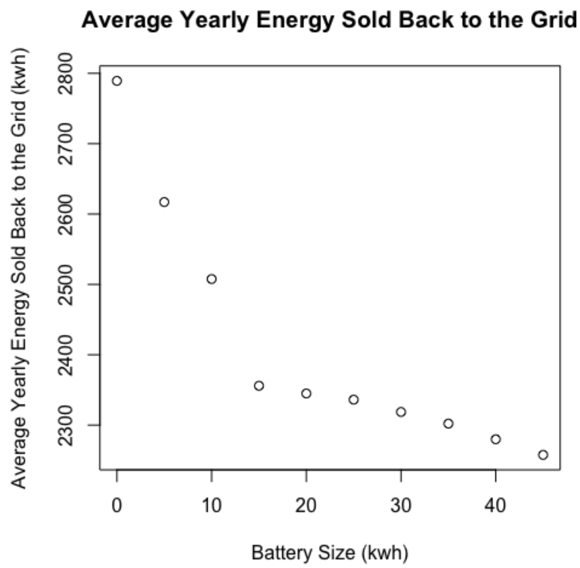


Fig 5. Avg Yearly Energy sold back

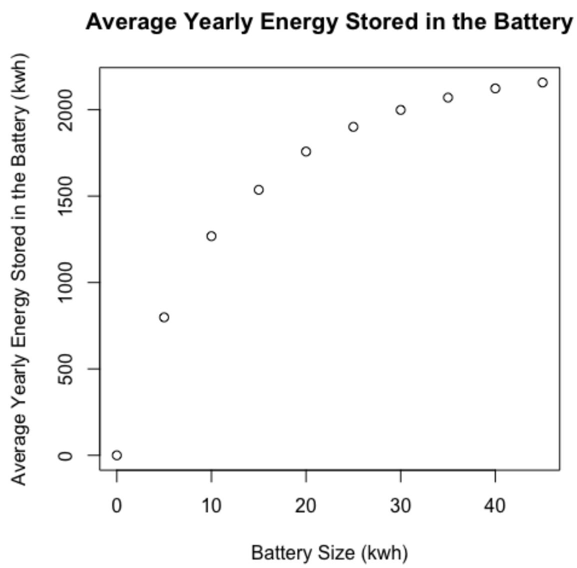
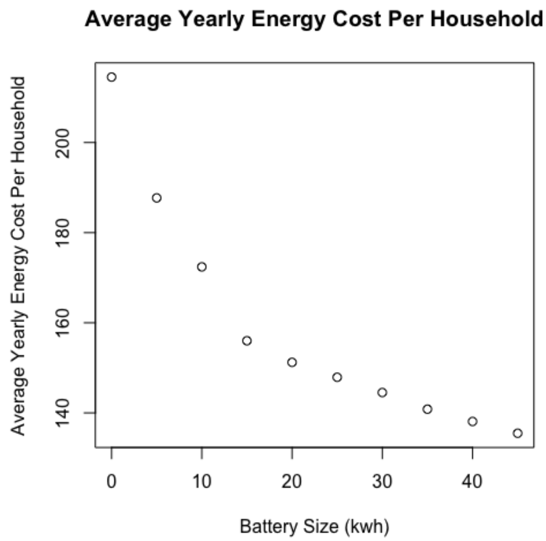


Fig 6. Avg Yearly energy stored in battery

The average yearly energy cost per household in Easton was found to be $2004 [5]. We found that once solar panels were installed the average yearly energy cost per household fell down to $218 even without any battery shown in Fig 7. This was approximately 10 times lower than the price of energy without solar panels. Moreover, using a battery further drove down this price as the battery size increased.

The average time to pay off the solar system showed a linear trend shown below in Fig 8 with the number of years to pay off the system increasing as the battery size increased. This is due to the fact that installation cost of a solar battery system increased as the size of the battery increased so it takes longer to pay off the total cost. The time taken to pay off the cost was 15 years when there is no battery involved.



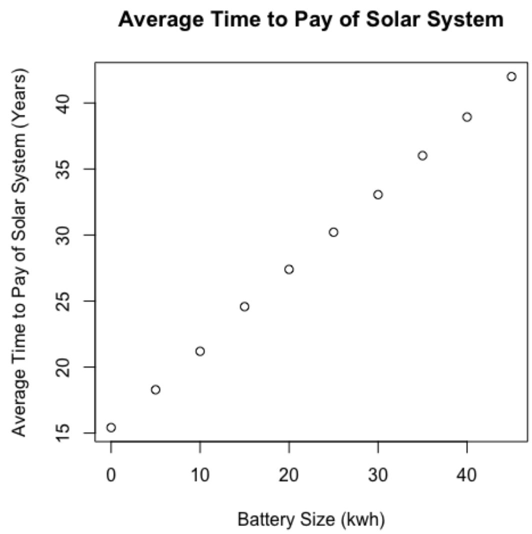


Fig 7. Avg Yearly energy cost/household

Fig 8. Avg time to pay off solar system

Discussion:

We found that the mean total energy production for a single household was greater than the mean total energy consumption on average which meant that there was excess energy left behind as shown in Table 1. This means that we must install some sort of solar battery alongside our solar panels so that any excess energy is not being wasted. If this excess energy could be stored in a battery, it could then be used to power other buildings in Easton as well.

Mean Total energy production	8120 kWh
Mean Total energy consumption	6844 kWh
Mean Excess Energy	1276 kWh

Table 1. Table showing mean excess energy generated in a single house in Easton

All of our results also showed that regardless of the size of the solar battery, there was always a need for some other source of energy. The reason for that is that solar batteries can not store energy for more than 5 days [9]. We saw that mean yearly solar production was higher than mean yearly consumption which was probably due to the fact that there was much more solar energy being produced in the summer months than the winter months. On the other hand, the summer months have lesser consumption of energy than in the winter months when a lot of energy is being used for heating homes up. While this leads to an excess of solar energy in the summer months it can not be utilized in the winter months due to the limit on how long a solar battery can store energy. It also needs to be noted that the efficiency of solar batteries decreases with increasing size so having a very large battery is also not the most efficient way of achieving the goal of renewable energy. Thus, additional sources of renewable energy than solar energy are required if Easton wants to fulfill its goal of 100% renewable energy by 2030. This energy could come from wind energy or hydro-thermal energy.

Based on the results of our simulation, we made the following two recommendations to the city of Easton in order to help them achieve their goals in the CAP. Our first solution was the most cost effective solution which involved not using any sort of solar battery because of the additional installation costs they incurred. In this case, it would take each household approximately 15 years to pay off the initial costs of installing the solar panels. The outside energy required would still be reduced by a significant amount, thus, working towards Easton's goal of renewable energy. However, it was also important to note that implementing this solution still caused a lot of generated solar energy to be wasted because of the inability to store it for long periods of time. The energy could,however, be used to provide the rest of Easton with solar energy.

The second solution was the most eco-friendly solution. This involved using a medium sized battery (15-20 kWh) alongside our solar panels. We chose this battery size because larger batteries were more costly to install but lost their efficiency as they increased in size. On the other hand, 15-20 kWh batteries were cheaper than larger batteries but stored energy with roughly the same efficiency. This system was substantially more expensive to install than having no battery and took 25 years to pay off. However, it made optimal use of all available solar energy which is a huge stride in Easton's goal of achieving renewable energy.

We believe that the residents of Easton would prefer to use the most cost effective solution if they are expected to finance this project on an individual basis. However, if the city of Easton is looking to finance this project for its residents, we believe that the most eco-friendly solution would be the most suitable one to implement as it most closely aligns with Easton's goal of working towards 100% renewable energy by 2030.

References

[1] "NPSCPurpleHouseSolar." *Solaredge*, https://monitoringpublic.solaredge.com/solaredge-web/p/site/public?name=NPSCPurpleHouseSolar#/dashboard.

[2] "GSE-ALLENTOWN-21." *Solaredge*, https://monitoringpublic.solaredge.com/solaredge-web/p/site/public?name=GSE-ALLENTOWN-21&locale=en_US#/dashboard.

[3] Savino, James. "Solar Panel Cost Guide in Easton." *EcoWatch*, Oct. 2022, https://www.ecowatch.com/solar/panel-cost/pa/easton.

[4] "Project Sunroof - Data Explorer | Easton." *Google*, Google, https://sunroof.withgoogle.com/data-explorer/place/ChIJX1hwGIJrxIkRK8iPdCcyIDc/

[5] "Cost of Electricity in Easton, PA." *EnergySage*, 23 Oct. 2022, https://www.energysage.com/local-data/electricity-cost/pa/northampton-county/easton/#:~:text=The%20average%20residential%20electricity%20rate,rate%20of%2018%20%C2%A2%2FkWh.

[6] Thoubboron, K. (2022, August 31). *SRECs: Solar Renewable Energy Credits explained: Energysage*. EnergySage Blog. Retrieved October 30, 2022, from https://news.energysage.com/srecs-complete-overview/

[7] "Homeowner's Guide to the Federal Tax Credit for Solar Photovoltaics." *Energy.gov*, https://www.energy.gov/eere/solar/homeowners-guide-federal-tax-credit-solar-photovoltaics

[8] *How Much Does Home Solar Battery Cost per Kwh?* https://www.bsl-battery.com/how-much-does-home-solar-battery-cost-per-kwh.html.

[9] Aurorasolardev, A. (2022, July 19). *Storing solar energy: Everything you need to know*. Aurora Solar. Retrieved November 2, 2022, from https://aurorasolar.com/blog/storing-solar-energy-everything-you-need-to-know/

Appendix:

```
#Get Estimates of Solar Data Production
#Loading Data and Calculating Distribution Parameters

setwd("~/Documents/Math 301")

#Initialize data arrays for each month
productionmean = 1:12
productionsd = 1:12
consumptionmean = 1:12
consumptionsd = 1:12
#January Production
solarJan = read.csv("JanProduction.csv", header=T)
productionmean[1] = mean(solarJan$System.Production..Wh.)
productionsd[1] = sd(solarJan$System.Production..Wh.)
#February
solarJan = read.csv("FebProduction.csv", header=T)
productionmean[2] = mean(solarJan$System.Production..Wh.)
productionsd[2] = sd(solarJan$System.Production..Wh.)
#March
solarMar = read.csv("MarProduction.csv", header=T)
productionmean[3] = mean(solarMar$System.Production..Wh.)
productionsd[3] = sd(solarMar$System.Production..Wh.)
#April
solarApr = read.csv("AprProduction.csv", header=T)
productionmean[4] = mean(solarApr$System.Production..Wh.)
productionsd[4] = sd(solarApr$System.Production..Wh.)
#May
solarMay = read.csv("MayProduction.csv", header=T)
productionmean[5] = mean(solarMay$System.Production..Wh.)
productionsd[5] = sd(solarMay$System.Production..Wh.)
#June
solarJun = read.csv("JunProduction.csv", header=T)
productionmean[6] = mean(solarJun$System.Production..Wh.)
```

```
productionsd[6] = sd(solarJun$System.Production..Wh.)
#July
solarJul = read.csv("JulProduction.csv", header=T)
productionmean[7] = mean(solarJul$System.Production..Wh.)
productionsd[7] = sd(solarJul$System.Production..Wh.)
#August
solarAug = read.csv("AugProduction.csv", header=T)
productionmean[8] = mean(solarAug$System.Production..Wh.)
productionsd[8] = sd(solarAug$System.Production..Wh.)
#September
solarSep = read.csv("SepProduction.csv", header=T)
productionmean[9] = mean(solarSep$System.Production..Wh.)
productionsd[9] = sd(solarSep$System.Production..Wh.)
#October
solarOct = read.csv("OctProduction.csv", header=T)
productionmean[10] = mean(solarOct$System.Production..Wh.)
productionsd[10] = sd(solarOct$System.Production..Wh.)
#November
solarNov = read.csv("NovProduction.csv", header=T)
productionmean[11] = mean(solarNov$System.Production..Wh.)
productionsd[11] = sd(solarNov$System.Production..Wh.)
#December
solarDec = read.csv("DecProduction.csv", header=T)
productionmean[12] = mean(solarDec$System.Production..Wh.)
productionsd[12] = sd(solarDec$System.Production..Wh.)

#Consumption
#January Consumption
Jan2018 = read.csv("Jan2018.csv", header=T)
attach(Jan2018)
consumptionmean[1]=mean(Consumption..Wh.)
consumptionsd[1]=sd(Consumption..Wh.)
#February Consumption
Feb2018 = read.csv("Feb2018.csv", header=T)
```

```
attach(Feb2018)
consumptionmean[2]=mean(Consumption..Wh.)
consumptionsd[2]=sd(Consumption..Wh.)
#March Consumption
March2018 = read.csv("March2018.csv", header=T)
attach(March2018)
consumptionmean[3]=mean(Consumption..Wh.)
consumptionsd[3]=sd(Consumption..Wh.)
#April Consumption
Apr2018 = read.csv("Apr2018.csv", header=T)
attach(Apr2018)
consumptionmean[4]=mean(Consumption..Wh.)
consumptionsd[4]=sd(Consumption..Wh.)
#May Consumption
May2018 = read.csv("May2018.csv", header=T)
attach(May2018)
consumptionmean[5]=mean(Consumption..Wh.)
consumptionsd[5]=sd(Consumption..Wh.)
#June Consumption
Jun2018 = read.csv("Jun2018.csv", header=T)
attach(Jun2018)
consumptionmean[6]=mean(Consumption..Wh.)
consumptionsd[6]=sd(Consumption..Wh.)
#July Consumption
July2018 = read.csv("July2018.csv", header=T)
attach(July2018)
consumptionmean[7]=mean(Consumption..Wh.)
consumptionsd[7]=sd(Consumption..Wh.)
#August Consumption
Aug2018 = read.csv("Aug2018.csv", header=T)
attach(Aug2018)
consumptionmean[8]=mean(Consumption..Wh.)
consumptionsd[8]=sd(Consumption..Wh.)
#September Consumption
```

```
September2018 = read.csv("September2018.csv", header=T)
attach(September2018)
consumptionmean[9]=mean(Consumption..Wh.)
consumptionsd[9]=sd(Consumption..Wh.)
#October Consumption
October2018 = read.csv("October2018.csv", header=T)
attach(October2018)
consumptionmean[10]=mean(Consumption..Wh.)
consumptionsd[10]=sd(Consumption..Wh.)
#November Consumption
Nov2018 = read.csv("Nov2018.csv", header=T)
attach(Nov2018)
consumptionmean[11]=mean(Consumption..Wh.)
consumptionsd[11]=sd(Consumption..Wh.)
#December Consumption
December2018 = read.csv("December2018.csv", header=T)
attach(December2018)
consumptionmean[12]=mean(Consumption..Wh.)
consumptionsd[12]=sd(Consumption..Wh.)

set.seed(10) #Keep randomness in loops keep results outside constant for comparison purposes

#Battery initialization and array setup
batterysize = c(0,5000,10000,15000,20000,25000,30000,35000,40000,45000)
armeantotenergyProd = 1:10 #Average total energy production for a house in Easton
armeantotenergyCon = 1:10 #Average total energy consumption for a house in Easton
armeantotdaysneedenergy = 1:10 #Days solar energy produced was not enough to cover all consumption
armeantotneedenergy = 1:10 #Average amount of energy a house with solar would need to get from the grid over 1 year
armeantexcesseng = 1:10 #Average amount of energy a house can sell back to the grid in a year
armeantbatteryusage = 1:10 #Average total energy stored in the battery at any point in a year
```

```
armpricewosolar = 1:10 #Average cost of energy without solar for a year
armpricewsolar = 1:10 #Average cost of energy with solar for a year
armenergysold = 1:10 #Average value of energy sold back to the grid over a year
armmoneysaved = 1:10 #Average total money saved in a year with solar
armtotalcost = 1:10 #Cost of a solar system including battery installation
armtimetopay = 1:10 #Average number of years to pay back the cost of solar installation

for(b in 1:13){ #Running each town simulation with

  #Home Simulation
  #Initialize City of Easton variables (Outer Loop)
  monthdays = c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
  totenergyProd = 0
  totenergyCon = 0
  totdaysneedenergy = 0
  totneedenergy = 0
  texcesseng = 0
  tbatteryusage = 0 #How much total energy has the battery stored
  tinstallcost = 0 #Installation cost
  peakpower = batterysize[b] #Normal battery can hold 10Kwh of energy
  ecount = 0
  N = 5400 #Number of houses/structures capable of energy production in Easton

  for (i in 1:N){ #All Easton solar capable homes
    pmodifyer = 0
    randSamp = runif(1,0,1) #Generating the modifier for production and consumption
    if (randSamp >= 0 && randSamp < 0.41){ #average 2.5 kwH system
      pmodifyer = runif(1,.1,.51) #The portion of these house energy production to our choose
house
      installcost = 6625 * .70 #2.65 per watt times 2,500 watt system and a 30% tax credit
      tinstallcost = tinstallcost + installcost
    } #Modifier is based on midpoint of graph
    else if (randSamp >= 0.41 && randSamp < 0.76){ #average 7.5 kwH system
      pmodifyer = runif(1,.51,1.03)
```

```
  installcost = 19500 * .70 #2.60 at this cost
  tinstallcost = tinstallcost + installcost
}
else if(randSamp >= 0.76 && randSamp < 0.87){ #average 12.5 kwH system
  pmodifyer = runif(1,1.03,1.54)
  installcost = 31250 * .70 #2.50 at this cost
  tinstallcost = tinstallcost + installcost
}
else if (randSamp >= 0.87 && randSamp <= 1){ #average 17.5 kwH and greater system
  pmodifyer = runif(1,1.54,3.09)
  installcost = 41125 * .70 #2.35 at this cost
  tinstallcost = tinstallcost + installcost
}
#Initialize inner variables
energyProd = 0
energyCon = 0
daysneedenergy = 0
needenergy = 0
battery = 0
bu = 0 #energy that has ever been in the battery
excesseng = 0
count = 0 #Days with energy going back to the gird
#Single Home Generation
#January
for(k in 1:12){
  numdays = monthdays[k] #Recall the number of days in each month
  for (j in 1:numdays){
    #First generating production and consumption
    dayprod = rnorm(1,productionmean[k],productionsd[k]) * pmodifyer
    daycon = rnorm(1,consumptionmean[k],consumptionsd[k]) * pmodifyer
    while (dayprod && daycon < 0) #Making sure values are non negative
    {
      dayprod = rnorm(1,productionmean[k],productionsd[k])
      daycon = rnorm(1,consumptionmean[k],consumptionsd[k])
```

```
      }
      energyProd = energyProd + dayprod
      energyCon = energyCon + daycon
      #Taking into account the battery
      dayeng = battery + dayprod #All energy available
      usage = dayeng - daycon #Enough energy for the day?
      if(usage < 0){ #Not enough energy produced to cover the day
        daysneedenergy = daysneedenergy +1
        needenergy = needenergy + abs(usage)
        battery = 0
      } else if (usage > 0 && usage < peakpower){ #Energy to spare
        bu = bu + dayprod - daycon #Energy stored in the battery is this diff. in consumption and production
        battery = usage
      } else if (usage >= peakpower){ #Too much energy to store
        peakbat = 0
        if(dayprod-daycon >= peakpower){ #Making sure there is never more energy stored than the battery can produce
          peakbat = peakpower #Can never hold more than peak power
        } else{
          peakbat = dayprod - daycon #Less than peak power when battery charge is large enough
        }
        bu = bu + peakbat
        battery = peakpower
        excesseng = excesseng + (usage-peakpower)
        count = count +1
      }
    }
  }
  #Carrying the inner Variables to the total variables
  totenergyProd = totenergyProd + energyProd
  totenergyCon = totenergyCon + energyCon
  totdaysneedenergy =totdaysneedenergy + daysneedenergy
```

```
    totneedenergy = totneedenergy + needenergy
    texcesseng = texcesseng + excesseng
    tbatteryusage = tbatteryusage + bu
    ecount = ecount + count
  }

  #Calculating all the means in KwH
  meantotenergyProd = (totenergyProd/N)/1000
  armeantotenergyProd[b] = meantotenergyProd
  meantotenergyCon = (totenergyCon/N)/1000
  armeantotenergyCon[b] = meantotenergyCon
  meantotdaysneedenergy = totdaysneedenergy/N
  armeantotdaysneedenergy[b] = meantotdaysneedenergy
  meantotneedenergy = (totneedenergy/N)/1000
  armeantotneedenergy[b] = meantotneedenergy
  meantexcesseng = (texcesseng/N)/1000
  armeantexcesseng[b] = meantexcesseng
  meantbatteryusage = (tbatteryusage/N)/1000
  armeantbatteryusage[b] = meantbatteryusage
  #Cost means
  #$800 per KwH
  batterycost = ((peakpower/1000)*800)*.70 #Cost of battery (The same for all houses in each
inner loop) and is eligible for tax rebate
  meantinstallcost = tinstallcost/N + batterycost

  #Cost calculation
  mpricewosolar = 0.14 * meantotenergyCon #All energy consumed
  armpricewosolar[b] = mpricewosolar
  mpricewsolar = 0.14 * meantotneedenergy #Energy still needed with battery
  armpricewsolar[b]  = mpricewsolar
  menergysold = (meantexcesseng/1000)*32.5 #SREC credits pay you on average $32.5 per
MWh
  armenergysold[b] = menergysold
```

```
  mmoneysaved = mpricewosolar - mpricewsolar + menergysold #Money saved from using less
energy and gained from net metering
  armmoneysaved[b] = mmoneysaved
  mtotalcost = meantinstallcost - mmoneysaved #Loss after 1 year
  armtotalcost[b] = mtotalcost
  mtimetopay = meantinstallcost/mmoneysaved #average time to make back cost of solar
installation and battery
  armtimetopay[b] = mtimetopay

}
#Printing Values of arrays
print(armeantotenergyProd)
print(armeantotenergyCon)
print(armeantotdaysneedenergy)
print(armeantotneedenergy)
print(armeantexcesseng)
print(armeantbatteryusage)
print(armpricewosolar)
print(armpricewsolar)
print(armenergysold)
print(armmoneysaved)
print(armtotalcost)
print(armtimetopay)

#Plotting important factors based in battery size
plot(batterysize,armeantotdaysneedenergy,xlab = "Battery Size (kwh)", ylab = "Days Needing
Energy From the Grid")
title(main = "Days Grid Energy is Needed")
plot(batterysize,armeantotneedenergy,xlab = "Battery Size (kwh)", ylab = "Average Yearly
Energy From the Grid (kwh)")
title(main = "Average Yearly Energy From the Grid")
plot(batterysize,armeantexcesseng,xlab = "Battery Size (kwh)", ylab = "Average Yearly Energy
Sold Back to the Grid (kwh)")
title(main = "Average Yearly Energy Sold Back to the Grid")
```

```
plot(batterysize,armeantbatteryusage,xlab = "Battery Size (kwh)", ylab = "Average Yearly Energy Stored in the Battery (kwh)")
title(main = "Average Yearly Energy Stored in the Battery")
plot(batterysize,armpricewsolar,xlab = "Battery Size (kwh)", ylab = "Average Yearly Energy Cost Per Household")
title(main = "Average Yearly Energy Cost Per Household")
plot(batterysize,armenergysold,xlab = "Battery Size (kwh)", ylab = "Average Yearly Profit for Selling Excess Energy")
title(main="Average Yearly Profit for Selling Excess Energy")
plot(batterysize,armmoneysaved,xlab = "Battery Size (kwh)", ylab = "Average Yearly Total Money Saved With Solar")
title(main="Average Yearly Total Money Saved With Solar")
plot(batterysize,armtotalcost,xlab = "Battery Size (kwh)", ylab = "Average Total Cost of Solar System")
title(main="Average Total Cost of Solar System")
plot(batterysize,armtimetopay,xlab = "Battery Size (kwh)", ylab = "Average Time to Pay of Solar System (Years)")
title(main="Average Time to Pay of Solar System")

#Printing important total values
print(mean(armeantotenergyProd))
print(mean(armeantotenergyCon))
```