Is It Possible to Create a Mathematical Model to Predict Photopollution Based on Population Density in Munster?

Conor Casey and Hannah Coombs

Teacher: Ms. Abbott

Pobalscoil Inbhear Sceine

Project Diary

Stand 3502 - BTYSTE 2019

“The universe is not required to be in perfect harmony with human ambition.”

— Carl Sagan

Weekly Events

The Week of the 10th of July, 2018

During this week, Conor started to lay out a plan of wanted we wanted to do and include in this project over the next few months. This project is a continuation of my SciFest project “How Population Density influences Light Pollution?”. Conor did some initial research for the project proposal. He also worked out what my experimental method is going to be and the necessary equipment that will need to be purchased. Conor also wrote down all the different town, cities, and villages that we are going to be collecting data from. Finally, Conor wrote the initial project proposal draft. Conor spent approximately 8 hours doing all of the above.

The Week of the 30th of July, 2018

During this week, a significant portion of the draft Introduction section of the Project Book was written as well as the framework for the Python program. Conor spent approximately 12 hours doing all of this over the last 2 days.

The Week of the 27th of August, 2018

During this week, version 0.1.0 of the main Python program was completed. Also, a draft of the Introduction section of the Project Book was finally written. It was approximately three pages, and was heavily based on the Project Book from Conor’s SciFest project.

The Week of the 10th of September, 2018

During this week, David O’Sullivan becomes the second group member to join Team Stargazers alongside Conor. Also, the pitch for this project was presented to our Science teacher, Ms. Abbott, at the school’s Science Club meeting about the BTYSTE. Following which, we immediately started working on the project submission requirements.

The Week of the 17th of September, 2018

Hannah Coombs officially joins the Team Stargazers :)

The Week of the 24th of September, 2018

Over the previous weekend, Conor, the Group Leader, submitted our project for consideration to be part of the BTYSTE. The One Page Project Proposal we submitted is included later in the Project Diary.

The Week of the 22nd of October, 2018

During this week, we got word that our project qualified to compete the in the BT Young Scientist and Technology Exhibition at the RDS in Dublin, from the 9th to the 12th of January 2019.

The Week of the 12th of November 2018

During this week, we carried out our data collection. Also, Conor created the Solver python program. This program printed the two data sets, population density, and photopollution, printed the correlation coefficient, printed the equation describing the relationship. It also graphed this data. We will now describe how we carried out the data collection.

Environmental Conditions

The experiment was carried out between the 11th of November 2018 & the 18th of November 2018. It was typically carried out between the hours of 18:00 & 00:00. We collected this data during the winter time as the night sky emerges earlier in the evening. This experiment was carried out during the Moon's waxing crescent phase. This was done in order to avoid sky glow from the Moon artificially increasing the photopollution of the town. There was a little precipitation when we collected data from the town of Tipperary, however, the weather cleared up thereafter. It was important that this experiment was carried out in November rather then December, as the Christmas lights put up during the latter period would have also artificially increased the photopollution value.

Locations We Visited

First Night

- Blackwater (Kerry)

- Tipperary Town (Tipperary)
- Cahir (Tipperary)
- Clonmel (Tipperary)
- Ballymacarbry (Waterford)
- Lismore (Waterford)
- Tallow (Waterford)
- Watergrasshill (Cork)
- Cork City (Cork)
- Macroom (Cork)

Second Night

- Tarbert (Kerry)
- Kilrush (Clare)
- Ennis (Clare)
- Shannon (Clare)
- Limerick City (Limerick)
- Adare (Limerick)
- Newcastle West (Limerick)
- Tralee (Kerry)
- Killarney (Kerry)
- Kenmare (Kerry)

Other Aspects

Time frame in which the experiment was carried out

- 20:00 - 03:00 (First Night)
- 17:00 - 01:00 (Second Night)

Drivers

- Ms. Abbott (First Night)
- Brian Harmon (Second Night)

Data Collection

- We placed the telescope as close to the centre of the town as physically possible. (We used the same telescope eyepiece lens, photometer app and telescope throughout the experiment)
- Once the telescope was set up, we placed the smartphones lens on the eyepiece lens of the telescope. We then opened the smartphone photometer application and prepared for recording the data.
- We recorded our measurements with the telescope pointing towards zenith. This value is the photopollution produced by this particular area. We also recorded a maximum and minimum LUX value from that same portion of the sky.
- We repeated Steps 1 - 4 at 20 different locations. We traveled to these sites using the fantastic Renault Clio, and included all the counties of Munster in this project.

The Week of the 19th of November, 2018

A draft version of the Experimental Method was completed, alongside, a second draft version of the Introduction section. Significant changes were also made to our Photopollution Calculator. We expect it to be released on GitHub on the 1st of December, 2018.

The Week of the 26st of November, 2018

A draft version of the Results, and References sections were completed. We offically launched the Photopollution Calculator on GitHub, and it is now available for use. We also plan on creating an app for release on the Google Play Store. We also had our first email correspondences with Professor Brian Espey, professor of Astrophysics at Trinity College Dublin.

The Week of the 3th of December, 2018

Unfortunately, David is unable to continue to being a group member of this project. We would like to thank him for his assistance in collecting data over the two separate days. He, unfortunately, had to drop out of this project for unforeseen health circumstances. A first draft version of the Project Book was finally completed.

The Week of the 10th of December, 2018

Significant changes were made to our Solver python program. This included making graphs for use in the Project Book in it. All of these graphs were made using a python module called Mathplotlib. It also now printed printed the standard deviation of the photopollution values we collected, and the correlation coefficient between population and photopollution. This meant the Results section had to be completely redone. Also, a good portion of the Introduction and Conclusions sections had to be redone. This marks the third draft of the Introduction.

The Week of the 17th of December, 2018

We developed an app called the Photopollution Calculator on a platform known as AppyPie. The Andriod version of the app was released on the Google Play Store. It has approximately 50 users at the time of writing, and will hopefully continue to grow in user base with the future release on the Apple App Store. The second draft of the Project Book was also completed. Conor made, using QGIS, a population density heat map, and a photopollution map of the Republic of Ireland using data provided by the Central Statistics Office. We plan on comparing these maps with a nighttime survey provided by the VIIRS March (2018) Radiance. We will add this into the Project Book next week.

The Week of the 24th of December, 2018

During this week, the Project Book was completely reformatted, and some sections were rewritten. Previously, the Project Book was written in a Word processor called "Pages", however, it was converted and now complied in LaTeX, specially using an online LaTeX editor called, Overleaf. Also, a new version of the Photopollution Calculator was released, and added features, such as, astronomy weather (provided by Accuweather).

The Week of the 31st of December, 2019

After a couple of sleepless nights rewriting some of the Project Book, Conor finally completed it. Our amazing teacher, Ms. Abbott, then got it printed and bounded. Also, our Photopollution Calculator app was finally submitted and approved for release on the Apple App Store.

The Week of the 7th of January, 2019

The week of the BTYSTE 2019, and when we finally completed the Project Diary in its entirety. Also, you (the reader) appear to have read all of the weekly events in this Project Diary, why? What were you expecting to uncover?

Bibliography

[1] I. D.-S. R. Kerry, *About us*, (accessed 2018-11-25) http://kerrydarksky.com/.

[2] IDA, *International dark sky reserves*, (accessed 2018-11-25) https://www.darksky.org/our-work/conservation/idsp/reserves/.

[3] F. Verheijin, "Photopollution: artificial light optic spatial control systems fail to cope with. incidents, causation, remedies", Experimental biology (1983).

[4] H. van de Groenendall, "Man-made pollution of radio spectrum is a growing problem", EngineerIT (January 28th, 2016).

[5] G. Matt Rosenberg, *Population density*, (accessed 2018-12-03) http://geography.about.com/od/populationgeography/a/popdensity.htm.

[6] B. Plumer, *The night sky is vanishing: 80% of americans can no longer see the milky way galaxy*, (accessed 2018-11-25) https://www.vox.com/2016/6/10/11905390/light-pollution-night-sky.

[7] K. O'Sullivan, *Only 5 per cent of ireland's night skies are free from artificial light says expert*, (accessed 2018-11-25) https://www.irishtimes.com/news/science/only-5-per-cent-of-ireland-s-night-skies-are-free-from-artificial-light-says-expert-1.3419673.

[8] A. about Ireland, *Light pollution*, (accessed 2018-12-17) http://www.askaboutireland.ie/enfo/irelands-environment/noise/light/.

[9] S. T. Conservancy, *Information about sea turtles: threats from artificial lighting*, (accessed 2018-12-17) https://conserveturtles.org/information-sea-turtles-threats-artificial-lighting/.

[10] B. Espey, *Light pollution in the irish context: historical changes recent data*, (accessed 2018-11-25) https://www.tcd.ie/Physics/news-events/events/light@night/abstracts/pdfs/Espey_LAN_20Apr2016_for_web.pdf.

[11] C. S. Office, *A0106: 1996 population density, and area size by electoral division, census year, and statistic*, (accessed 2018-12-30) https://www.cso.ie/px/pxeirestat/statire/SelectVarVal/Define.asp?MainTable=A0106&PLanguage=0&PXSId=0.

[12] C. S. Office, *E2014: population density, and area size 2016 by towns by size, censusyear, and statistic*, (accessed 2018-12-30) https://www.cso.ie/px/pxeirestat/statire/SelectVarVal/Define.asp?MainTable=E2014&PLanguage=0&PXSId=0.

[13] C. S. Office, *E2013: population density, and area size 2016 by electoral division, census year, and statistic*, (accessed 2018-12-30) https://www.cso.ie/px/pxeirestat/statire/SelectVarVal/Define.asp?MainTable=E2013&PLanguage=0&PXSId=0.

[14] C. POPULATION, *Munster, providence in ireland.* (accessed 2018-12-30) https://www.citypopulation.de/php/ireland-admin.php?adm1id=MUN.

[15] A. Upgren, "The measurement of light pollution and the population of cities", IAU Colloq 112 *LightPollution RadioInterference andSpaceDebris* **17**, 79 (1991).

[16] R. C. Shirkey, "Sky glow from cities *thearmyilluminationmodel*", Army Research Laboratory **2**, 2–6 (September 2011).

[17] B. E. Newling, "The spatial variation of urban population densities", Geographical Review **59**, 242–252 (1969).

[18] N. N. G. D. C. Earth Observation Group, *Nighttime imagery of dublin provided by the viirs 2018 (march) radiance*, (March 2018) https://www.lightpollutionmap.info/#zoom=9&lat=7052493&lon=-696059&layers=B0TFFFFFFFF.

[19] N. N. G. D. C. Earth Observation Group, *Nighttime imagery of madurai provided by the viirs 2018 (march) radiance*, (March 2018) https://www.lightpollutionmap.info/#zoom=10&lat=1110319&lon=8696742&layers=B0TFFFFFFFF.

[20] NLPIP, "How sky glow is measured or calculated", **7** (February 2007).

[21] O. D. of English, *Zenith*, (accessed 2018-12-10) https://en.oxforddictionaries.com/definition/zenith.

[22] M. Keating, *Active maths 4* (Folens Publisher, 2012).

[23] P. Chen, and A. Krauss, "The sage encyclopedia of social science research methods", 808–810 (2004).

[24] C. Casey, *Photopollution calculator*, (2019) https://github.com/conorcaseyc/Photopollution-Calculator.

[25] C. Casey, *Photopollution calculator*, (2019) https://play.google.com/store/apps/details?id=com.app.photopollutioncalculator.

[26] QGIS Development Team, *Qgis geographic information system*, Open Source Geospatial Foundation (2009).

[27] N. N. G. D. C. Earth Observation Group, *Nighttime imagery of ireland provided by the viirs 2018 (9th, of march 2018) radiance*, (March 2018) https://www.lightpollutionmap.info/#zoom=9&lat=7052493&lon=-696059&layers=B0TFFFFFFFF.

[28] K. Telescopes, *Celestron astromaster 102az with 20 mm telescope eyepiece lens*, (accessed 2018-11-25) http://www.ktectelescopes.ie/Christmas-Gift-Ideas/Christmas-Gift-Ideas-Telescopes-Under-250/celestron-astromaster-102az.html.

[29] E. Polyanskaya, *Lux light meter pro*, (accessed 2018-11-25) https://itunes.apple.com/ie/app/lux-light-meter-pro/id1292598866?mt=8.

[30] Toshiba, *Toshiba satellite l50-b-1nm*, (accessed 2018-11-25) `http://www.toshiba.ie/discontinued-products/satellite-l50-b-1nm/`.

[31] G. van Rossum, *Python tutorial*, tech. rep. CS-R9526 (Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995).

[32] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: open source scientific tools for Python*, [Online; accessed 2018-12-31], 2001–.

[33] Renault, *Renault clio*, (2012) `https://www.renault.ie/vehicles/new-vehicles/clio.html`.

[34] G. W. H. Carl W Hall, "Dictionary of energy", 84 (1983).

[35] C. S. Office, *Electoral division boundaries*, (accessed 2018-12-21) `http://census.cso.ie/censusasp/saps/boundaries/eds_bound.htm`.

Project Proposal

The purpose of this project is to determine if there is a correlation between population density and photopollution, if so, can I create a mathematical model to predict it? Using this model, can I extrapolate what the data collected means. Photopollution is defined as the presence of anthropogenic visible light in the night environment.

It is apparent that densely populated areas tend to have far more photopollution than sparsely populated ones. Just look at an aerial photograph of a city taken during the night compared to a more rural area. In this modern world, photopollution data is provided by satellites so as a consequence the idea for this project came to me when I was pondering whether it would be possible to determine the severity of photopollution without the need of satellites.

To investigate this problem, I would measure the photopollution produced in an area. I would do this in accordance with the method cited by the National Lighting Product Information Program in one of their articles. This method is often used by professional astronomers, it involves using an instrument called a photoelectric photometer in combination with a telescope. They measure the dark portion of the sky with said astronomical instruments. Typically such measurements are made at the zenith.

I would use this experimental method in my experiment to measure the photopollution at various locations. The photometers are easily accessible and can be bought for around 17 on Amazon. I already have a telescope removing the need to purchase one. This photometer has a high accuracy with an error rate of +- 4%. At worst to include 15 different sites in the experiment, it should take approximately one month, which is sufficient time to complete this project before the BTYSTE in January. I plan to include 15 sites in my experiment. The population density data necessary for this experiment is provided by the results from the Census in the year 2016 which can be accessed from the Central Statistics Office website. In order to create a mathematical model and find a relationship between these two variables, I would use a scatter plot and a trend line in Excel, however, I am exploring other options at this present time. I could possibly use another program called GeneticSharp. To determine what these values mean I would use the photopollution grading data provided by Globe At Night, which is a website where amateur astronomers grade the quality of photopollution in their area, in order to extrapolate what the data collected means in relation to the severity of photopollution. However, again I am exploring other options for grading and understanding the data. A Python Program will be developed with the mathematical model developed built in to show what a potential use for the model could be.

Preliminary results show that there is a correlation between the two variables however it was over a relatively low sample size and as such more data is required.

Python Code

Solver Program

```
from numpy import *
from scipy.interpolate import *
from matplotlib.pyplot import *
from scipy.stats import *

#The Data
population_density = array([2.4, 1205.4, 1351.3, 1636.4,
   748.5, 1254, 1064, 1892, 3300, 1230.3, 658, 821.6, 1286.6,
    1387.5, 1591.2, 1292, 1423.6, 1206.6, 1796.3, 768.5])
population = array ([146, 4979, 3593, 17140, 138, 1374, 946,
   1346, 119320, 3765, 540, 3287, 25276, 9729, 94192, 1129,
   6619, 23691, 14504, 2175])
lux = array([1, 21, 28, 38, 14, 34, 21, 50, 112, 25, 12, 14,
   42, 28, 48, 24, 29, 20, 51, 12])
print population_density
print population
print lux

#Finds the formula to describe the correlation between
   population density and photopollution
p1 = polyfit(population_density,lux,1)
print(p1)
#Finds the correlation coefficient in the case of population
   density
slope,intercept,r_value,p_value,std_err = linregress(
   population_density,lux)
correlation = (pow(r_value,2))
print correlation
#Finds the correlation coefficient in the case of population
slope,intercept2,r_value2,p_value2,std_err2 = linregress(
   population,lux)
correlation2 = (pow(r_value2,2))
```

```
print correlation2
#Calculate Standard Deviation
print std(lux)
#Gets the root of the equation
root = 14.32414198 / 0.03510566
print root

#Plotting Graphs
#Mean LUX Values Scattering Plot
scatter(population_density,lux)
title("Mean␣LUX␣Values␣(All␣Sites)")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(population_density), poly1d(p1)(unique(
   population_density)))
text(0,100, "r␣=␣" + str(round(correlation, 3)))
stddev = std(lux)
errorbar(population_density, lux, stddev, color='g',
   linestyle='None')
show()

#Rest of Graphs Data
max_lux = array([1, 23, 32, 44, 16, 44, 23, 55, 125, 34, 16,
   16, 52, 32, 52, 32, 34, 23, 58, 16])
min_lux = array([0, 18, 23, 32, 11, 23, 16, 44, 98, 16, 7,
   11, 32, 23, 44, 16, 23, 16, 44, 7])
firsttrip_pd = array([2.4, 1205.4, 1351.3, 1636.4, 748.5,
   1254, 1064, 1892, 3300, 1230.3])
secondtrip_pd = array ([658, 821.6, 1286.6, 1387.5, 1591.2,
   1292, 1423.6, 1206.6, 1796.3, 768.5])
firsttrip_pp = array([1, 21, 28, 38, 14, 34, 21, 50, 112,
   25])
secondtrip_pp = array([12, 14, 42, 28, 48, 24, 29, 20, 51,
   12])
kerry_pd = array([2.4, 658, 1206.6, 1796.3, 768.5])
kerry_pp = array([1, 12, 20, 51, 12])
tipp_pd = array([1205.4, 1351.3, 1636.4])
tipp_pp = array([21, 28, 38])
water_pd = array([748.5, 1254, 1064])
water_pp = array([14, 34, 21])
cork_pd = array([1892, 3300, 1230.3])
cork_pp = array([50, 112, 25])
clare_pd = array([821.6, 1286.6, 1387.5])
clare_pp = array([14, 42, 28])
lim_pd = array([1591.2, 1292, 1423.6])
```

```
lim_pp = array([48, 24, 29])
#p(n)
p2 = polyfit(population_density,max_lux,1)
p3 = polyfit(population_density,min_lux,1)
p4 = polyfit(firsttrip_pd,firsttrip_pp,1)
p5 = polyfit(secondtrip_pd,secondtrip_pp,1)
p6 = polyfit(kerry_pd,kerry_pp,1)
p7 = polyfit(tipp_pd,tipp_pp,1)
p8 = polyfit(water_pd,water_pp,1)
p9 = polyfit(cork_pd,cork_pp,1)
p10 = polyfit(clare_pd,clare_pp,1)
p11 = polyfit(lim_pd,lim_pp,1)

#Other Main Graphs
#Max LUX Graph
slope,intercept,r_value,p_value,std_err = linregress(
    population_density,max_lux)
correlation = (pow(r_value,2))
scatter(population_density,max_lux)
title("Max LUX Values (All Sites)")
xlabel("Population Density (people per km^2)")
ylabel("LUX Values")
plot(unique(population_density), poly1d(p2)(unique(
    population_density)))
text(0,120, "r = " + str(round(correlation, 3)))
stddev = std(max_lux)
errorbar(population_density, max_lux, stddev, color='g',
    linestyle='None')
show()
#Min LUX Graph
slope,intercept,r_value,p_value,std_err = linregress(
    population_density,min_lux)
correlation = (pow(r_value,2))
scatter(population_density,min_lux)
title("Minimum LUX Values (All Sites)")
xlabel("Population Density (people per km^2)")
ylabel("LUX Values")
plot(unique(population_density), poly1d(p3)(unique(
    population_density)))
text(0,95, "r = " + str(round(correlation, 3)))
stddev = std(min_lux)
errorbar(population_density, min_lux, stddev, color='g',
    linestyle='None')
show()
#Mean LUX Graph (First Trip)
```

```
slope,intercept,r_value,p_value,std_err = linregress(
    firsttrip_pd,firsttrip_pp)
correlation = (pow(r_value,2))
scatter(firsttrip_pd,firsttrip_pp)
title("Mean␣LUX␣Values␣(First␣Trip)")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(firsttrip_pd), poly1d(p4)(unique(firsttrip_pd)))
text(0,100, "r␣=␣" + str(round(correlation, 3)))
stddev = std(firsttrip_pp)
errorbar(firsttrip_pd, firsttrip_pp, stddev, color='g',
    linestyle='None')
show()
#Mean LUX Graph (Second Trip)
slope,intercept,r_value,p_value,std_err = linregress(
    secondtrip_pd,secondtrip_pp)
correlation = (pow(r_value,2))
scatter(secondtrip_pd,secondtrip_pp)
title("Mean␣LUX␣Values␣(Second␣Trip)")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(secondtrip_pd), poly1d(p5)(unique(secondtrip_pd))
    )
text(620,50, "r␣=␣" + str(round(correlation, 3)))
stddev = std(secondtrip_pp)
errorbar(secondtrip_pd, secondtrip_pp, stddev, color='g',
    linestyle='None')
show()

#County Graphs
#Kerry Graph
slope,intercept,r_value,p_value,std_err = linregress(kerry_pd
    ,kerry_pp)
correlation = (pow(r_value,2))
scatter(kerry_pd,kerry_pp)
title("Kerry␣Mean␣LUX␣Values")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(kerry_pd), poly1d(p6)(unique(kerry_pd)))
text(0,50, "r␣=␣" + str(round(correlation, 3)))
stddev = std(kerry_pp)
errorbar(kerry_pd, kerry_pp, stddev, color='g', linestyle='
    None')
show()
#Tipperary Graph
```

```
slope,intercept,r_value,p_value,std_err = linregress(tipp_pd,
    tipp_pp)
correlation = (pow(r_value,2))
scatter(tipp_pd,tipp_pp)
title("Tipperary␣Mean␣LUX␣Values")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(tipp_pd), poly1d(p7)(unique(tipp_pd)))
text(1200,37.5, "r␣=␣" + str(round(correlation, 3)))
stddev = std(tipp_pp)
errorbar(tipp_pd, tipp_pp, stddev, color='g', linestyle='None
    ')
show()
#Waterford Graph
slope,intercept,r_value,p_value,std_err = linregress(water_pd
    ,water_pp)
correlation = (pow(r_value,2))
scatter(water_pd,water_pp)
title("Waterford␣Mean␣LUX␣Values")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(water_pd), poly1d(p8)(unique(water_pd)))
text(725,34, "r␣=␣" + str(round(correlation, 3)))
stddev = std(water_pp)
errorbar(water_pd, water_pp, stddev, color='g', linestyle='
    None')
show()
#Cork Graph
slope,intercept,r_value,p_value,std_err = linregress(cork_pd,
    cork_pp)
correlation = (pow(r_value,2))
scatter(cork_pd,cork_pp)
title("Cork␣Mean␣LUX␣Values")
xlabel("Population␣Density␣(people␣per␣km^2)")
ylabel("LUX␣Values")
plot(unique(cork_pd), poly1d(p9)(unique(cork_pd)))
text(1200,100, "r␣=␣" + str(round(correlation, 3)))
stddev = std(cork_pp)
errorbar(cork_pd, cork_pp, stddev, color='g', linestyle='None
    ')
show()
#Clare Graph
slope,intercept,r_value,p_value,std_err = linregress(clare_pd
    ,clare_pp)
correlation = (pow(r_value,2))
```

```
scatter(clare_pd,clare_pp)
title("Clare Mean LUX Values")
xlabel("Population Density (people per km^2)")
ylabel("LUX Values")
plot(unique(clare_pd), poly1d(p10)(unique(clare_pd)))
text(800,40, "r = " + str(round(correlation, 3)))
stddev = std(clare_pp)
errorbar(clare_pd, clare_pp, stddev, color='g', linestyle='None')
show()
#Limerick Graph
slope,intercept,r_value,p_value,std_err = linregress(lim_pd, lim_pp)
correlation = (pow(r_value,2))
scatter(lim_pd,lim_pp)
title("Limerick Mean LUX Values")
xlabel("Population Density (people per km^2)")
ylabel("LUX Values")
plot(unique(lim_pd), poly1d(p5)(unique(lim_pd)))
text(1300,45, "r = " + str(round(correlation, 3)))
stddev = std(lim_pp)
errorbar(lim_pd, lim_pp, stddev, color='g', linestyle='None')
show()
```

Main Program

```
print """Photopollution Calculator Copyright (C) 2018 Conor Casey
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. For further details, type ‘license'.

Don't know how to use the program? Type "help" for support.
To exit the program, type 'quit'.

Updates:
v0.1.0: Initial beta version based on the mathematical model from the SciFest
project called "How Population Density Influences Light Pollution".
v0.1.1: Allowed population density figures to be inputted.
v0.5.0: The mathematical model is updated, and is now based
```

```
    on the
BT Young Scientist project called "Is it Possible to Create a
     Mathematical Model
to Predict Photopollution Based on Population Density in
    Munster".
v0.5.1: Includes licensing agreement.
v0.5.2: Updated population density figures to include the
    figures from the
2016 Census.
v0.5.3: See those bugs, no? Well, they are gone!
v0.5.4: Slight changes to the mathematical model.
v0.6.0: Slight changes to the grading system and mathematical
     model.
v0.6.1: Fixed TypeError when entering a text value for
    population density.
v0.6.2: See those bugs, no? Well, they are gone!
v0.6.3: Updated to support Windows and macOS alongside Linux.
v1.0.0: Worldwide Public Release on Github!
v1.0.1: Day 1 Bug Patch.
v1.1.0: Let's Get Py Crazy! update.
v1.1.1: Small Changes and Fixes
v1.2.0: Major Changes to Data Calculations
v1.5.0: Fin 2018 update
v1.5.1: Emergency Bug Fix Update
v1.5.2: Small Changes and Fixes
v1.6.0: Errer Update

Current Edition: v1.5.2"""

def main():
        #Imports Population Density Data from the Central
            Statistics Office
        import pandas as pd
        import math
        pd_filename = 'Data/Population_Density/
            populationdensitycensustowns.csv'
        pd_data = pd.read_csv(pd_filename)

        place_town = raw_input("""
Is the name of the town being entered: """).lower()

        #Licensing Agreement
        if place_town == "license":
                license = open('Help␣and␣Licensing␣Agreement/
                    license.txt','r')
                license_read = license.read()
                print license_read
                license.close()
                main()
        #Force Closes Application
        elif place_town == "quit":
                quit()
        elif place_town == "help":
                text = open('Help and Licensing Agreement/
                   help.txt','r')
                text_read = text.read()
                print text_read
                text.close()
                main()

        #Population Density or Town Input
        elif place_town == "no":
                place = raw_input("""
Enter the Population Density: """)
                if place.isdigit() == False:
                        print "It appears you have entered 
                           words instead of numbers, please 
                           try again."
                        main()
                else:
                        print place
                        user_input = float(place)

        #If A Town Is Entered
        elif place_town == "yes":
                towns = raw_input("""
Please input the name of the town: """).title()
                town = pd_data[pd_data.Towns.isin([towns])]
                town.reset_index(inplace = True, drop = True)
                print town
                user_input = float(town.PD)
                if not town.empty:
                        print town
                        user_input = float(town.PD)
                else:
                        print "No population density data for
                            the town of " + towns + " is 
                           available."
                        main()
        else:
                print "Invalid Entry"
                main()

        #Calculation of Light Pollution
        lux = 0.03510566 * user_input - 14.32414198

        #Get Limerick's Population Density In Order to
           Understand what the LUX Values Mean
        limerick = pd_data[pd_data.Towns.isin(["Limerick City
           "])]
        limerick.reset_index(inplace = True, drop = True)
        limerick_pd = float(limerick.PD)
        limerick_lux = 0.03510566 * (limerick_pd) -
           14.32414198

        #Understanding LUX Values
        calculations = (lux) / (limerick_lux) * 100
        def conditions():
                if calculations >= 80:
                        return " Terrible Stargazing
                           Conditions"
                elif calculations >= 60:
                        return " Poor Stargazing Conditions"
                elif calculations >= 40:
                        return " Fair Stargazing Conditions"
                elif calculations >= 20:
                        return " Good Stargazing Conditions"
                elif calculations >= 0:
                        return " Excellent Stargazing
                           Conditions"

        #Result/Output
        if calculations >= 0:
                print """
Photopollution in this location is approximately """ + str(
   int(math.ceil(lux))) + " LUX, this should" + """
correlate to""" + conditions()
        elif calculations < 0:
                print """
Oops, it appears we are getting a negative LUX value. Your
   population density
is extremely low, therefore, this correlates to Excellent
   Stargazing Conditions."""

        #Restarts Program
```

```
    restart = raw_input("""
Do You Want to Restart the Program: """).lower()
    if restart == "yes":
        print """
        Restarting..."""
        main()
    else:
        quit()
main()
```

Last Minute Minor Corrections to the Project Book

These corrections were provided by Professor Brian Espey, professor of Astrophysics at Trinity College Dublin. They are included here because the Project Book was already printed when we recieved the corrections.

Spelling and Grammar Errors

Spelling

- photometer −− > photometre
- providence −− > provience

Grammar

In the Introduction section, the following sentence is used: "Photopollution is different from light pollution in the sense that the former is more of a generalised term that can also deal with radio spectrum pollution meanwhile photopollution deals specifically with light pollution in the visible spectrum." In this the word former, is used incorrectly. The word former is used to refer to an item at the beginning of a list. This sentence means I was referring to photopollution when I was actually trying to refer to light pollution.

Other Language Type Changes

electromagnetic radiation pollution −− > e.m. pollution

Satellites Accuracy over Models Expanded

Another important note to mention is that satellites provide a way to see detail at a relatively small scale as well as (weather permitting!) updates on a daily level, while census data is only available some years after it has been taken, and even then only updated every five years. Also, based on the last two censuses, typically there is a 4% increase in population density throughout the Republic of Ireland every five years.

Comparing to Walker's Law

We stated in the Project Book that the correlation coefficient between photopollution, and Walker's Law is approximately 0.64. A more accurate representation of the previous statement is: "The correlation coefficient between population, and photopollution is approximately 0.64."

Our Mathematical Model

We reduced the number of significant figures in our mathematical model as the precision used suggests an accuracy of better than one part in a million. As suggested by Professor Brian Espey, we reduce it to three significant figures. This is shown in equation 1.

$$I = 0.035d_x - 14.324 \tag{1}$$

Standard Deviation

It is important to note that the limits (plus or minus one standard deviation) can be exceeded 100%-68.2%=31.8% of the time, so this could happen 3 times for every 10 towns (approximately), even if the values are perfectly normally distributed.

The Moon's Possible Influence on Photopollution

The Moon may have a slight influence, but the observations were from pretty bright locations, for the most part, and this would help reduce the effect, especially early in the lunar cycle.

Emails