ACKNOWLEDGMENTS

Thank you Christen McDonough Boone, Kat Abner, and Sibel Solagan from Fund for the Arts for access to the data and permission to do this project. Thank you Ben Hickson for your guidance and troubleshooting skills. Thanks to Alex and Kendra for your support, and LB for the final push.
To Sharon Elaine Pell, for your charm, grace, and kindness. Thank you for the example you set forth. You will always stand as the matriarch of my castle.
# TABLE OF CONTENTS

ACKNOWLEDGMENTS.........................................................................................................................2

DEDICATION........................................................................................................................................3

TABLE OF CONTENTS........................................................................................................................4

LIST OF FIGURES...............................................................................................................................5

LIST OF ABBREVIATIONS......................................................................................................................7

ABSTRACT...........................................................................................................................................8

INTRODUCTION.....................................................................................................................................9

METHODS.........................................................................................................................................13

  Study Area...............................................................................................................................13
  Software..................................................................................................................................14
    Vector Data...........................................................................................................................15
    Other Data............................................................................................................................15
  Step 1.........................................................................................................................................15
  Step 2.........................................................................................................................................17
  Step 3.........................................................................................................................................19
  Step 4.........................................................................................................................................22
  Step 5.........................................................................................................................................23
  Assumptions..............................................................................................................................24

RESULTS AND DISCUSSION..............................................................................................................24

  Step 1.........................................................................................................................................24
  Step 2.........................................................................................................................................26
  Step 3.........................................................................................................................................26
  Step 4.........................................................................................................................................26
  Step 5.........................................................................................................................................27

CONCLUSIONS.................................................................................................................................41

APPENDIX A: Download Links........................................................................................................43

REFERENCES....................................................................................................................................46
LIST OF FIGURES

Figure 1: Location Map .................................................................................................................. 13
Figure 2: Public Schools Map ......................................................................................................... 14
Figure 3: Batch file ......................................................................................................................... 16
Figure 4: Raw Data ......................................................................................................................... 17
Figure 5: Filtered data ..................................................................................................................... 18
Figure 6: Summarized data .............................................................................................................. 19
Figure 7: Default shapefile styles .................................................................................................. 20
Figure 8: Stylized shapefiles .......................................................................................................... 21
Figure 9: Stylized and joined shapefile .......................................................................................... 21
Figure 10: Print Composer view ..................................................................................................... 22
Figure 11: Script view in Windows Command Prompt ...................................................................... 23
Figure 12: Script Error ....................................................................................................................... 25
Figure 13: Bath file path location .................................................................................................... 25
Figure 14: Script for Arts Group Impact map .................................................................................. 28
Figure 15: Arts Group Impact map ................................................................................................ 29
Figure 16: Script for Grade Impact map ........................................................................................ 30
Figure 17: Grade Impact Map ......................................................................................................... 31
Figure 18: Script for Program Type Impact map ............................................................................ 32
Figure 19: Program Type Impact Map ............................................................................................ 33
Figure 20: Script for Schools Impact map ...................................................................................... 34
Figure 21: Schools Impact Map ....................................................................................................... 35
Figure 22: Script for Students Impact map ..................................................................................... 36
Figure 23: Students Impact Map ..................................................................................................... 37
Figure 24: Script for Zip Code Impact map .................................................................................... 38
Figure 25: Zip Code Impact Map........................................................................................................39
Figure 26: Project path location........................................................................................................40
Figure 27: Template path location......................................................................................................40
Figure 28: Map title locations...........................................................................................................41
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV</td>
<td>Comma-separated value</td>
</tr>
<tr>
<td>ESRI</td>
<td>Environmental Systems Research Institute</td>
</tr>
<tr>
<td>FftA</td>
<td>Fund for the Arts</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information Systems</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
</tbody>
</table>
ABSTRACT

Fund for the Arts is one of the oldest arts fund in the country. Since its formation in 1949, the organization has raised over 200 million for the community, which includes Kentucky and Southern Indiana. This Master’s project will focus on one of the organization’s programs entitled 5x5. The goal of 5x5 is to expose elementary school students to five art experiences before they finish the fifth grade. Several years’ worth of data has been compiled, including school names, performance names, performance type, number of students served, and total cost, among others. Using a combination of these parameters, maps will be auto-generated using CSV templates. The auto-generated maps will show a variety of data, including the amount of art funding per zip code, per program type, per grade, per art group, per school, and per student. The maps will serve as visual evidence of the program’s progress and will be shared with Fund for the Arts Board of Directors and CEO, internal staff, as well as other community stakeholders such as community liaisons, participating schools, current and potential donors and the Louisville Metro Council. Fund for the Arts is a nonprofit that does not have access to ESRI products. This Master’s project combines cartography and scripting to create a functioning deliverable using open-source GIS software that enables the organization to auto-generate maps at will and forego the need to request maps from the local university once a year.
INTRODUCTION


No matter how you pronounce it, the capital of Kentucky is nationally known for many contributions. The city produces one-third of all bourbon manufactured in the world [CITATION Lau13 \ l 1033], it hosts the Kentucky Derby, is known as “The Gateway to the South”, is home of the Louisville Slugger, and is the birthplace of the great heavyweight champion, Muhammad Ali. Louisville is also the location of one of the oldest arts fund in the country. Since 1949, Fund for the Arts has raised more than 200 million for Kentucky and Southern Indiana. As a regional nonprofit, Fund for the Arts does not readily have access to GIS product subscriptions or support. Being such, once a year the organization requests maps from the University of Louisville to assist with their reporting. A GIS is a powerful spatial analysis tool but it also serves an equally powerful role as visual interpretation. Maps provide geographic context and are persuasive tools when it comes to sharing a story. The use of GIS should not be exclusive to high-budget organizations, but instead, should be accessible to organizations of all sizes. In the attempt to make GIS more accessible, this Master’s project will use data provided by Fund for the Arts to auto-generate maps using open-source GIS software and scripting. When maps specific to the organization are easily attainable, sharing progress reports and compiling narratives to Louisville Metro Council, current and potential donors, Fund for the Arts CEO and board members, as well as other community stakeholders, becomes only the click of a mouse away.

Fund for the Arts supports several programs benefiting the Kentucky and Southern Indiana region. These programs include 5x5, School’s Out = Art’s In!, NeXt!, Anthem Healthy Living Through the Arts, and Scholastic Arts Awards, among others. This Master’s project will focus on the 5x5 program using 2016-2017 school year data. 5x5 is an every child arts in
education initiative, setting a goal to expose elementary school students (kindergarten to fifth grade) to five art experiences before they finish the fifth grade. These experiences include field trips, in-school residency programs, in-school performances, and in-school workshops. 5x5 focuses on providing art funding to Title 1 schools in Kentucky and Southern Indiana. Title 1 schools are defined as schools eligible for federal funding assistance based on their high percentage of low-income students.

The purpose of this Master’s project is to develop CSV templates using Fund for the Arts 5x5 data and cartographically design map templates to run on a Python developed script that will allow maps to be auto-generated using open-source GIS software. The maps will generate without the need to open any specific GIS software and will display agency funding data related to area schools, program types, grades served, participating students, art group locations, and Zip Codes.

Numerous studies have been conducted over the years that provide a strong, positive correlation between art programs and communities in need. A study published in February of this year from Penn State’s Social Impact of the Arts Project found that low-income neighborhoods with cultural institutions had an increase on exam scores (18%), and a decrease in obesity (5%), child neglect (14%), and serious crime (18%)[ CITATION Dan17 \l 1033 ]. Another study conducted by Harvard professor, Dr. Felton Earls, examined several Chicago neighborhoods in a fifteen-year analysis that resulted in a vast understanding of community conditions and influences. “Collective efficacy” is the term Dr. Earls deemed responsible for greater health and lower crime. Collective efficacy is the “social cohesion among neighbors combined with their willingness to intervene on behalf of the common good...” [ CITATION Rob97 \l 1033 ].
Collective efficacy can be translated into the use of public spaces, and the engagement and participation of community youth [CITATION Tom09 \l 1033].

With a cause worth fighting for, Fund for the Arts faces a challenge of budgeting program and administrative costs with promotion and fundraising efforts. GIS can help nonprofits make strategic decisions through spatial analysis, better interpret data patterns, and showcase the results in a compelling format. Impressive benefits that come with a price.

Providing maps to Fund for the Arts can be accomplished in few ways. The uninspired approach is allowing FftA to continue requesting maps from the University of Louisville. This solution forces Fund for the Arts to be dependent on another organization’s time constrains, budget-concerns, and a rotating cast of students. A second option is to use ESRI’s platform. ESRI is the leading GIS company in the world. Maps can be auto-generated using ArcMap and ArcPy (a Python package specific to Arc products), however access to the ESRI suite requires a subscription and a learning curve. ESRI product subscriptions are costly (hundreds to thousands a year, depending on the services sought). The intent of having the maps auto-generate is for the cartography portion to be complete and to avoid opening any GIS software. It would be a poor use of funds to pay for a subscription if no one on staff knows how to navigate the software and if there would be no need to open the software to create the maps. There is a hesitancy to use any ESRI services, including ArcGIS Online, since access to the software suite disappears after graduation. A third option is for the University of Louisville to host the data and scripts under their ESRI subscription. This option would be time consuming and may not elicit an agreement. Residing across the country from the University of Louisville makes this optional further challenging. Other alternatives were considered such Jupyter Notebooks, R, Leaflet, and Google but ultimately QGIS and Python programming language were the chosen providers.
QGIS is a free and open-source GIS platform. Open-source means the source code is available for modification, collaboration, and contribution. Python is a programming language that operating systems can read to complete tasks and create GIS applications. Python libraries exist to allow specific sets of Python code to be read and run. The decision to use open-source QGIS and Python was a three-part decision. First, to accept the challenge of learning a new GIS software that will diversify my skill set, second, to keep the project cost-free, and third, to grant Fund for the Arts the freedom to generate maps when desired.
METHODS

Study Area

Fund for the Arts is based in Louisville, Kentucky and supports programs throughout Kentucky and Southern Indiana. 5x5 serves kindergarten through fifth grade students within Jefferson County Public Schools, KY; New Albany Floyd County Public Schools, IN; and Clark County, IN Public Schools (Figure 1). Preference is given to Title 1 schools within each school district (Figure 2).

Figure 1: Location map of Fund for the Arts Headquarters and the 5x5 program study area.

Figure 2: Public schools in Kentucky and Southern Indiana that participate in the 5x5 program. (Mercedes McPherson, UA-GIST, Tucson, 2017).

Software

The project used Microsoft Excel 2016; the open-source GIS software QGIS, version 2.18; the open-source IDE software PyScripter, version 2.6.0.0 x86; and the Windows 10 Command Prompt. The project used the following QGIS libraries: qgis.core, qgis.gui, PyQt4.QtCore, and PyQt4.QtXml. All data have a common projection of NAD83 (HARN) / Kentucky North (ftUS), EPSG: 2891.
Vector Data

Interstate shapefiles were obtained from the UA GIST Transportation dataset in July 2017. County and Zip Code data was acquired from the Census website. The Ohio River shapefile came from the Kentucky Community & Technical College System. The Kentucky public schools data came from the Kentucky Geographic Network, and the Indiana public schools data came from the Indiana Geographic Information Council. Road data was obtained from the Kentucky Geographic Network. All data was obtained in October 2017, unless otherwise noted (see Appendix A for link).

Other Data

Fund for the Arts school year funding data was provided by Kat Abner, Impact Officer for Fund for the Arts. The base scripting code was written by Tim Sutton and posted by Gavin Fleming. The script was obtained online from Kartoza (see Appendix A for link). Montserrat was used as the font for all the maps (see Appendix A for link).

This Master’s project is completed in a five-step approach. It begins with Setting the Environment, followed by Data Curation, Map Compilation, Map Template Creation, and finally Script Development.

Step 1 – Setting the Environment

For the maps to auto-generate successfully, a few key components must be installed. First, QGIS 64-bit for Windows should be installed via the downloads page on the official QGIS website (see Appendix A for link). Next, the folder containing the data for the project will be
downloaded and saved locally the new user’s computer. This will be a compressed folder and will contain the CSV templates, map data, data styles, map templates and scripts. Third, an IDE should be downloaded, such as PyScripter. PyScripter can be installed via SourceForge (see Appendix A for link). In lieu of downloading PyScripter, a webpage, such as Code Beautify should be bookmarked (see Appendix A for link). PyScripter and Code Beautify make computer language easily read by humans and can make finding an error or changing data paths much easier than using a text editor. Finally, the environmental variables should be set so the scripts can run outside of the QGIS Python console. Creating a batch file enables a script to run independent from the Python libraries installed with the QGIS software and allows more Python libraries to be used. Typically, a script runs in an IDE but creating a batch file means the script will be run via the Windows Command Prompt based on a text file (Figure 3).

![Grades Map batch file. (Mercedes McPherson, UA-GIST, Tucson, 2017).](image-url)
Step 2 - Data Curation

Organizations collect vast amounts of data regarding their operations and programming.

Fund for the Arts separates data by school year. Collected data for the 5x5 program include school names, school system, program name, program type, arts group name, number of participating students, point of contact name and contact information, voucher amount, bus funding amount, and awarded mileage (Figure 4). Not all the information collected by the organization is pertinent to the mapping project. After obtaining the data, the first step is to curate said data based on what each map will showcase and strip all formatting from the file (Figure 5). As previously identified, this project will highlight six different areas: art groups, schools, program types, grades, students, and Zip Codes.

Figure 4: Raw 5x5 data provided by Fund for the Arts. (Kat Abner, Fund for the Arts, Louisville, 2017).
Once the raw data has been filtered into six different spreadsheets, further thinning of the data is necessary. This is achieved by summing the total funding amounts or total count for each spreadsheet (Figure 6). Each spreadsheet should also account for latitude, longitude and school or arts group address. Once the records have been summed and the additional data added, each spreadsheet gets saved as an Excel and CSV file type. The Excel workbook serves as the working document to make edits to, while the CSV will allow the latitude and longitude to be mapped in QGIS. Both file types are important to the process.

![Table](image)

Figure 5: Filtered FtA data that has been stripped of formatting and shows only one attribute: Zip Codes. (Mercedes McPherson, UA-GIST, Tucson, 2017).
To complete the data curation component, it is necessary to locate shapefiles that will add context to the maps. For the 5x5 program the following shapefiles were used: interstates, counties, rivers, public schools, roads, and Zip Codes. The shapefiles were manipulated to only display data for Kentucky and Indiana. The data curation portion of the process builds the templates that will be used when storing and pulling data to display in QGIS. Each data template will look slightly different depending on the map type.

**Step 3 - Map Compilations**

The data templates will become the foundation for each of the maps. The map compilation piece of the project begins by opening a blank QGIS project and setting the project projection to: NAD83(HARN) / Kentucky North (ftUS), EPSG: 2891. Once the base shapefiles
The interstates, water, counties, and the specific shapefiles (grades, students, program type, arts groups, schools, or Zip Codes) have been added to the map, each layer can be symbolized and saved as a layer style (Figure 7 and 8). This process is repeated until all map types have been created (Figure 9).

Figure 7: Base shapefiles added to QGIS canvas and given default style. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 8: Base shapefiles after they’ve been stylized in QGIS. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Step 4 – Map Template Creation

Map templates speed up the process of creating multi-series maps. Map templates allow a user to save the structure of their map design for use with other map data. In QGIS, the Print Composer functions as a ‘print preview’ or a ‘layout view’ display. Crucial map components include a title, logo, date, legend (if necessary), scale bar, north arrow, and the map canvas itself. In the case of this project, the dates are dynamic, which means anytime the Print Composer is accessed, the current date is stamped on the page. The color scheme used by Fund for the Arts is mimicked in the map templates, and the official Fund for the Arts logo is color matched on each map. The font chosen, Montserrat, is a free alternative to the font used in official Fund for the Arts literature (Figure 10). Once the Print Composer has been set to the user’s desire, it should be saved as a Template.

Figure 10: Print Composer view in QGIS. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Step 5 – Script Development

At this point, the data has been filtered from the raw data provided by the agency to manageable content appropriate for mapping purposes. The data has been compiled and styled in QGIS and the map templates have been laid out and saved. Now a Python script will be written to pull all the pieces together and produce a series of maps. The intent is to have the maps auto-generate without the need to open QGIS. This means the Python console inside QGIS will need to be circumvented. There is a wide variety of Python libraries that need to be accessed, so setting a batch file will allow the scripts to override the default QGIS libraries from being used. The scripts will run using the Windows Command Prompt and have been fitted with a ‘pause’ to keep the window open after the script has run in case an error has occurred and needs to be identified (Figure 11).

Figure 11: Script once it has run in the Windows Command Prompt. (Mercedes McPherson, UAGIST, Tucson, 2017).

23
Assumptions

The following assumptions are made regarding this project. First, that the scripts will be run on a Windows operating system. The scrips were developed using the Windows Command Prompt to avoid opening any GIS software and thus will not run when accessed from another operating system, namely Apple. Another assumption is that the user running the scripts will have a basic comprehension of the Windows operating system or can at the least follow instructions to change the project and template paths in an IDE. Third, the assumption is made that the user will have access to the Microsoft Suite of software to manipulate templates in Excel when changes to the data occur. The last assumption made is that the funding amounts and participating schools and students do not change drastically. If the entire state of Kentucky were to participate and receive funding from the 5x5 program, the amount of data needing to be displayed would increase and cause placement issues. In such a case, the map templates would need to be reexamined and redesigned to ensure the styles and viewing extents are visible.

RESULTS AND DISCUSSION

Step 1 – Setting the Environment

As previously mentioned, the scripts will only run inside the Windows Command Prompt. An error will generate if a script is run inside an IDE (Figure 12). The errors will be based on Python libraries not found inside the QGIS software, which is why it is critical to create batch files to circumvent the QGIS installed Python libraries. A single path location in the batch files will need to be updated once downloaded to the new user’s computer (Figure 13). The path will point to the location that the QGIS software is installed.
Figure 12: IDE script error message. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Step 2 - Data Curation

Challenges facing the data curation portion of the project involve the raw data provided. The raw data itemizes each category based on funding provided. This means for one school there could be seven different listings due to the school receiving funding for seven different programs. The raw data also includes a lot of information that is not pertinent to mapping purposes. It was thus necessary to create master CSV templates that collect only the relevant agency data that will be mapped and accepts only one record with a total funding sum for the category. A master template file was also created using the Kentucky and Indiana public schools shapefile data. The shapefile was compiled by Demetrio Zourarakis with the Kentucky Division of Geographic Information. The data was condensed to only include elementary school data in case Fund for the Arts accepts new participating schools.

Step 3 - Map Compilations

Issues regarding map compilation relate to projections. If the projections are changed in the properties dialog box, instead of using the “Reproject” tool, the data will not cooperate with each other. QGIS cannot display data from a CSV when joining to a polygon. The CSV needs to be converted into a DBF for it to display appropriately. Such was the case for the Zip Code Impact Map.
Step 4 – Map Template Creation

The issues surrounding the map templates also revolve around projections. If all the shapefiles are not projected the same, the map scale will not display properly.
Step 5 – Script Development

The scripts and the maps they generate can be found on the following pages (Figures 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, & 25).
# Script for QGIS auto-map generation based on Fund for the Art 5x5 data
# Import libraries
from qgis.core import QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
from PyQt4.QtCore import QFileInfo
from PyQt4.QtXml import QDOMDocument

# Update the Project path
project_path = r'Z:\909\Staging\FftA_artsGroup.qgs'

# Update the Template path
template_path = r'Z:\909\Staging\Templates\artgroup.qpt'

# Replace the file if it already exists by deleting it first
if os.path.isfile('ArtsGroup_ImpactMap_2016-2017.pdf'):
  os.remove('ArtsGroup_ImpactMap_2016-2017.pdf')
  print "File replaced"

# Load the canvas and project layers
make_pdf()

def make_pdf():
  canvas = QgsMapCanvas()
  QgsProject.instance().read(QFileInfo(project_path))
  bridge = QgsLayerTreeMapCanvasBridge(QgsProject.instance().layerTreeRoot(), canvas)
  bridge.setCanvasLayers()

  # Load the template settings
  template_file = file(template_path)
  template_content = template_file.read()
  template_file.close()
  document = QDOMDocument()
  document.setContent(template_content)

  composition = QgsComposition(canvas.mapSettings())
  # Set the date as dynamic
  substitution_map = {
    'DATE_TIME_START': 'foo',
    'DATE_TIME_END': 'bar'}
  composition.loadFromTemplate(document, substitution_map)

  # Set Item IDs in template (1 = map, 4 = legend)
  map_item = composition.getComposerItemById('1')
  legend_item = composition.getComposerItemById('4')
  legend_item.updateLegend()

  composition.refreshItems()

  # Export map as PDF, set file name
  composition.exportAsPDF('ArtsGroup_ImpactMap_2016-2017.pdf')

  QgsProject.instance().clear()
  make_pdf()
Figure 14: Script for Arts Group Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 15: Generated Arts Group Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
# Script for QGIS auto-map generation based on Fund for the Art 5x5 data

```python
# Import libraries
from qgis.core import QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
from PyQt4.QtCore import QFileInfo
from PyQt4.QtXml import QDomDocument

gui_flag = True
app = QgsApplication(sys.argv, gui_flag)
app.initQgis()

# Update the Project path
project_path = 'Z:\909\Staging\FftA_Grades.qgs'
# Update the Template path
template_path = 'Z:\909\Staging\Templates\grades.qpt'
# Replace the file if it already exists by deleting it first
if os.path.isfile('Grade_ImpactMap_2016-2017.pdf'):
    os.remove('Grade_ImpactMap_2016-2017.pdf')
print "File replaced"

# Load the canvas and project layers

# Load file from project

canvas = QgsMapCanvas()
QgsProject.instance().read(QFileInfo(project_path))
bridge = QgsLayerTreeMapCanvasBridge(QgsProject.instance().layerTreeRoot(), canvas)
bridge.setCanvasLayers()

# Load the template settings
template_file = file(template_path)
template_content = template_file.read()
template_file.close()
document = QDomDocument()
document.setContent(template_content)
composition = QgsComposition(canvas.mapSettings())
# Set the date as dynamic
substitution_map = {
    'DATE_TIME_START': 'foo',
    'DATE_TIME_END': 'bar'}

composition.loadFromTemplate(document, substitution_map)
# Set Item IDs in template (1 = map, 4 = legend)
map_item = composition.getComposerItemById('1')
legend_item = composition.getComposerItemById('4')

legend_item.updateLegend()
composition.refreshItems()

# Export map as PDF, set file name
composition.exportAsPDF('Grade_ImpactMap_2016-2017.pdf')
QgsProject.instance().clear()
make_pdf()
```
Figure 16: Script for Grade Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 17: Generated Grade Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
#Script for QGIS auto-map generation based on Fund for the Art 5x5 data

#Import Libraries

```python
3 import sys, os
4 from qgis.core import QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry
5 from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
6 from PyQt4.QtCore import QFileInfo
7 from PyQt4.QtXml import QDomDocument
8 gui_flag = True
9
10 app = QgsApplication(sys.argv, gui_flag)
11 app.initQgis()
12
13 #Update the Project path
14 project_path = 'Z:\909\Staging\FftA_program.qgs'
15
16 #Update the Template path
17 template_path = 'Z:\909\Staging\Templates\programtype.qpt'
18
19 if os.path.isfile('ProgramType_ImpactMap_2016-2017.pdf'):
20     os.remove('ProgramType_ImpactMap_2016-2017.pdf')
21     print "File replaced"
22
23 #Load the canvas and project layers
24 def make_pdf():
25     canvas = QgsMapCanvas()
26     QgsProject.instance().read(QFileInfo(project_path))
27     bridge = QgsLayerTreeMapCanvasBridge(QgsProject.instance().layerTreeRoot(), canvas)
28     bridge.setCanvasLayers()
29     template_file = file(template_path)
30     template_content = template_file.read()
31     template_file.close()
32     document = QDomDocument()
33     document.setContent(template_content)
34     composition = QgsComposition(canvas.mapSettings())
35     #Set the date as dynamic
36     substitution_map = {
37         'DATE_TIME_START': 'foo',
38         'DATE_TIME_END': 'bar'}
39     composition.loadFromTemplate(document, substitution_map)
40     #Set Item IDs in template (1 = map, 4 = legend)
41     map_item = composition.getComposerItemById('1')
42     legend_item = composition.getComposerItemById('4')
43     legend_item.updateLegend()
44     composition.refreshItems()
45     #Export map as PDF, set file name
46     composition.exportAsPDF('ProgramType_ImpactMap_2016-2017.pdf')
47     QgsProject.instance().clear()
48
49 make_pdf()
```
Figure 18: Script for Program Type Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 19: Generated Program Type Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
# Script for QGIS auto-map generation based on Fund for the Art 5x5 data

```python
# Import libraries
from qgis.core import (QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry)
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
from PyQt4.QtCore import QFileInfo
from PyQt4.QtXml import QDomDocument

gui_flag = True
app = QgsApplication(sys.argv, gui_flag)
app.initQgis()

# Update the Project path
project_path = 'Z:\909\Staging\FftA_school2.qgs'

# Update the Template path
template_path = 'Z:\909\Staging\Templates\schools.qpt'

# Replace the file if it already exists by deleting it first
if os.path.isfile('Schools_ImpactMap_2016-2017.pdf'):
    os.remove('Schools_ImpactMap_2016-2017.pdf')
    print "File replaced"

# Load the canvas and project layers
def make_pdf():
    canvas = QgsMapCanvas()
    QgsProject.instance().read(QFileInfo(project_path))
    bridge = QgsLayerTreeMapCanvasBridge()
    QgsProject.instance().layerTreeRoot(), canvas
    bridge.setCanvasLayers()

    template_file = file(template_path)
    template_content = template_file.read()
    template_file.close()
    document = QDomDocument()
    document.setContent(template_content)
    composition = QgsComposition(canvas.mapSettings())

    # Set the date as dynamic
    substitution_map = {
        'DATE_TIME_START': 'foo',
        'DATE_TIME_END': 'bar'}
    composition.loadFromTemplate(document, substitution_map)

    # Set Item IDs in template (1 = map, 4 = legend)
    map_item = composition.getComposerItemById('1')
    legend_item = composition.getComposerItemById('4')

    legend_item.updateLegend()
    composition.refreshItems()

    # Export map as PDF, set file name
    composition.exportAsPDF('Schools_ImpactMap_2016-2017.pdf')
    QgsProject.instance().clear()

make_pdf()
```
Figure 20: Script for Schools Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 21: Generated Schools Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
# Script for QGIS auto-map generation based on Fund for the Art 5x5 data

```python
# Import libraries
import sys, os
from qgis.core import (QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry)
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
from PyQt4.QtCore import QFileInfo
from PyQt4.QtXml import QDomDocument

# gui_flag = True
app = QgsApplication(sys.argv, gui_flag)
app.initQgis()
# Update the Project path
project_path = '\\999\Staging\FftA_kids.qgs'
# Update the Template path
template_path = '\\999\Staging\Templates\students.qpt'
# Replace the file if it already exists by deleting it first
if os.path.isfile('Student_ImpactMap_2016-2017.pdf'):
    os.remove('Student_ImpactMap_2016-2017.pdf')
print "File replaced"
# Load the canvas and project layers
def make_pdf():
    canvas = QgsMapCanvas()
    QgsProject.instance().read(QFileInfo(project_path))
    bridge = QgsLayerTreeMapCanvasBridge(
        QgsProject.instance().layerTreeRoot(), canvas)
    bridge.setCanvasLayers()
    # load the template settings
    template_file = file(template_path)
    template_content = template_file.read()
    template_file.close()
    document = QDomDocument()
    document.setContent(template_content)
    composition = QgsComposition(canvas.mapSettings())
    # Set the date as dynamic
    substitution_map = {
        'DATE_TIME_START': 'foo',
        'DATE_TIME_END': 'bar'}
    composition.loadFromTemplate(document, substitution_map)
    # Set Item IDs in template (1 = map, 4 = Legend)
    map_item = composition.getComposerItemById('1')
    legend_item = composition.getComposerItemById('4')
    legend_item.updateLegend()
    composition.refreshItems()
    # Export map as PDF, set file name
    composition.exportAsPdf('Student_ImpactMap_2016-2017.pdf')
    QgsProject.instance().clear()
    make_pdf()
```

42
Figure 22: Script for Students Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 23: Generated Students Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
# Script for QGIS auto-map generation based on Fund for the Art 5x5 data

## Import libraries

```python
from qgis.core import (QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry)
```

```python
from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
```

```python
from PyQt4.QtCore import QFileInfo
```

```python
from PyQt4.QtXml import QDomDocument
```

```python
gui_flag = True
```

```python
app = QgsApplication(sys.argv, gui_flag)
```

```python
app.initQgis()
```

## Update the Project path

```python
project_path = 'Z:\909\Staging\FftA_zipcodes-yes.qgs'
```

## Update the Template path

```python
template_path = 'Z:\909\Staging\Templates\zipcodes.qpt'
```

## Replace the file if it already exists by deleting it first

```python
if os.path.isfile('ZipCode_ImpactMap_2016-2017.pdf'):
  os.remove('ZipCode_ImpactMap_2016-2017.pdf')
```

```python
print "File replaced"
```

## Load the canvas and project layers

```python
def make_pdf():
  canvas = QgsMapCanvas()
```

```python
QgsProject.instance().read(QFileInfo(project_path))
```

```python
bridge = QgsLayerTreeMapCanvasBridge(  
  QgsProject.instance().layerTreeRoot(), canvas)
```

```python
bridge.setCanvasLayers()
```

## Load the template settings

```python
template_file = file(template_path)
```

```python
template_content = template_file.read()
```

```python
template_file.close()
```

```python
document = QDomDocument()
```

```python
document.setContent(template_content)
```

```python
composition = QgsComposition(canvas.mapSettings())
```

## Set the date as dynamic

```python
substitution_map = {
  'DATE_TIME_START': 'foo',
  'DATE_TIME_END': 'bar'}
```

```python
composition.loadFromTemplate(document, substitution_map)
```

## Set Item IDs in template (1 = map, 4 = legend)

```python
map_item = composition.getComposerById('1')
```

```python
legend_item = composition.getComposerById('4')
```

```python
legend_item.updateLegend()
```

```python
composition.refreshItems()
```

## Export map as PDF, set file name

```python
composition.exportAsPDF('ZipCode_ImpactMap_2016-2017.pdf')
```

```python
QgsProject.instance().clear()
```

## make pdf()
Figure 24: Script for Zip Code Impact map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Figure 25: Generated Zip Code Impact Map. (Mercedes McPherson, UA-GIST, Tucson, 2017).
Certain elements will need to be altered for the scripts to run locally on a computer other than the programmers. First, the data path will need to updated (Figure 27), as will the template path (Figure 27). The maps will be generated and saved into the same folder as the scripts. The title of each map can be changed in the script (in three locations) as well if so desired (Figure 28). The map canvas will not be drawn to the output file if all the projections are not the same. This applies to data that is present in the Table of Contents but is not displayed. Another issue revolves around reference layers. Using Google Maps or any other reference layer to point check data will skew projections and not allow maps to generate. Finally, graphic files should be stored in the Map Templates as absolute paths rather than relative paths.

```
1 #Script for QGIS auto-map generation based on Fund for the Art 5x5 data
2 #Import libraries
3 import sys, os
4 from qgis.core import (QgsProject, QgsComposition, QgsApplication, QgsProviderRegistry)
5 from qgis.gui import QgsMapCanvas, QgsLayerTreeMapCanvasBridge
6 from PyQt4.QtCore import QFileInfo
7 from PyQt4.QtXml import QDomDocument
8 gui_flag = True
9 app = QgsApplication(sys.argv, gui_flag)
10 app.initQgis()
11 #Update the Project path
12 project_path = 'r'Z:\909\Staging\FftA_artsGroup.qgs'
```

Figure 26: Project path location (line 13). (Mercedes McPherson, UA-GIST, Tucson, 2017).

```
14 #Update the Template path
15 template_path = 'r'Z:\909\Staging\Templates\artgroup.qpt'
```

Figure 27: Template path location (line 15). (Mercedes McPherson, UA-GIST, Tucson, 2017).
CONCLUSIONS

This project developed CSV templates using Fund for the Arts data, to design cartographic map templates that will run on Python developed scripts to auto-generate maps using open-source GIS software. The maps generate without the need to open any specific GIS
software and display Fund for the Arts data related to area schools, program types, grades served, participating students, art group locations, and Zip Codes. The project chose the open-source GIS software, QGIS, and Python programing language as the solution and took a five-step approach, which included Setting the Environment, Data Curation, Map Compilation, Map Template Creation, and Script Development. The result are scripts that run without errors and generate maps locally without the need to open GIS software.

Next steps for this project are to refine a separate script that will convert CVS files to shapefiles, thus avoiding the need to open QGIS when new data is compiled for the forthcoming school years. Second, to follow-up with the organization once they have set their environmental variables for the scripts to run locally on their machines. Finally, to develop a script that will run on Apple operating systems in case of a change in software or agency personnel.
APPENDIX A:

Download Links
Counties shapefile data
https://www.census.gov/geo/maps-data/data/cbf/cbf_counties.html

Ohio River shapefile data
http://techcenter.jefferson.kctcs.edu/gis/data/Country/USA/USA.html

Kentucky public schools shapefile data

Indiana public schools shapefile data
http://maps.indiana.edu/layerGallery.html?category=Schools

Roads shapefile data

Zip Codes shapefile data
https://www.census.gov/geo/maps-data/data/cbf/cbf_zcta.html

Base scripting code

Montserrat font
https://fonts.google.com/specimen/Montserrat

QGIS download
download.qgis.org
PyScripter download

https://sourceforge.net/projects/pyscripter/

Code Beatify

https://codebeautify.org/
REFERENCES


