This is part of my submission for the Data Analytics course. My assigned task was to examine the relationship between church appointments and population in England between 1500 - 1800. The code covers:

1. Creation of a database containing church appointments in England between the years 1500 and 1800 by
   (i) the scraping of webpages from the Clergy Database (https://theclergydatabase.org.uk/),
   (ii) The extraction of data regarding church appointments from these webpages for the location IDs [2,50000] and [235000, 250000],
   (iii) The cleaning and creation of a final dataset from this data.
2. The creation of a balanced panel dataset by merging data on population separated into 25-year panels and the earlier church data.
3. Analysis and prediction using this merged data.

Order in which to run files:

I. scraping_MB_modified.py
II. church.r
III. analysis.r

Part 1: Church appointment data

The input data was scraped from the Clergy Database (linked above). Using a webscraper in Python, I scraped 20193 HTML files from the database. In the document labelled as log, I have compiled the location_id and status code when trying to read the webpage. These files were labelled file(*loc_id)*.html and saved to the Input folder.

These HTML files were then processed (in Python with Beautiful Soup) to extract the information relevant to the project into csv files in the Output folder. This yielded 36563 files in total. These included a data and location files. The data files included data about the people in a particular church entity over time, while the location files were information on each church entity's location. Some files were lost during this processing, and these are specified in sheet 2 of *log.*

The two steps above took place in the file *scraping_MB_modified.py*. The next step imports the csv files into R, using the *church.r* file.

These files were then cleaned and merged. I imported 18192 files of each type (location.csv and data.csv) from the Output folder. These were processed to yield a dataset with 683845 observations. Some summary statistics are included here in the table below:

	vars	n	mean	sd	min	max	range	se
cced_id	1	683845	15224.72	44548.48	1	239503	239502	53.87
Diocese_Jurisdiction	2	683845	NaN	NA	Inf	-Inf	-Inf	NA
Diocese_Geographic	3	683845	NaN	NA	Inf	-Inf	-Inf	NA
Parish	4	683845	NaN	NA	Inf	-Inf	-Inf	NA
County	5	465322	NaN	NA	Inf	-Inf	-Inf	NA
Names	6	683845	NaN	NA	Inf	-Inf	-Inf	NA

	vars	n	mean	sd	min	max	range	se
PersonID	7	683845	77695.40	50314.90	9	177569	177560	60.84
Year	8	683845	1698.20	86.70	1540	1835	295	0.10
Type	9	683845	NaN	NA	Inf	-Inf	-Inf	NA
Full	10	683845	NaN	NA	Inf	-Inf	-Inf	NA
Office	11	683837	NaN	NA	Inf	-Inf	-Inf	NA

Part 2: Merging data to create dataset for analysis

The output of the *church.r* file is final2.csv. Using *analysis.r* I import this file and the provided population data file into R and after some cleaning, merge them and create a balanced panel out of them. This generates the file data2, whose summary statistics are:

vars	N	mean	sd	min	max	range	se
year_panel	102706	1674.97	88.31	1550.00	1850.00	300.00	0.28
C_ID	102706	540.82	298.13	1.00	1106.00	1105.00	0.93
latitude	96499	52.27	1.14	50.10	55.77	5.67	0.00
longitude	96499	-1.16	1.41	-5.56	1.75	7.32	0.00
app_count	102706	523.44	1079.16	0.00	5293.00	5293.00	3.37
Population	96499	1654.25	6185.90	0.00	317095.50	317095.50	19.91
PersonID	89523	81430.23	49720.65	18.00	177569.00	177551.00	166.18

Part 3: Analysis

data2 is then used for all the remaining analysis (continuing to use *analysis.r)*. This analysis include, in this order:

1. Point estimates from an OLS regression of Population on Appointments (app_count is Appointments from the cced, but summed by town and time) for the train and total data. The estimators are then used to predict values for the test data (Figure 1).
2. Point estimates from a FE regression of the same variables, with town and time fixed effects, for train data.
3. The same regressions as above, but with log-transformed values (Figure 2). To account for 0 values, I added a constant to all observations before transforming them.
4. Prediction with random forests prediction algorithm (Figure 3)
5. Prediction with lasso and ridge prediction algorithms (Figure 4)

Figure 1: Predictions of the OLS model (black, dashed line) vs the actual values(blue)

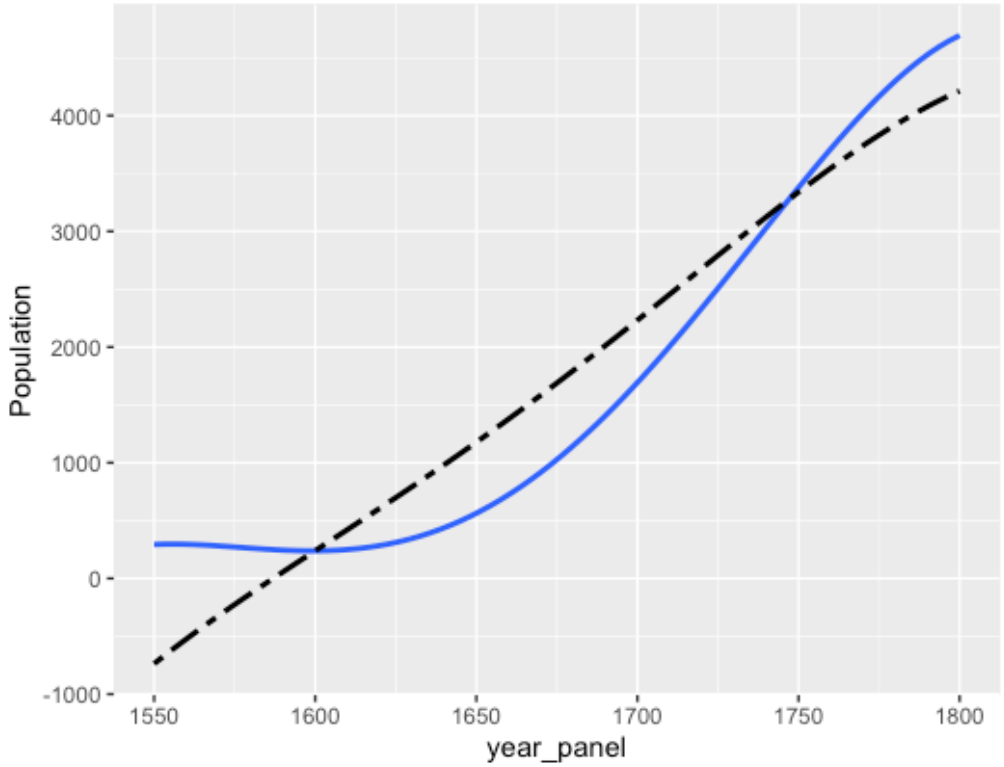

Figure 2: Predictions of the OLS model with log-transformed variables (black, dashed line) vs the actual values(blue)

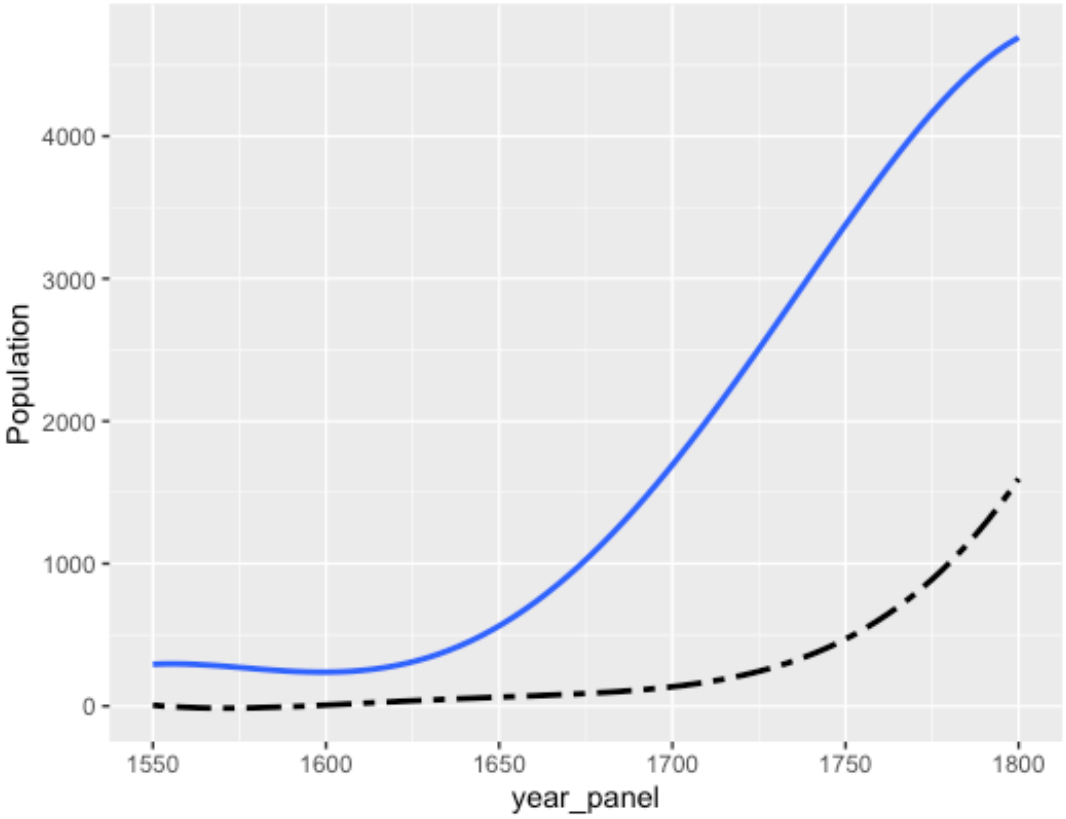

Figure 3: Predictions of the random forest prediction algorithm (black) vs the actual values(blue) and the OLS predictions (black, dashed line)

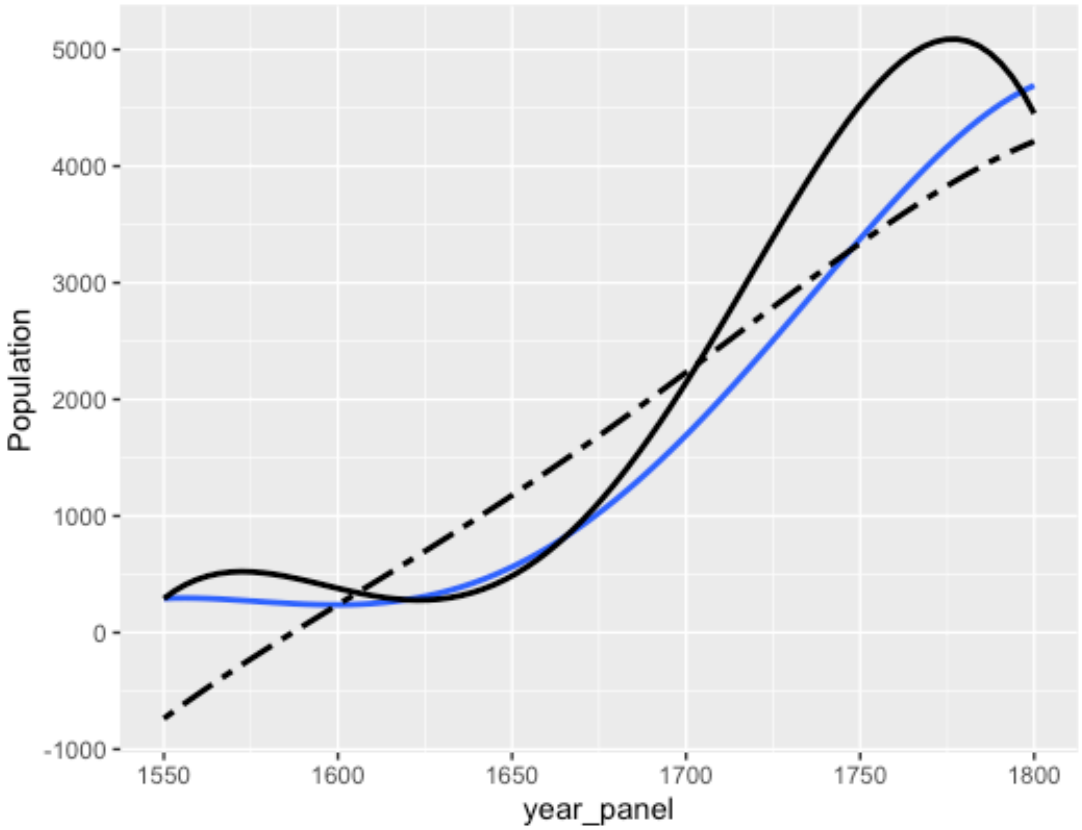

Figure 4: Predictions of the ridge prediction algorithm (red) vs the actual values(blue) and the OLS predictions (black, dashed line)