WRITING SAMPLE

Demographic and Economic Patterns for Opportunity zones in Indiana

Michael Wilcox, Indraneel Kumar, and Santiago Ruiz

Summary

Different politics have aimed to reduce poverty in the United States through public incentives to foster private investment. The Opportunity Zones is one of these initiatives that started on 2018 and that targeted low-income census tracts. The purpose of this research is to shed light on the socio-economic composition of these census tracts designated as Opportunity Zones in the state of Indiana. By using information from the American Census Survey during the last 50 years, we analyzed the trends regarding population growth, economic attainment, racial composition, and industrial diversification. We found that there exists a racial segregation across those areas designated as Opportunity Zones in comparison to the rest of the census tracts. Besides, the difference of educational attainment between people living in those low-income areas has narrowed for low levels of education such as high school. However, there is evidence that shows that people living in opportunity zones struggle to achieve undergraduate and graduate diplomas. As for the economic composition, Opportunity Zones tend to have more establishments designed in the industrial category of entertainment. These findings hope to help policymakers and private enterprises to address their investment into Opportunity Zones by taking advantage of the economic composition of each area.

1. Introduction

The history of Opportunity Zones, from now on designated in this paper as OZs, could be traced back until the president Reagan administration and the Enterprises Zones. The former program looked forward to giving a key role to two forces: free enterprise and the profit motive, in order to substitute the social programs that had failed. In the beginning, the Department of Housing and Urban Development had the power to designate the opportunity zones but not at all to offer tax incentives. It was until 1990 when Reagan's effort saw the light with the Enterprise Zone Tax Incentive act (Weaver, 2016).

This program was designed to boost jobs creation in economically distressed areas through tax cuts. Enterprises Zones spread out across the United States and they vary from state to state regarding duration and type of incentives. Typically, they last between 10-15 years. Furthermore, the tax incentives vary in each state or even county. Tax incentives comprise state loans, property tax credits, income tax credit, investment tax credit, to name several. For instance, Empire Zones in the state of New York or the Pine Tree Development Zones in Maine, which have had a great success in increasing jobs and total investment in these areas (Crawford, 2017).

Among the same type of tools, the New Market Tax Credit (NMTC) was enacted under the Community Renewal Tax Relief Act of 2000. The NMTC permits individuals and corporate taxpayer to receive a credit against federal income taxes for investments in Qualified Community Development Entities (CDEs). CDEs are a domestic corporation or partnership whose primary mission is serving or providing investment capital for low-income communities. They need to be certified by the Community Development Financial Institutions Fund (CDIF). CDEs are required to invest at least 61% of their fund in Low-Income Communities (LIC). For a census tract to be designated as LIC, the poverty rate shall be at least 20 percent, or the median income for such census tracts shall not exceed 80 percent of the statewide family income, or of the metropolitan area if the given census tract is located within one.

Table 1 Tax incentives programs over time

	Enterprises Zones	New Market Tax Credit	Opportunity Zones
Year of creation	1986-1990	2000	2017
Duration	Around 10-15 years	2001-2018	2018-2028
Population targeted	Economically distressed communities	Low-Income Communities	Up to 25% of Low-Income communities and contiguity census tracts under some conditions
Type of Incentives	State loans Property tax credits Income tax credits Investment tax credits	Tax Credits	Tax credits
Geographical extend	Statewide and Countywide	National-wide	National-wide
Process	Vary from state to state	Taxpayers invest in CDIs. After, investments are assessed by the CDFI that allocates them. Once they have been allocated an investment, the taxpayer can ask for a tax credit.	Department of Treasury is still crafting the rules and regulations.
Requirements to get Tax Incentives	Vary from state to state	Investments can only be in cash. The NTMC only can be issued if the investment is done after the CDFI have allocated the investments. Once an investment has been allocated by the CDFI, CDIs shall use it within the first five years.	The capital gains shall be invested within the 180 days after the asset was sold or exchanged

Table 1 was elaborated with information from U.S Code 45D, Investing in Opportunity Act, and press.

Some conditions need to be met for the taxpayer to get NMTC. First, investments in CDEs can only be in cash and they shall be after CDE enters into an allocation agreement with CDIF. Second, once their investments have been allocated by the CDIF, they must be used within the first five years in which such entity receives the allocation. After all the requirements are met, the investors will be entitled to claim NMTC equals to 39% of their investments deferred in 7 years, as follows, 5 percent every year for the first third years and 6 percent every year for the next four years. The CDFI Fund was authorized to allocate to CDEs the authority to issue credits to their investors up to the aggregate amount of $21.5 billion. The NMTC program is still going on. It has

existed since 2001 and there are investments designated until 2019. Additionally, 3.500.000.000 dollars per year is the limit allocation from 2010 to 2019.

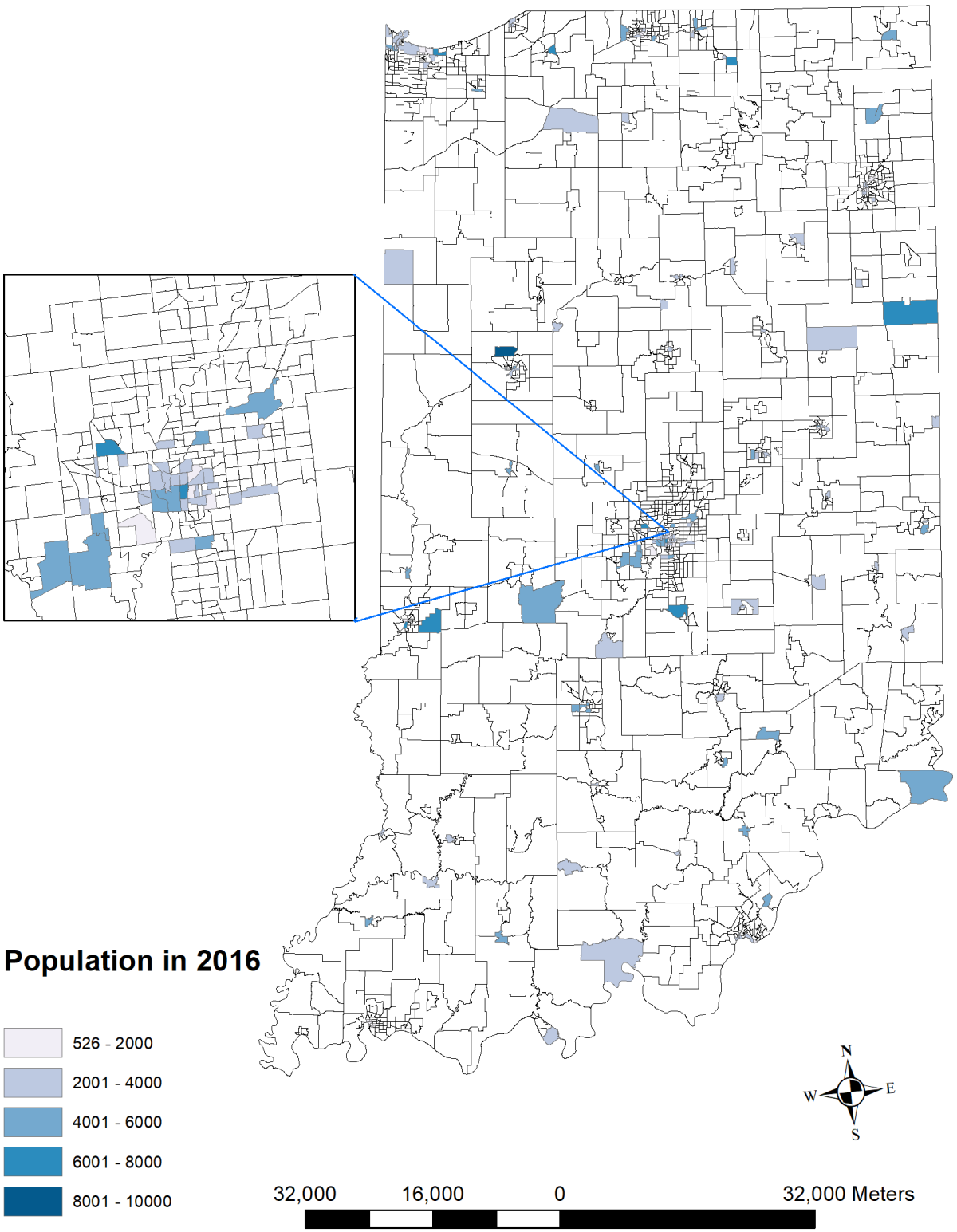
1. The New Tax Incentive

The Tax Cuts and Jobs Act (TCJA) issued by the United States Department of Treasury in 2017 approved tax benefits in form of capital gains for certain areas designated as Opportunities Zones (OZs). In this regard, if profits from a sale are invested (OZs) within the 180 days after the transaction, they will be deferred until 2026. Besides, any gain that arises from an investment in opportunity funds will be excluded from tax if it is held for 10 years. OZs were designated by each governor and certified by the Department of Treasury. They could be either up to 25 % of low-income census tracts or a contiguous census tract under certain circumstances (Rosenthal, 2018). Once selected the OZs will keep this category for 10 years (Theodos, Hedman, Maiexell, & Hangen, 2018)

A capital gain occurs when the sale price you received for an asset is greater than the basis, the former understood as the previous price you paid for it. The TCJA allows investors to increase the basis associated with the investment in the Opportunity Funds (OFs) by an amount equal to 10 % of the amount of the gain deferred when they hold their investments for five years, and an additional 5 % if it is held by two more years. Investors will have to include the gain deferred the earlier of the date when the investment in OZ is sold or in December 2028.

The OFs would be vehicles that provide investors with the tax benefit in three ways. First, by deferring the capital gain taxes during the time investors hold their investments in OZs. Second, by reducing the total amount of taxes paid by increasing the basis of assets, which were sold to be invested in the OFs. Third and finally, if taxpayers hold their investments in the OZs for ten years, they will not pay capital gain taxes for the benefits gotten in the OZs. In other words, the basis of such property shall be equal to the fair market value of such investment on the date the investment is sold. Thus, there will not be capital gains to declare.

Figure 1 Population for OZs in 2016



Note:

The U.S Department of Treasury has not already set completely rules for OZs. However, in comparison with the previous programs, OZs have three new features. First, a taxpayer needs only to reinvest gains from the sale of a given asset. Second, it allows investors to defer as well gains from the sale of assets that are out of the OZ, and third, Investors may organize and market the opportunity funds, which can invest more expansively than earlier programs.

Some authors, such as Rosenthal (2018) point out that the main problem with OZs is that the social benefits of the initiative are unclear. Since there is a disconnection between the size of the potential tax cuts and the social benefits, which could be hard to measure. It could become a way to evade taxes because taxpayers could characterize or restructure existing business arrangements. For these reasons, it may have been better to invest directly in distressed communities.

Among some governmental institutions, concerns raised about unintended consequences of the OZs[1] and its effectiveness to boost economic development. For instance, Displacement and concentration in high-cost cities. As capital seeks the highest returns these funds could be targeted toward areas already in process of gentrifying. Moreover, these investments could concentrate in higher cost areas since people in these places can support higher rents. In addition, there are no restrictions that enforce establishments in these areas to hire local people and consequently it won't have an impact in local's people welfare. It is worth noting that in some cases business ranks their tax incentives low on the list of investments criteria.

2. Database

Our sources come mainly from a spatial harmonized dataset which joins the Panel Study of Income Dynamics (PSID) available from the University of Michigan and the data at the Purdue Center for Regional Development. A spatial harmonized data allows studying locations over time

[1] https://www.stlouisfed.org/on-the-economy/2018/september/opportunity-zones-help-economic-development

since it focuses on adjusting different categories or variables to make them comparable and compatible in different points of time.

Furthermore, some data available from the website American Fact Finder was used to compute some indices ahead, specifically, the American Community Survey. The chosen reference year was the Decennial Census 2010. Shapefile for OZs was taken from the Community Development Financial Institutions Fund (CDFI, 2018) of the U.S. Department of Treasury. Additionally, a set of data came from the National Geographic Information System (NHGIS) (Steve Manson, Jonathan Schroader, David Van Riper, and Steven Ruggles, 2018) .

3. Demographics in OZs

The United States Census Bureau defined a census tract as a "statistically permanent subdivision of a county that is updated by locals. Despite these areas have shifted over time, recent techniques such as spatial data harmonization allow us to analyze trends over these geographical areas. Using the spatial harmonized database of poverty built by PCRD and available through (IPUM, 2018). This review aims to provide stakeholders and economic development corporation with information about the recently established Opportunity Zones (OZ). In 2018, 156 out of 1508 CT in Indiana were designated by the United States Department of Treasury as OZ, which means that around one-ninth of the census tracts in Indiana are OZs.

The population of Indiana has grown evenly since 1970 in most of the census tracts. However, this is barely the case for the OZs, which have climbed far slower than both the rest of the census tracts (CTs) and the entire Indiana state. For simplicity, we will label all census tracts in Indiana that are not OZs as CTs. Whereas population in OZs increased by 2 percent from 1970 to 2016, for CTs it was nearly 29.6 % over the same frame of time. Indeed, in 1970, OZs accounted for around 10 % of the population in Indiana, in 2016 this percentage dropped to 8 %. In terms of area, OZs account for only 3 percent of the land area in Indiana, whereas it accounts for around 8 percent of the entire state. Table 2 summarizes the decennial growth rate of population for both samples from 1970 to 2010, and growth rate from 2010 to 2016. Overall, OZs have undergone a less rate of change than CTs. From 1980 to 1990 OZs grew much faster in population than CTs. It was due mainly to the population growth in cities such as Vincennes, Seymour, Washington, and Huntington.

Table 2 Growth rate of population

Years	Opportunity Zones	Rest of census tracts
1970 - 1980	-2.1%	6.6%
1980 - 1990	7.2%	0.3%
1990 - 2000	-0.3%	10.8%
2000 - 2010	-3.3%	7.6%
2010 - 2016	1.3%	1.7%

Estimated by using the American community survey 2016 5-year estimates and Spatial harmonized Data from PCRD. * Statistically significant at 0.05 by using the Wilcox Rank Sum

The lens of Metropolitan, Micropolitan, and Non-Core statistical subdivision allow analyzing demographic patterns. The U.S. Office of Management and Budget oversees the definition. Metropolitan, Micropolitan, and Non-Core areas are made up of one or more county. When a county has an urbanized area with more than 50.000 inhabitants, it will be designated as Metropolitan. Non-core areas are counties, which do not have an urbanized area with more than 10.000 inhabitants. Finally, Micropolitan areas are between the two former categories. This categorization considers as well the economic integration. If a sizeable share of a county's population commute permanently toward either Metropolitan or Micropolitan areas, this county will be designated as a Metropolitan or Micropolitan area respectively. Despite this classification is not available for census tracts, we used it for the county level and then we matched every census tract with is statistical subdivision. Census tracts in Non-core counties tend to be larger with respect to the land area and thus to have a lower demographic density. It is worth noting that almost 25 % of the area of Indiana is in Non-Core census tract, but this Non-core CT represents only 8.2 % of the total census tracts. That said, 79 % of the OZs are in Metropolitan counties, which could mean that the lion's share of the OZ are in urbanized areas. Namely, 123 out of 156 OZs are in Metropolitan areas, 23 in Micro and only 10 in Non-Core.

As it could be noticed in Figure 2, from 1970 to 2016, OZs climbed slightly in total population, however it was not an upward trend during the former decennials. In fact, in 1980 there was not an increase of population at all, when comparing the median and mean values for the OZ over time. Only in 1990, OZ had a sizeable rise in their population. Nevertheless, since 1990, OZ

have been losing population, even when there are some OZs that have constantly grown, most of them have remained with the same population they had in 1970 or less. At the state level, only 54 % of the CT raised from 1970 to 2016, OZ behaved alike with 52%. In addition, the population in OZs has grown much more slowly and for some census tracts even remained the same. Furthermore, they tend to be in a county where there is an urbanized area and have a far higher density.

Table 3 Number of inhabitants over time

Years	**Opportunity Zones**	**Rest of the census tracts**			
		Average	Median	Average	Median
1970		3353	3354	3454	3184
1980		3282	3350	3682	3491
1990		3518	3473	3695	3521
2000		3508	3369	4093	3921
2010		3393	3119	4404	4042
2016		3437	3157	4477	4075

Estimated by using the American community survey 2016 5-year estimates and Spatial harmonized Data from PCRD. * Statistically significant at 0.05 by using the Wilcox Rank Sum

Table 2 shows the sizeable difference that has been growing up between OZs and CTs. Since 2000, OZs have had progressively less population than CTs.

4. Age Distribution for Opportunity Zones

A population could be split into three mutually exclusive categories: children aged under 15, elderly people aged over 65, and the working-age population. The former population group is made up of people aged between 15 and 65. An analysis of the distribution of various age groups in a population allows understanding the underlying labor market structure. To accomplish this task, we computed the following indices: child dependency ratio, aging index, dependency child ratio and aged dependency ratio (Swanson, 2004), which mathematical formulas could be found in appendix 1. If a given society ages and the fertility rate has been reduced steadily since a long

time ago, working-age population decreases. As this happens, economic activity slows down since it produces a lack of labor force to match the current needs of the market. Regional planners for census tracts that aim to improve their economic status should care about the working-age population in such a way that this population could be able to meet the firm's needs of human capital. These indices were computed for three periods of time, namely 2000, 2010, and 2016.

Table 4 Average age indices over time

		2006-2010	2012-2016
Opportunity zones	Aging Index	75	80
	Dependency Child ratio	31	29
	Aged dependency ratio	19	19
	Dependency ratio	49	48
Rest of census tracts	Aging Index	84	84
	Dependency Child ratio	31	31
	Aged dependency ratio	23	23
	Dependency ratio	54	54
Differences between both samples	Aging Index	-9 *	-4
	Dependency Child ratio	0	-2
	Aged dependency ratio	-4 *	-4 *
	Dependency ratio	-5 *	-6 *

Estimated by using the American community survey 2016 5-year estimates and Spatial harmonized Data from PCRD. * Statistically significant at 0.05 by using the Wilcox Rank Sum

The Aging Index (AI) is a ratio of people aged more than 65 divided by children aged under 15, for a given area, and then it is multiplied by a hundred. This index shows how many elders are for each one hundred children. Put in another words, when the values of AI are high in comparison to another group, it implies an older population. For the ACS of 2006-2010, on average, OZs tended to have a younger population than CTs. However, the gap between the two samples regarding AI has become narrower as time has gone on. In 2012-2016, in OZs there were nearly 80 elders for every one hundred children whereas this number was 75 in 2006-2010. The increasing trend in AI in both OZs and CTs implies an aging process. In fact, on average, the percent of elders aged over 65 in CTs, increase in 1.8 percentage points from 2006-2010 to 2012-2016, as it is shown in table 5, this percentage was 0.5 for OZs, but it was not found statistically significant though.

Table 5 Average percent of age categories

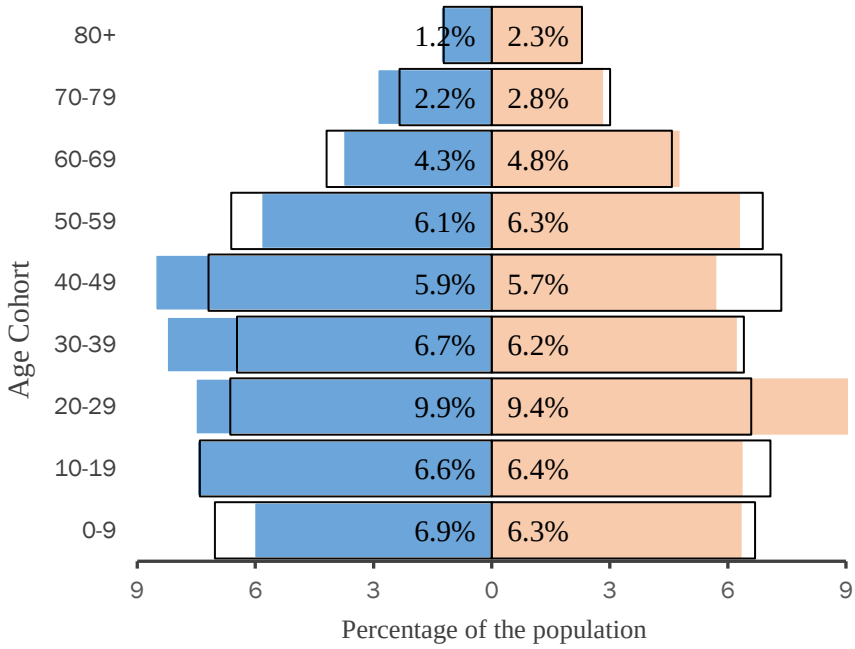
	Opportunity Zones			Rest of the Census Tracts		
	NHGIS 2006-2010	NHGIS 2012-2016	Change	NHGIS 2006-2010	NHGIS 2012-2016	Change
Elderly Population	12.2%	12.6%	+ 0.5%	12.6%	14.4%	+ 1.8%*
Children	20.0%	20.8%	+ 0.8%	20.8%	19.9%	- 0.9%
Working Age	67.8%	66.6%	- 1.3%	66.6%	65.7%	- 0.9%

Estimated by using the American community survey 2016 5-year estimates and Spatial harmonized Data from PCRD. * Statistically significant at 0.05 by using the Wilcox Rank Sum

Both samples have undergone a decrease in the percent of the working-age population. This decrease was greater in OZs than in CTs. Nevertheless, by 2016 OZs have a greater percentage of the population in the working-age. The dependency ratios are computed as the number of children plus the number of elderlies all divided by the number of persons in the working-age. It sheds some light about what the household constitution is. It has barely changed for OZs and CTs since 2006. On average, OZs had between 49 and 48 children or elderlies for every one hundred people in the working-age over time. On the other side, This Index was 54 for CTs. Dependency ratios match with what is shown in table 5 where it could be noticed that the working-age population is a smaller percent in CTs than in OZs for our given frames of time.

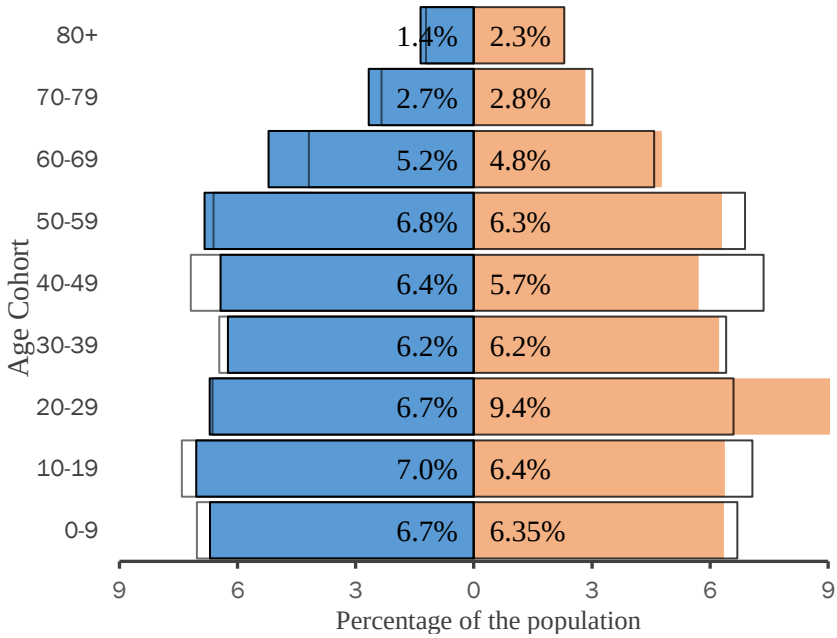
The Aged dependency ratio and the Child dependency ratio did not vary too much overtime either for OZs nor for CTs. The Child Dependency ratio was similar for OZs and CTs. On average, in CTs, there were 31 children for every one hundred people in the working-age whereas for OZs this number ranges between 31 to 29. Using the paired t-test and the Wilcox Rank Sum, we tested the differences of the means overtime for the age indices in table 4 above. It turns out, that there has not been a statistically significant change over time for any of the age indices. However, when comparing OZs and CTs across the same period, we found that these two samples have different age compositions at least when it comes to the percent of elders. In this regard, CTs tended to have more elderly population than OZs. This fact was as well reflected in part through the Aging index. There were 23 elderly people for each one hundred people age between 15 and 65, on average.

Figure 2 Opportunity Zones



Created by using the American community survey 2016-2012, 2006-2010 5-year and estimates and Spatial harmonized Data from PCRD.

Figure 3 Rest of the Census Tracts



Created by using the American community survey 2016-2012, 2006-2010 5-year and estimates and Spatial harmonized Data from PCRD. Sum

Population pyramids are visual representations of the age distribution of the population by gender (PCRD, 2018). These are useful tools to establish which percentage of the population could be found in each age ranges. In figure 3 and 4 are depicted the pyramid populations for OZs and CTs in ACS 5 years 2012-2016, percentages for this frame of time are shown as well: not filled bars represent the ACS 5 years 2006-2010. In the ACS 5 years 2012-2016, around 49 percent of the people in census tracts are men. The greatest percent for an age range in OZs and CTs is people aged between 20-29, this percent is 19.3 for OZs and 16.1 for CTs. As it was discussed before, OZs have a younger population able to work. It means that on average OZs tended to have a greater percent of their population aged between 30-39 and 20-29 in comparison to CTs.

The index of relative difference (Swanson, 2004) is another approach to analysis age distribution. It compares between two percentage age distributions whether for different areas, dates or population subgroups. We compared the OZs and CTs age distribution with the whole United States. Closer to 0 more equal are two age distribution between two different areas. The index for OZs and CTs was 2.5 and 1.5, which means that the age distribution of OZs differs from the United States much more than CTs. As it was suggested above, it is due mainly to the fact that OZs have a greater percentage of its population aged 20-24 and 25 -29 than the United States.

In short, we found that OZs and CTs have different age distributions and these distributions have remained barely without a change since the last decade. In addition, OZs have younger working-age population than CTs. Comparatively, CTs were on average older, which is reflected in a higher Aged dependency ratio and AI.

5. Educational Attainment

In the theory of endogenous economic growth, Human Capital (HC) is strongly related to economic growth and employment. Vijay Mutter defines HC as “an accumulated stock of skills and talent” (1999, p.205). Usually, HC is measured through the number of years of education that a given person earned. A shortcoming of this method is that it does not identify what is the quality of the knowledge acquired. Regardless of the quality, education can be acquired through informal or formal education and by job-training. HC boosts economic development through different channels such as increasing productivity of capital and labor, fostering cluster of firms, and

entrepreneurial activities (Mathur, 1999). Furthermore, as it is explained by Brigitte Waldorf (2007), HC plays an important role in the post-industrial societies, where there is a high demand for professionals that need to be met. In this regard, a highly educated society creates a vibrant business climate.

In the economic literature HC is made up of two components; health and education. There tend to be more information related to education than there is for health, this was the case for the database that we assembled from the National Historic Geographic Information System NHGIS. The importance of educational attainment for economic development has been highlighted over time through several papers. For instance, better-educated societies show lower rates of child mortality across the countries and have a higher life expectancy (Baker, Juan Leon, Greenaway, Collins, & Movit, 2011). Understanding educational attainment of a population gives some insights about which are the main challenges that a society faces before economic development.

The United States Census Bureau measured educational attainment from 1970 to 2016 through four different categories: No High School, High School, Associate's or bachelor's degree, and Master's or Doctoral degree. In each of these categories, educational attainment refers to the highest level of education that an individual aged 25 or over has completed. These categories are reported as a percent of the total population aged 25 or over In the following paragraphs will be described what has been the evolution of the human capital in OZs and CTs for each of the educational categories.

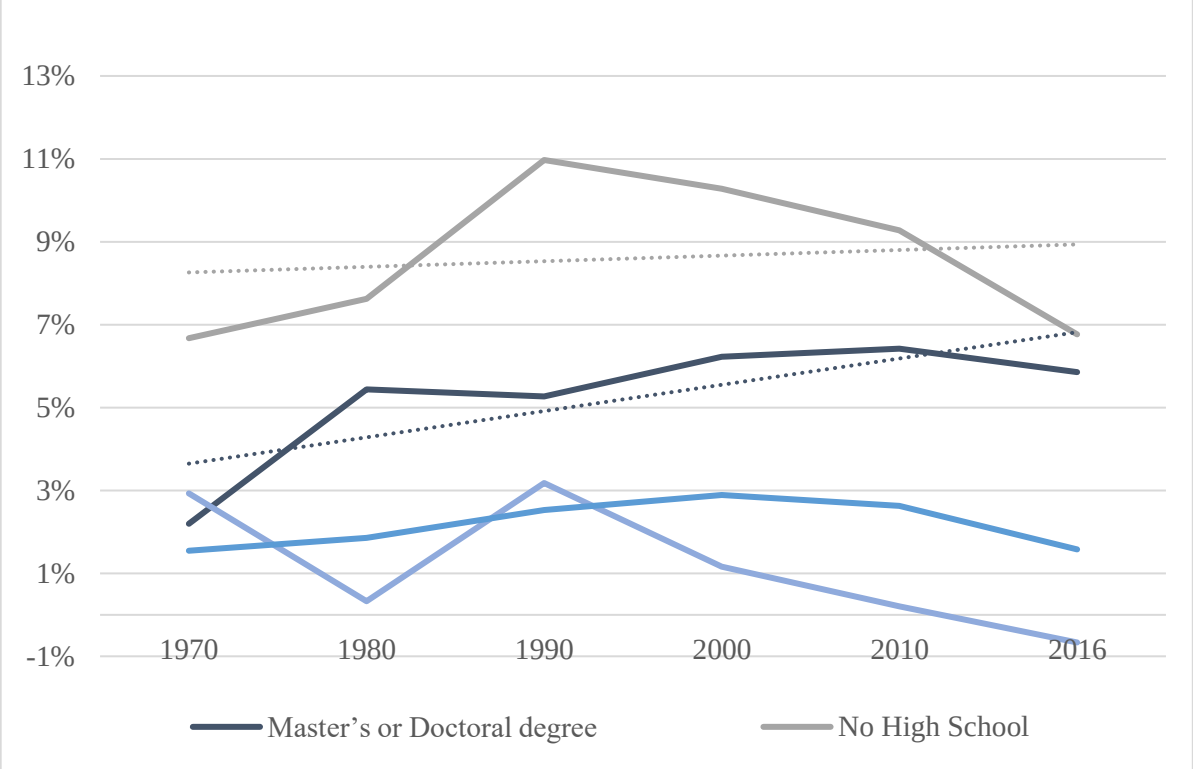
Master's or Doctoral degree

Although the percentage of the population with a master's or Doctoral degree has raised during the last half-century for both groups OZs and CTs, on average OZs tended to have lower percentages of people with Master's or Doctoral degree than other CTs. In 2016 the percent of people in OZs with a higher than college degree was 17% whereas for the rest of the census tracts it was 23%. Figure 5 depicts the gap over time for Master's or Doctoral degree between OZs and CTs. The gap was computed as the difference of the mean percent of the population who owned a Master's or Doctoral degree and aged 25 or older between OZs and CTs. As time has gone on, the gap in this educational category has become wider. It is shown in the dark blue line -Figure 5- below. In 1970, CTs had 2.2 percentage points more population with a higher college degree than OZs, in 2010 this number raised to 6.4 percentage point, which means that CTs have year by year

become a higher educated population than OZs when it comes to degrees higher than college. It is worth noting that there was a change in the trend in the last ACS 5-years in 2016. The gap in this educational category narrowed in 0.6 percentage points in the next six years after 2010.

Some studies have focused on the educational gap between Indiana and the United States. For instance, Waldorf (2005) suggested that the gap between Indiana and the United States percent of adults with at least a 4-years college degree had grown steadily from 1970 to 2004 (p.4). This gap might be due to some census tracts that have been left behind. Specifically, census tracts with high poverty rates such as OZs, which have not been able to catch up with the CTs. For Indiana to enhance its national competitiveness, special attention should be paid to improve the access to college education for people who are living in OZs.

Figure 5 Percentage points gap between OZs and CTs



The graph was made by using U.S. Department of treasury, American community survey 2016 5-year estimates and Spatial harmonized Data from PCRDC. The gap was calculated as the difference between the means of the percent of both samples over time. We found some of the former differences to be statistically significant at 0.05 when computing the ANOVA and Wilcox Rank sum tests, see table 4.

Figure 5- above- shows a time series of the educational gap for each category. It is easy to notice that two educational categories account for most of the differences between OZs and CTs. These are; No High School and Master's or Doctoral degree. Both gaps have had a decreasing trend over time.

Table 4 Percentage point change between CTs and OZs

Year	1970	1980	1990	2000	2010	2016
Master's or Doctoral degree	2.2%	5.4%*	5.3%*	6.2%*	6.4%*	5.9%*
No High School	-6.7%*	-7.6%*	-11.0%*	-10.3%*	-9.3%*	-6.8%*
High School	2.9%	0.3%	3.2%	1.2%	0.2%	-0.7%
Associate's or bachelor's degree	1.5%	1.9%	2.5%	2.9%	2.6%	1.6%

Estimated by using the American community survey 2016 5-year estimates and Spatial harmonized Data from PCRD. Differences were computed as the percent of people in CTs aged 25 or more holding a degree in each category and those living in OZs. Statistically significant at 0.05 by using the Wilcox Rank Sum (*).

Associate's or bachelor's degree and high school

When it comes to bachelor's or associate's degree, there is not so much difference between OZs and the rest of the census tracts over the time. In 2016, the percent of people who had completed a bachelor's or associate's degree in OZs was on average 28 %, whereas it was 29 % for CTs (Table 7). During the frame of the time considered, the gap of the former educational category between these two groups was on average 2.1 % percentage points. Regarding High School, these two groups behaved similarly than the former category. In 2016, the percentage of people with a high school degree in OZs and CTs was 36 % and 35 % respectively. It is worth noting that CTs had a higher percentage of people with a high school degree over our frame of time, on average 1.1 % percentage points more. In fact, 2016 was the first year when OZs had a bigger percent of people who had earned a High School than the rest of CTs in Indiana.

In table 4 are depicted the percentage point for each of the educational categories. In the case of Associate's or Bachelor's degree and High School, we did not find any statistical difference between OZs and CTs. As we will describe below, the difference between OZs and CTs comes when looking at the bottom of the educational spectrum.

No high school

The main difference between OZs and CTs is the percent of people who owe a lower than high school degree, in the last 50 years OZs have had chronically a larger percentage of people in this educational category. In 2016, OZs had 19 % of their population in these range. In contrast, for the rest of the census tracts it was only 13 %. OZs have tended to have more population who have not completed their high school degree, in comparison to CTs. The gap between these two groups was on average over time 8.5 % percentage points, which means that OZs have had 8.5 more of their population without a high-school degree than the rest of the census tracts have.

Table 5 Average of percent of educational attainment over time for the population aged 25 or older

	1970	1980	1990	2000	2010	2016
Opportunity Zones						
Lower Highschool	31%	23%	34%	28%	23%	19%
Highschool	54%	57%	35%	36%	37%	36%
Some bachelor and Associate	8%	11%	20%	20%	25%	28%
Master's or Doctoral degree	7%	9%	11%	16%	15%	17%
Census Tracts						
Lower Highschool	25%	16%	23%	18%	14%	13%
Highschool	57%	57%	38%	37%	37%	35%
Some bachelor and Associate	9%	13%	22%	23%	28%	29%
Master's or Doctoral degree	9%	15%	17%	22%	21%	23%

Estimated with data from U.S. Department of treasury, American community survey 2016 5-year estimates and Spatial harmonized Data from PCRDC.

By large, OZ have more people who do not hold a high-school degree and a lower percent of population with a Master's or Doctoral degree. This could represent an economic burden for enterprises or firms that are looking forward to investing in these areas. They will incur in more costs, because they will have to train the workforce. This might discourage enterprises when setting up in OZs. Further research needs to be done in order to understand what is driven the difference of educational attainment between OZs and CTs.

6. Race and Ethnicity

According to the Census Bureau, a race is *"a person's self-identification with one or more social groups"*, such as White, Black, African American, Asian, American Indian or Alaska Native among others. On the other hand, ethnicity determines whether a person is a Hispanic origin or not. Regarding race, OZs and CTs in Indiana have a different composition when it comes to the percent of black people and whites who are living in these spatial units. OZs had on average a greater black population than CTs. Indeed, the black population was 8 percentage points greater in OZs than in CTs[2]. Regarding ethnic and other racial categories, OZs and CTs have almost the same composition. Additionally, blacks and whites make up 90 % of the population in OZs, whereas they make up 94 % for CTs. This is mainly due to a greater white population in CTs.

Some authors, for instance, Julius (2008), attribute poverty to racial segregation and biased policies that have concentrated black people in inner-city neighborhoods. It reflects the spatial nature of poverty and race. In order to understand how segregation is taking place in Indiana among census tracts, we follow two approaches. First, we computed the Global and Local Moran's I. By doing so, it is possible to portray where the segregation is occurring (Brown & Chung, 2006). Moran's I capture one dimension of the segregation, that is clustering. Second, we estimated the index of dissimilarity and decomposed it into OZs and CTs. The later index considers the evenness of the distribution of races among space. Evenness and clustering are two of the dimensions of segregation mentioned by Massey & Denton (1998, p.283).

Global Moran's I assesses the extent to which a given variable is spatially correlated. It has been widely used to measure clusters through spatial software such as GeoDa or ArcMap. Simply put, it tests whether there are clusters for a given variable or not. However, it does not suggest the location of the cluster. The formula for global Moran's I could be found in appendix 4. This index was estimated by using ACS 5-years in 2016. It turns out that global Moran's I for whites and blacks was 0.50 and 0.61 respectively, which means that census tracts with a high percent of black people tended to be around other areas with a high percent of black people as well. In comparison, the Moran's I for whites was lower. It means that black inhabitants of Indiana are more clustered than white people. Both indices of Moran's I were significant at $p = 0.05$. [3]A relevant finding from

[2] All the percentage points difference mentioned were statistically significant by using both ANOVA and Wilcox rank sum, unless stated otherwise.

[3] Moran's I was computed by using GEODA. We set each statistical test with 99999 permutations.

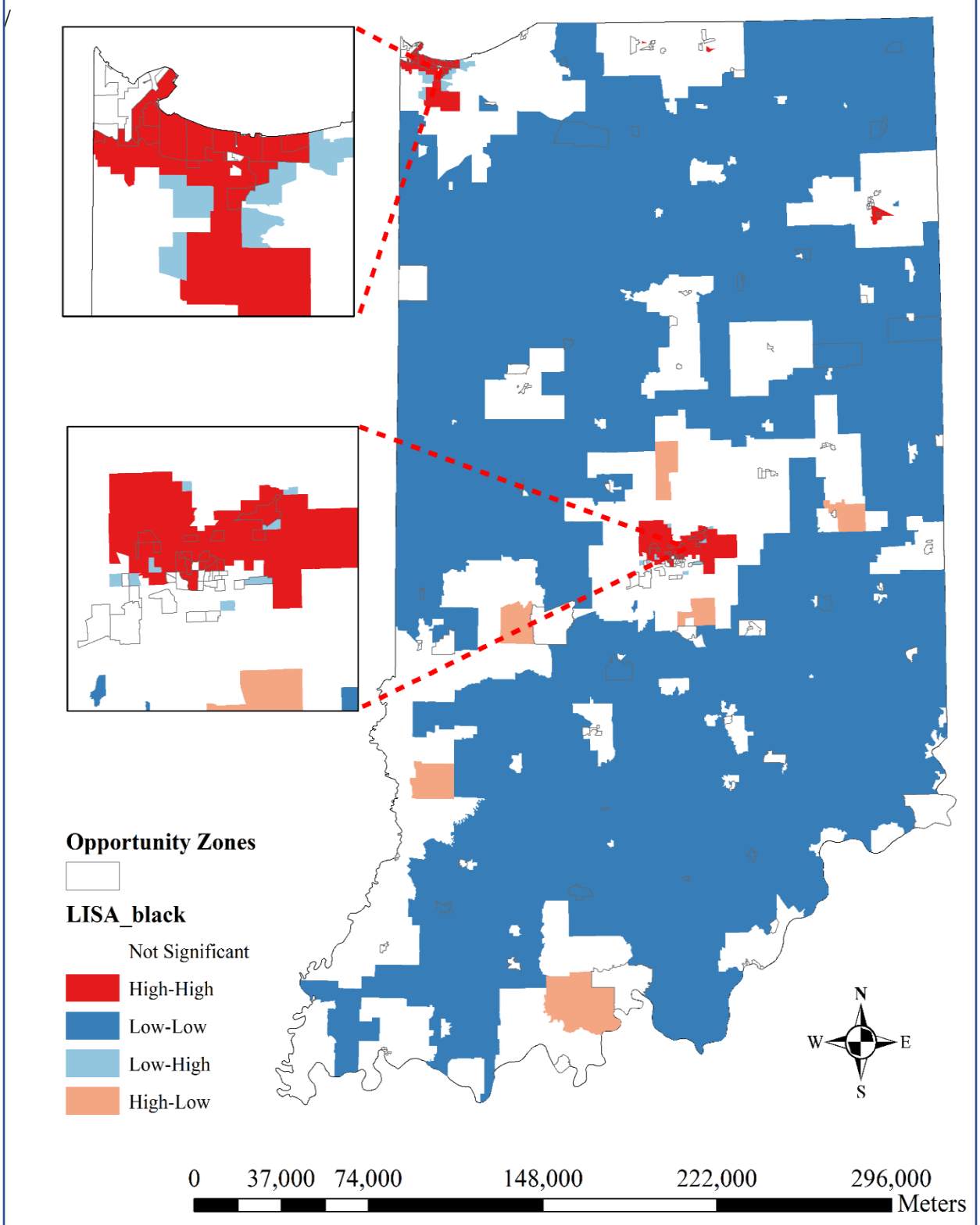
the spatial analysis was given by the bivariate Moran's I between the variables number of population of whites and blacks in OZs. That is a possible segregation between whites and blacks in the counties of Lake, Marion, and Hamilton, where OZs are clustered.

The multivariate Moran's I for blacks and whites was -0.32[4] which means that census tracts with high values of white population tended to be around census tracts whit low black population. A shortcoming of our analysis is that not all OZs share boundaries with other OZs, then these census tracts were not considered when computing the Moran's I. Furthermore, when looking at the county level, segregation between blacks and whites shows up. 2 out of 92 counties in Indiana account for 69 % of the population of black people. That is Marion and Lake, both counties are located close to great cities namely Indianapolis and Chicago.

The global Moran's I does not say much information about where the segregation is taking place. For this reason, we computed the local Moran's I (Anselin, 1995). It portrays the place where the segregation could be occurring. The local Moran's I might be obtained from decomposing the Global Moran's I. Nevertheless, the level of significance needs to be treated carefully[5]. Below, Chart 7 shows the results for the local Moran's I. It shows in red the places were black people might be clustered. As suggested previously, there are two main clusters located around the cities of Indianapolis and Chicago. In this same chart, it is easy to see that a sizeable amount of OZs are located within these clusters. As a result, we claim that there is – at least in some extend- a spatial segregation of black people in Indiana. Given that OZs have on average a greater percent of black inhabitants, and that there has been found a strong relationship between poverty and segregation, OZs need to address this issue. Although it is true that OZs have on average a higher percent of the black population, it does not mean that this population could be explaining the low-income of OZs in itself. In fact, OZs in Indiana are made up mainly by whites. As it is shown in Table 6 below.

[4] It is significant at $p = 0.05$ by using 99999 permutations.
[5] For this reason, we set a lower level of significance, namely $p = 0.01$.

Figure 7 Local Moran's I for black population in 2016 for Indiana



Elaborated with information from the U.S. Department of treasury, American community survey 2016 5-year estimates and Spatial harmonized Data from PCRDC

Table 6 Racial composition in 2016

	Rest of the census tracts	Opportunity Zones	
	Percentage	Percentage	Percentage points difference
White alone	85%	73%	12% *
Black or African American alone	9%	17%	-9% *
American Indian and Alaska Native alone	0%	0%	0%
Asian alone	2%	2%	0%
Native Hawaiian and others Pacific Islander alone	0%	0%	0%
Some other race alone	2%	5%	-3%
Two or more races:	2%	3%	-1%
Two or more races: Two races including Some other race	0%	0%	0%
Two or more races: Two races excluding Some other race, and three or more races	2%	2%	0%
Percent of Indiana population	85%	15%	
Total population of Indiana		6053430	

Created by using the U.S. Department of treasury, American community survey 2016 5-year estimates and Spatial harmonized Data from PCRDC

Several studies have been done regarding the relationship between poverty and racial characteristics. Wilson (2008) explains why the poor inner-cities neighborhoods were a result of an array of political economic issues that turned out being prejudicial, especially for poor communities. For instance, freeways that were built between white and blacks and that became walls that isolate neighborhoods from jobs. Therefore, the spatial distribution of races among Indiana sheds some light about what kind of additional challenges OZs might be facing, in terms of segregation.

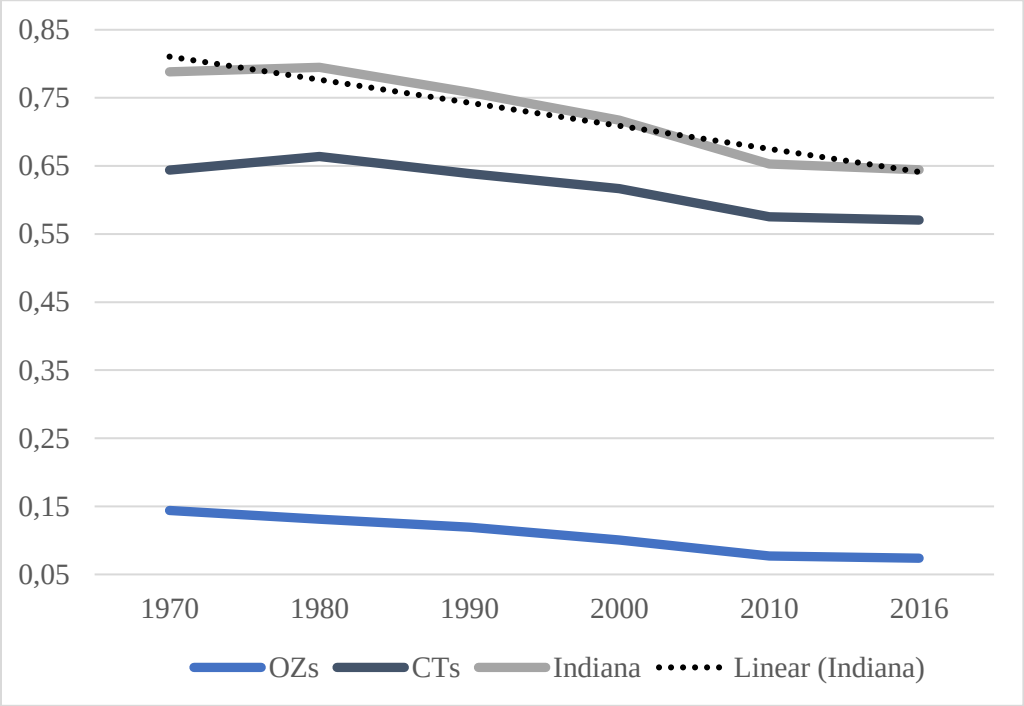
In order to know what is the role of races and ethnicities over the demographic and economic spectrum. We calculated the index of dissimilarity, following one of the dimensions of segregation stated by Dougla S. Massey and Nancy A. Deaton (1988). The former index is a measure of

"the proportion of minority members that would have to change their area of residence to achieve an evener distribution, with the number of minority members moving expressed as

a proportion of the number that would have to move under conditions of maximum segregation" (Massey & Denton, 1988, p. 284).

Put in another way, the index of dissimilarity is a weighted sum of differences between the percent of a minority group in each spacial unit and its given city as a whole, or state in our case. We designated black people as a minority and estimated the former index over time, see the chart 5 above. The formula for the disaggregation index can be found in appendix 3, equation 8. The index ranges between 0 and 1. If it is close to 0, the minority is unevenly distributed over the space. On the other hand, closer to 1, the minority is evener distributed among the space.

Figure 6 Index of dissimilarity for Indiana and its composition according to OZs and CTs



Created by using the American community survey 2016-2012, 2006-2010 5-year and estimates and Spatial harmonized Data from PCRD.

The figure 4 above is depicting the dissimilarity index over time for Indiana and how much of this index was due to OZs and CTs. The index of dissimilarity decreases from 1970 to 2016. Following that, the census tracts over indiana have become more evenly distributed in terms of the racial composition between whites and blacks groups.

The dissimilarity index is a global measure. However, as it is obtained from adding up block groups or spatial units, it can be decomposed (Reardon & Firebaugh, 2002). Consequently, we split the sum according to OZs and CTs, in order to see the effect of each group over the total evenness. Then, we evaluated the value for each spatial unit and estimated the average over time for OZs and CTs. That is how unevenly distributed was the black population in every census tract regarding Indiana. Then, we sorted them among OZs and CTs. Afterward, we got an average participation of each census tract in OZs and CTs for the dissimilarity index. The table 7 shows in some extend the average percent of black people in each census tract that would have to change their area of residence to achieve an evener distribution. It is easy to notice that the percent for OZs was statistically significant higher from 1970 to 2000. Additionally, it's been decreasing to become closer to 0 in 2016, which means that OZs and CTs have become more equally distributed over time, in terms of black people.

Table 7 Average participation of census tracts in the dissimilarity index according to OZs and CTs

	1970	1980	1990	2000	2010	2016
Opportunity Zones	0.09%	0.08%	0.08%	0.06%	0.05%	0.05%
Rest of the Census Tracts	0.05%	0.05%	0.05%	0.05%	0.04%	0.04%
Difference between means over time	0.04%*	0.03%*	0.03%*	0.02%*	0.01%	0.01%

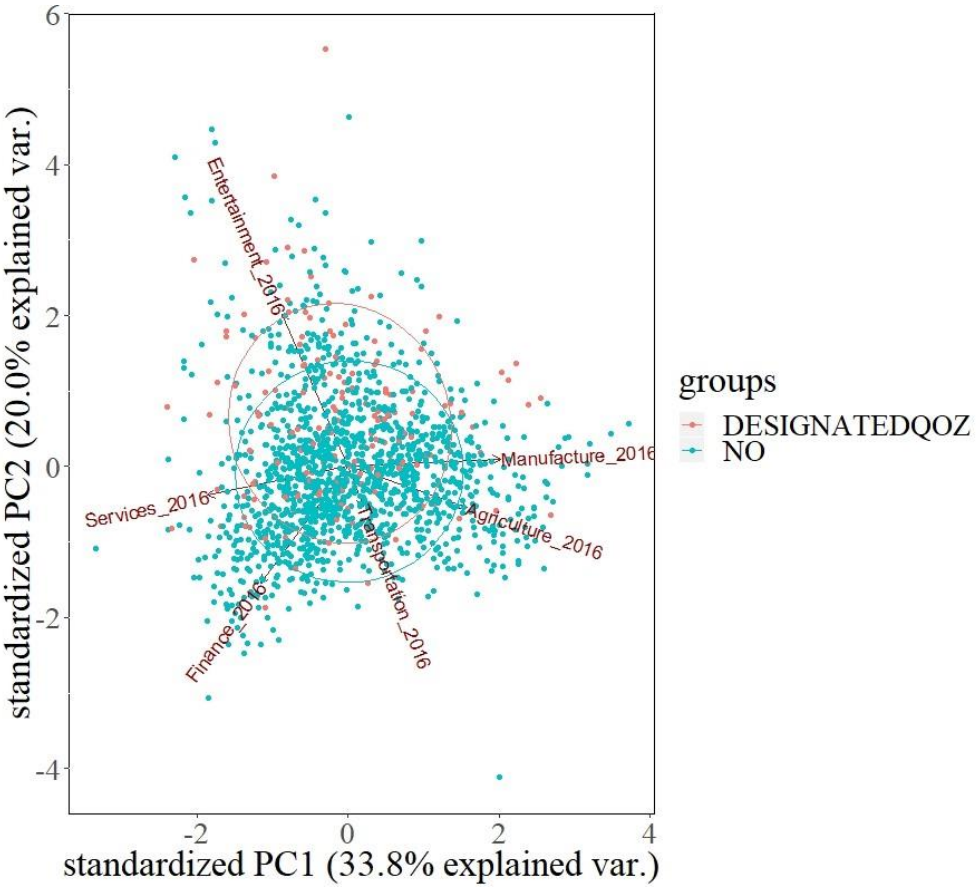
Created by using the U.S. Department of treasury, American community survey 2016 5-year estimates and Spatial harmonized Data from PCRDC. Statistically significant at 0.05 by using the Wilcox Rank Sum (*).

7. Economic composition

As the survey has shifted over time, it is not possible to track each of the NAICS 2 digits variables from1970 to 2016. However, there have been a few economic sectors for which is possible to study their trend. They are manufacturing, transport, finance, agriculture, services, and entertainment. We gathered information regarding the percent of people employed in each of the former sectors for 8 points of time; 1970, 1980, 1990, 2000, 2010, and 2016. Then, in order to understand the economic performance of OZs over time, we performed a Principal Component Analysis (PCA). PCA is a data reduction technique. It is a linear transformation of a dataset and allows to analyze

the variance of observations among a set of variables in a few dimensions. Each principal component has an eigenvalue and an eigenvector. Usually, the first and second component span most of the variance from a dataset. The eigenvalue is the amount of variance explained for a given principal component. Moreover, eigenvectors contain loadings or weights which indicate the amount of variance or correlation explained for each variable in a given principal component.

Figure 8 PCA for the economic composition of OZs and CTs



Estimated by using R's stat package, points in red represent the opportunity zones, and points blue portray rest of the census tracts or NO designated OZs. Information was gathered from Census Bureau, then downloaded from NHGIS, and finally processed by PCRD.

One of the characteristics of the principal components is that they are a set of uncorrelated orthogonal data. In other words, principal components are the directions where is the most variance and therefore the data is more spread out. After computing PCA, each observation is plotted in two dimensions, which correspond with the two first principal component. All the information of the variables concerning a given observation is put into a graph where is possible to observe what is

the location of this observation regarding the direction in which the variance of the variables occurs.

As we will explain later, for most of the economic sectors there is not a significant difference in the economic composition between OZs and CTs over time, regarding mean values for both samples over time[6]. PCA shows that OZs and CTs spread out in almost the same pattern through the scatter plot whose X and Y axis are the first and second component respectively. The two samples for OZs and CTs are overlapped each other, highlighted with blue and red respectively in figure 7. Except for Entertainment, there is not a sizeable difference between both groups when it comes to the sectorial economic composition.

Table 8 Economic composition over time

Opportunity Zones	**1970**	**1980**	**1990**	**2000**	**2010**	**2016**
Agriculture	0.02	0.02	0.01	0.01	0.01	0.01
Entertainment	0.19	0.24	0.24	0.25	0.27	0.12
Finance	0.04	0.04	0.05	0.05	0.05	0.04
Manufacture	0.33	0.29	0.24	0.22	0.17	0.18
Services	0.22	0.23	0.30	0.31	0.34	0.30
Transportation	0.06	0.06	0.06	0.05	0.05	0.05
Rest of the census tracts						
Agriculture	0.04	0.03	0.03	0.02	0.02	0.02
Entertainment	0.19	0.23	0.22	0.22	0.23	0.09
Finance	0.04	0.05	0.06	0.06	0.05	0.05
Manufacture	0.34	0.30	0.25	0.23	0.19	0.19
Services	0.20	0.21	0.28	0.30	0.34	0.30
Transportation	0.06	0.06	0.07	0.05	0.05	0.05
Differences between mean values						
Agriculture	0.01	0.01	0.01	0.01	0.01	0.00
Entertainment	0.00	-0.01	-0.02*	-0.03*	-0.04*	-0.03*
Finance	0.00	0.01	0.01	0.01	0.01	0.01
Manufacture	0.00	0.00	0.01	0.01	0.02*	0.01
Services	-0.02	-0.02	-0.01	0.00	0.00	0.00
Transportation	0.00	0.00	0.00	0.00	0.00	0.00

Economic composition refers to the average percent of employees in the later sectors. Information is from Census Bureau, then downloaded from NHGIS, and finally processed by PCRD.

[6] We performed ANOVA and Wilcox test to test the statistical difference of the means over time. At $p = 0.05$. Only entertainment was statistically different steadily since 1990.

However, there are some limitations to PCA. In this case, the first and second component capture only around 55%[7] of the total variance. It implies that there is another sizeable percent of the variation that is not explained. For this reason, we computed the table 8 above, which shows the changes of mean, variance and median percentage of employees in each economic sector in 2016 for OZs and CTs.

As it was suggested by the PCA, table 8 confirms that there is any difference between the percent in which most of the economic sectors are allocated among the two samples[8]. Whit one important exception, that is Entertainment. The former sector has been statistically more important in OZs than in CTs. This is confirmed by PCA, ANOVA, and the Wilcox test. In Figure 7, the circle that surrounds OZs is closer to the vector that indicates the variable Entertainment. Besides, in table 8, the only statistically significant differences between the means of both samples are the ones of OZs.

Another approach is to arrange census tracts regarding Metropolitan, Micropolitan and Non-core areas. One of the main differences between the statistical categories is in Agriculture. Census tracts located in Non-core areas tended to have a higher economic activity in the agriculture sector than Micro and Metro. However, this gap has narrowed over time. On the other hand, for census tracts in Non-core or Micropolitan areas, both have had consistently over time a greater percent of Manufacturing than Metropolitan areas alone. It is worth noting, that despite the former statistical areas had a bigger manufacturing sector than Metropolitan areas, this sector has dropped continuously since 1970 in the economic composition.

Table 9 Industrial composition over time regarding statistical areas

	Year 1970	**Year 1980**	**Year 1990**	**Year 2000**	**Year 2010**	**Year 2016**
Metropolitan						
Agriculture	2%	2%	2%	1%	1%	1%
Entertainment	20%	24%	23%	24%	24%	10%
Finance	4%	5%	6%	6%	6%	5%

[7] Principal components decay exponentially as it is shown in the scree plot in the Appendix 4. However, the first and second component juts capture around half of the total variance.

[8] Each axe is a principal component. Variables that point out horizontally have more influence with the PC1, similarly to variables pointing out vertically

Manufacturing	33%	28%	23%	20%	17%	16%
Services	21%	22%	30%	32%	35%	32%
Transportation	6%	7%	7%	5%	5%	5%
Micropolitan						
Agriculture	6%	6%	4%	3%	3%	3%
Entertainment	18%	21%	20%	20%	21%	8%
Finance	3%	3%	4%	4%	4%	4%
Manufacturing	37%	35%	33%	33%	27%	27%
Services	19%	19%	25%	26%	30%	26%
Transportation	5%	6%	6%	4%	4%	5%
Non-core						
Agriculture	9%	9%	7%	4%	5%	5%
Entertainment	16%	19%	18%	20%	19%	8%
Finance	2%	3%	4%	3%	4%	4%
Manufacturing	34%	33%	32%	31%	26%	27%
Services	17%	16%	23%	25%	28%	24%
Transportation	5%	5%	6%	5%	6%	5%

Estimated by using Information that was gathered from Census Bureau, then downloaded from NHGIS, and finally processed by PCRD.

Furthermore, services have grown up for every single one statistical area. For instance, in metropolitan areas, the share of this sector grew from 23 % in 1970 to 32 % in 2016 [9]. Transportation and finances did not climb much in any of the statistical areas. These sectors almost remained with the same share over the frame of time. In 2016, the highest share in Metropolitan areas was for the sector of service, while for Non-metro and Micropolitan areas, it was for manufacturing. Another remarkable difference between Non- core and other statistical areas is the sector of entertainment. In every year, except for 2016, Non-core areas had a lower share of the entertainment sector than Metro and Micro areas.

[9] The change over time for was found to be statistically significant by using ANOVA and Wilcox test at $p = 0.05$.

8. Appendix

1. Age Indices

As it is described by (Swanson, 2004) the aging index or the aged child ratio is computed as the ratio of the number of elderly persons over the number of children. It may be represented by the following formula.

$$Aging\ Index = \left(\frac{P^{65+}}{P_{0-14}}\right) * 100 \quad (1)$$

The subscripts refer to the range of ages of each subgroup. For instance, P^{65+} includes every single person who is aged more than 65. Similarly ,the dependency ratio and the child ratio are defined as

$$Dependency\ ratio = \left(\frac{P^{65+} + P_{0-14}}{P_{15-64}}\right) * 100 \quad (2)$$

The dependency ratio is shown in equation 2. Represents the sum of the child population and an aged population divided by the population of intermediate age. Finally, the child dependency ratio is calculated as follows

$$Child\ ratio = \left(\frac{P_{0-14}}{P_{15-64}}\right) * 100 \quad (3)$$

2. Testing statistical differences

In order to test whether there is a difference between two means for independent groups or not, it is important to keep in mind the next assumptions. First, the two populations that are being tested have the same variance. Second, the populations are normally distributed, and finally, each value is sampled independently from each other value. Nevertheless, the fact of violating the two first assumptions, does not change significantly the results.

$$H_0: \mu_{OZ} = \mu_{NDOZ} \quad (4)$$

In our case, ANOVA testes whether the median value for our two samples could be statistically different. This is achieved by estimating an *F-statistic or ratio*:

$$F = \frac{Between\ groups\ variance}{Error\ variance\ within\ groups}, \qquad F = \frac{SS}{W} \quad (5)$$

$$SS = n_{OZ}(\overline{x_{OZ}} - \bar{x}) + n_{NDOZ}(\overline{x_{NDOZ}} - \bar{x}) \quad (6)$$

$$W = (n_{OZ})^2(n_{OZ} - 1) + (n_{NDOZ})^2(n_{NDOZ} - 1) \quad (7)$$

Where n_{OZ} is the size of the smaple for the opportunity zones, namely 156. Besides, $NDOZ$ is the sample for the rest of the census tracts, that is 1352. To obtain a P-Value, the F ration can be tested against a F-distribution.

3. Index of Disaggregation

The formula for the dissimilarity index used by (Massey & Denton, 1988, p. 284) is:

$$D = \sum_{i=i}^{n} \frac{t_i |p_i - P|}{2TP(1-P)} \quad (8)$$

Where p_i is the percent of the minority and t_i is the total population in each aereal unit i, whereas P and T are the percent of the minority and total population respectively.

4. Global Moran's I

$$I = \frac{\sum_i \sum_j w_{ij} z_j \, z_i}{S_0} \frac{N}{\sum_i Z_i} \quad (9)$$

$$S_0 = \sum_i \sum_j w_{ij} \quad (10)$$

Where $z_i = y_i - median_i$, that is deviation from the median. The values depends on the matrix of weights S_0.

References

Anselin, L. (1995). Local indicators of spatial association. *Geography anal*, 93-115.

Baker, D. P., Juan Leon, E. G., Greenaway, S., Collins, J., & Movit, M. (2011). The education effect on population health: a reassesment. *Population development*, 307-322.

Brown, L., & Chung, S.-Y. (2006). Spatial Segregation, Segregation Indices and the Geographical Perspective. *Population, Space and Place*, 125-143.

CDFI. (2018, 9 18). *Opportunity Zones Resources.* Retrieved from https://www.cdfifund.gov/pages/opportunity-zones.aspx

Census Bureau. (2018, 10 8). Retrieved from https://www.census.gov/

Crawford, M. (2017, November 8). Retrieved from AreaDevelopment: www.areadevelopment.com/siteSelection/nov08/enterprise-zones-cost-effective.shtml

Federal Reserve Bank of S.T Louis. (2018, September 24). Retrieved from https://www.stlouisfed.org/on-the-economy/2018/september/opportunity-zones-help-economic-development

Massey, D., & Denton, N. (1988). The Dimensions of Residential Segregation. *Oxford University Press*, 281-315.

Mathur, V. K. (1999). Human Capital-Based Strategy for Regional Economic Development. *Econmic Development Quarterly*, 203-216.

PCRD. (2016). *Spatial Data Documentation.* West Lafayette: Purdue University.

PCRD. (2018). *Purdue Center for Regional Development*. Retrieved from https://www.pcrd.purdue.edu/signature-programs/demographic-profiles.php#Data-Snapshot

Reardon, S., & Firebaugh, G. (2002). Measures of multigroup segregation. *Sociological Methodology, 32*, 33-67.

Rosenthal, S. (2018, June 11). *Opportunity Zones may help investors and sindicators more than distressed communities*. Retrieved from Forbes: https://www.forbes.com/sites/stevenrosenthal/2018/08/20/opportunity-zones-may-help-investors-and-syndicators-more-than-distressed-communities/#5b76d2b876f2

Steve Manson, Jonathan Schroader, David Van Riper, and Steven Ruggles. (2018). *IPUMS National Historical Geographic Information System*. Retrieved from Version 13.0 [Database]. Minneapolis: University of Minnesota. 2018: http://doi.org/10.18128/D050.V13.0

Swanson, J. S. (2004). *The Methods and Materials of Demography.* London: Elsevier Academic press.

Theodos, B., Hedman, C., Maiexell, B., & Hangen, E. (2018, June 11). *Urban Instituts*. Retrieved from https://www.urban.org/policy-centers/metropolitan-housing-and-communities-policy-center

Waldorf, B. (2005). *No County Left Behind? The Persistance of Educational Deprivation in Indiana.* West Lafayette: Purdue Center for Regional Development.

Waldorf, B. S. (2007). Is human capital accumulation a self-propelling process? Comparing educational attainment levels of movers and stayers. *The annals of regional science*, 323-344.

Weaver, T. P. (2016). *Blazing the Neoliberal Trail.* Philadelphia: University of Pennsylvania press.

Wilson, W. J. (2008). The Political and Economical Forces Shaping Cencentrated Poverty. *Political Science Quarterly*, 555-571.

PYTHON CODING EXAMPLE

Johan Santiago RUIZ MORENO

Introduction

This practical work is about the creation of elementary models in keras.

It focuses on image classification, and it illustrates many important functionalities of the keras framework. I relied on Kera's API to create the classses and functions created during this short program.

I was based mostly on the kera's api https://keras.io/api/

But I also used information in this tutos:

- https://blog.keras.io/
- https://machinelearningmastery.com/

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount
=True).
```

In [ ]:
```
from keras.datasets import mnist, fashion_mnist, cifar10
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Model
from keras.layers import Conv2D, Dense, Dropout, MaxPooling2D, Flatten, Input
from tensorflow.keras.utils import to_categorical
import numpy
import os
from tensorflow.keras.optimizers import SGD, Adam
from matplotlib import pyplot as plt
```

In [ ]:
```
import tensorflow as tf
from tensorflow import keras
```

Setup

This notebook was written in Google Colab. So, to start with, I need to upload the datasets into my googledrive.

In [ ]:
```
OUTDIR = "/content/drive/My Drive/Classification"
if not os.path.exists(OUTDIR):
  os.makedirs(OUTDIR)
# First, check dimensions of the datasets !

# shape of images in cifar10: (32, 32, 3)
# shape of images in mnist: (28,28)
# shape of images in fashion_mnist: (28, 28)
```

In [ ]:
```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

In [ ]:
```
(x_train_fas, y_train_fas), (x_test_fas, y_test_fas) = fashion_mnist.load_data()
```

In [ ]:
```
(x_train_cifar, y_train_cifar), (x_test_cifar, y_test_cifar) = cifar10.load_data()
```

In [ ]:
```
#each image is composed of 28 pixels by 28 inside the mnist dataset
print (len(x_train[1]))
print (len(x_train[1][1]))
```

```
28
28
```

In [ ]:
```
# each image is composed of
print (len(x_train_fas[1]))
print (len(x_train_fas[1][1]))
```

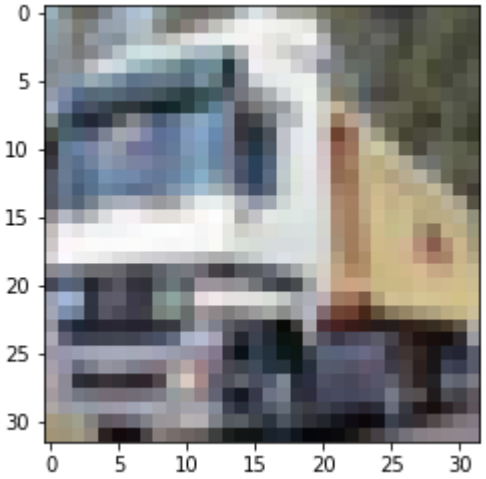
```
28
28
```

In [ ]:
```
# each image is composed of 32 x 32 pixels
print (len(x_train_cifar[1]))
print (len(x_train_cifar[1][1]))
print (len(x_train_cifar[1][1][1]))
```

```
32
32
3
```

In [ ]:
```
# That's how the images look like
%matplotlib inline
import matplotlib.pyplot as plt
plt.imshow(x_train_cifar[1])
```

Out[ ]:
```
<matplotlib.image.AxesImage at 0x7fd07ca57c50>
```



EX1: Lenet-like classifier.

1- Build model architecture

The architecture is a CLASS called "Lenet_like".

- I can define the width: number of filters in the first layer.
- I can define the depth: number of conv layers.

- I can specify the number of classes: output neurons.

Lenet*like has no input tensor. But it has a* _call__ function and can therefore be called on an input later.

The rule to go from depth d to depth d+1, is to reduce the spatial size by a factor of 2 in each direction.

Hidden dense layer will have 512 units.

In [ ]:

```
# first model
# Lenet-like
class Lenet_like:
  """
  Lenet like architecture.
  """
  def __init__(self, width, depth, drop, n_classes):
    """
    Architecture settings.

    Arguments:
      - width: int, first layer number of convolution filters.
      - depth: int, number of convolution layer in the network.
      - drop: float, dropout rate.
      - n_classes: int, number of classes in the dataset.
    """
    self.width = width
    self.depth = depth
    self.drop = drop
    self.n_classes = n_classes

  def __call__(self, X):
    """
    Call classifier layers on the inputs.
    """

    for k in range(self.depth):
      # Appliying the convolutional network and its layers
      # The first parameter refers to the number of filters, the second to the nu
          x = Conv2D(self.width*k, 3, strides=2, padding="same")(X)
          x = Dropout(self.drop)(x)
          x = MaxPooling2D(3, strides=2, padding="same")(x)

    x = Flatten()(x)
    x = Dense(1024, activation='sigmoid')(x)
```

```
    x = Dropout(self.drop)(x)
    x = Flatten()(x)
    x = Dense(512, activation='sigmoid')(x)
    x = Flatten()(x)
    x = Dense(self.n_classes, activation='sigmoid')(x)

    Y = x
    # Perceptron
    # This is the classification head of the classifier
    ...

    return Y
```

2- A function to create a model with Lenet_like architecture

I want the model to be able to fit on the following datasets:

- mnist
- fashion-mnist
- cifar-10

For that purpose, I created a function called "make_lenet_model" that take the name of one of these dataset as a str. It returns a keras Model object. It should obviously take all arguments to init the Lenet_like architecture. Arguments other than the dataset might have default values. I want to be able to monitor the accuracy of the model.

In [ ]:

```
def make_lenet_model(dataset,
                     width=32,
                     depth=3,
                     drop=0.25,
                     n_classes=10):
  """
  Create a Lenet model adapted to the dimensions of a given dataset.
  """
  if dataset == "cifar10":
    # dimensions of input are: (32, 32, 4)
    X = Input(batch_shape=(None,32,32,3))
  elif dataset == "mnist" or dataset == "fashion_mnist":
    # dimensions of input are: (28, 28)
    X = Input(batch_shape=(None,28,28))
  else:
```

```
    raise NotImplementedError("Model not implemented for datastet {}".format(dataset))

  Y = Lenet_like(width, depth, drop, n_classes)(X)

  model = Model(inputs = X, outputs = Y)
  # Remember I wanna monitor accuracy
  model.compile(optimizer= Adam(learning_rate=1e-3), loss=tf.keras.losses.BinaryCrossentropy(), metrics=[tf.keras.metrics
  return model
```

In [ ]:

```
# It does look like a convolutional model
make_lenet_model('cifar10').summary()
```

```
Model: "model_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_30 (InputLayer)       [(None, 32, 32, 3)]       0

 conv2d_55 (Conv2D)          (None, 16, 16, 64)        1792

 dropout_57 (Dropout)        (None, 16, 16, 64)        0

 max_pooling2d_41 (MaxPoolin  (None, 8, 8, 64)         0
 g2D)

 flatten_20 (Flatten)        (None, 4096)              0

 dense_48 (Dense)            (None, 1024)              4195328

 dropout_58 (Dropout)        (None, 1024)              0

 flatten_21 (Flatten)        (None, 1024)              0

 dense_49 (Dense)            (None, 512)               524800

 flatten_22 (Flatten)        (None, 512)               0

 dense_50 (Dense)            (None, 10)                5130

=================================================================
Total params: 4,727,050
Trainable params: 4,727,050
Non-trainable params: 0
_________________________________________________________________
```

3- Create a fitting function

I made a function "fit_model_on" with the following arguments:

- dataset: str name of the dataset
- epochs: number of times you fit on the entire training set
- batch_size: number of images to average gradient on

The function must create a Lenet model and fit it following these parameters. The function should:

- fit the model, obviously
- store the model architecture in a .json file in your output directory
- store the model's weights in a .h5 file in your output directory
- store the fitting metrics loss, validation_loss, accuracy, validation_accuracy under the form of a plot exported in a png file.

In [ ]:

```
def fit_model_on(dataset,
                 epochs=100,
                 batch_size_m=32,
                 n_classes=10):

  model_filename = "lenet_{}.json".format(dataset)
  weight_filename = "lenet_{}_weights.h5".format(dataset)
  lossplot_filename = "lenet_{}_loss.png".format(dataset)
  accplot_filename = "lenet_{}_accuracy.png".format(dataset)

  # create your model and call it on your dataset
  model = make_lenet_model(dataset,
                    width=32,
                    depth=2,
                    drop=0.25,
                    n_classes=10)

  # create a Keras ImageDataGenerator to handle your dataset
  datagen = ImageDataGenerator(
    rotation_range=20,
    rescale=1.0/255.0,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    validation_split=0.2)
```

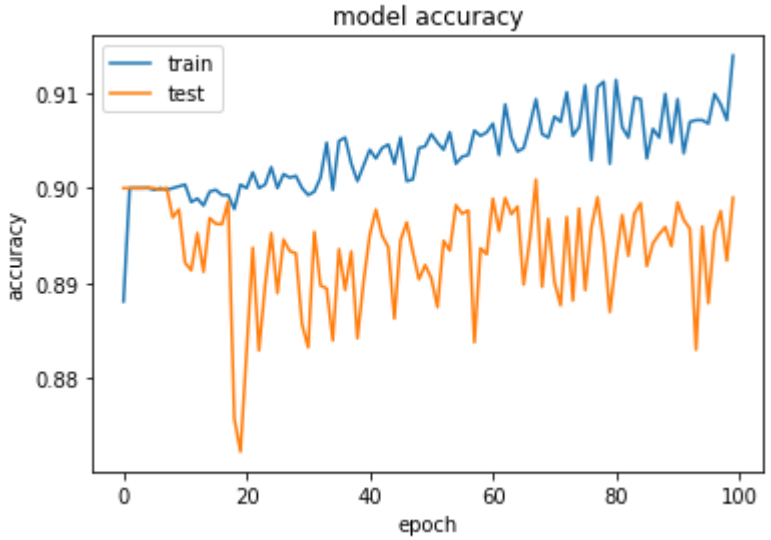
```
  if dataset == "cifar10":
    (x_train, y_train), (x_test, y_test) = cifar10.load_data()
  elif dataset == "mnist":
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
  elif dataset == "fashion_mnist":
    (x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
  else:
    raise NotImplementedError("Model not implemented for datastet {}".format(dataset))

  # Convert class vectors to binary class matrices (one-hot encoding).
  y_train = to_categorical(y_train, n_classes)
  y_test = to_categorical(y_test, n_classes)

  # Be sure that your training/test data is 'float32'
  x_train = x_train.astype("float32")
  x_test = x_test.astype("float32")



  try:
    # Fit with keras using 'datagen', the previously defined image generator
       history = model.fit(datagen.flow(x_train, y_train, batch_size=32,
         subset='training'),
         validation_data=datagen.flow(x_train, y_train,
         batch_size=batch_size_m, subset='validation'),
         steps_per_epoch=len(x_train) / batch_size_m, epochs=epochs)

  except KeyboardInterrupt:
    print("Training interrupted!")

  # first, save the model
  json_str = model.to_json()
  model_path = os.path.join(OUTDIR, model_filename)
  weight_path = os.path.join(OUTDIR, weight_filename)
  with open(model_path, "w") as txtfile:
    txtfile.write(json_str)

  # then, save the weights
  model.save_weights(weight_path)

  # finally, plot and save the metrics
  print(history.history.keys())
# summarize history for accuracy
```

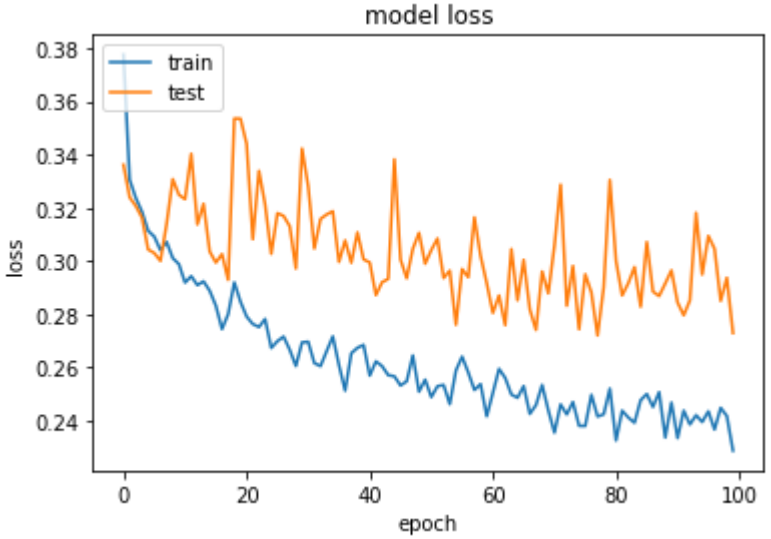
```
  plt.plot(history.history['binary_accuracy'])
  plt.plot(history.history['val_binary_accuracy'])
  plt.title('model accuracy')
  plt.ylabel('accuracy')
  plt.xlabel('epoch')
  plt.legend(['train', 'test'], loc='upper left')
  plt.show()
# summarize history for loss
  plt.plot(history.history['loss'])
  plt.plot(history.history['val_loss'])
  plt.title('model loss')
  plt.ylabel('loss')
  plt.xlabel('epoch')
  plt.legend(['train', 'test'], loc='upper left')
  plt.show()
```

In [ ]:

```
# fit your model
# IT is important to choose apropriater activating functions depending on the problem we are dealing with
fit_model_on("cifar10", epochs=100, batch_size_m=100*30, n_classes=10)
```

```
Epoch 91/100
16/16 [==============================] - 6s 400ms/step - loss: 0.2333 - binary_accuracy: 0.9094 - false_negatives_21: 42
9.0000 - val_loss: 0.2843 - val_binary_accuracy: 0.8985 - val_false_negatives_21: 8273.0000
Epoch 92/100
16/16 [==============================] - 6s 406ms/step - loss: 0.2436 - binary_accuracy: 0.9037 - false_negatives_21: 45
1.0000 - val_loss: 0.2795 - val_binary_accuracy: 0.8967 - val_false_negatives_21: 8278.0000
Epoch 93/100
16/16 [==============================] - 6s 404ms/step - loss: 0.2384 - binary_accuracy: 0.9070 - false_negatives_21: 44
3.0000 - val_loss: 0.2851 - val_binary_accuracy: 0.8957 - val_false_negatives_21: 8246.0000
Epoch 94/100
16/16 [==============================] - 6s 406ms/step - loss: 0.2419 - binary_accuracy: 0.9072 - false_negatives_21: 44
8.0000 - val_loss: 0.3181 - val_binary_accuracy: 0.8830 - val_false_negatives_21: 8309.0000
Epoch 95/100
16/16 [==============================] - 6s 402ms/step - loss: 0.2394 - binary_accuracy: 0.9072 - false_negatives_21: 44
7.0000 - val_loss: 0.2949 - val_binary_accuracy: 0.8960 - val_false_negatives_21: 8187.0000
Epoch 96/100
16/16 [==============================] - 6s 403ms/step - loss: 0.2434 - binary_accuracy: 0.9068 - false_negatives_21: 45
2.0000 - val_loss: 0.3094 - val_binary_accuracy: 0.8879 - val_false_negatives_21: 8417.0000
Epoch 97/100
16/16 [==============================] - 6s 401ms/step - loss: 0.2365 - binary_accuracy: 0.9099 - false_negatives_21: 43
1.0000 - val_loss: 0.3046 - val_binary_accuracy: 0.8954 - val_false_negatives_21: 8360.0000
Epoch 98/100
16/16 [==============================] - 6s 402ms/step - loss: 0.2447 - binary_accuracy: 0.9088 - false_negatives_21: 43
3.0000 - val_loss: 0.2850 - val_binary_accuracy: 0.8976 - val_false_negatives_21: 8263.0000
Epoch 99/100
16/16 [==============================] - 6s 400ms/step - loss: 0.2416 - binary_accuracy: 0.9072 - false_negatives_21: 45
3.0000 - val_loss: 0.2937 - val_binary_accuracy: 0.8924 - val_false_negatives_21: 8171.0000
Epoch 100/100
16/16 [==============================] - 6s 398ms/step - loss: 0.2286 - binary_accuracy: 0.9140 - false_negatives_21: 42
2.0000 - val_loss: 0.2729 - val_binary_accuracy: 0.8990 - val_false_negatives_21: 8038.0000
dict_keys(['loss', 'binary_accuracy', 'false_negatives_21', 'val_loss', 'val_binary_accuracy', 'val_false_negatives_21'])
```


model accuracy
train
test
accuracy
epoch
0.91
0.90
0.89
0.88
0
20
40
60
80
100


model loss
train
test
loss
epoch
0.38
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0
20
40
60
80
100

STATA CODING EXAMPLE

```
/***********************************************
************************************************

 Johan Santiago RUIZ

 This code was written for the course of
 Program Evaluation during the first year of my Master
 in Statistics and Econometrics at the Toulouse School
 of Economics.

 It aimed to reply the results obtained by Brooks,
 Donovan and Johnson (2018) in the paper "Mentors or Teachers?
 Microenterprise Training in Kenya"

***********************************************
***********************************************/


*---------
ssc install cmogram
ssc install rd
ssc install binscatter
ssc install rdrobust
ssc install estout
*---------


clear all

/* Change the working directory*/

cd ""


use "rct_kenya.dta", clear



// Setting the dataset as panel data

xtset id wave



//generating the treatement variable
generate treatb = 2 if control_b == 1
replace treatb = 4 if treat_class_b == 1
replace treatb = 3 if treat_mentor_b == 1


// Analyzing the variability of the control and treatment groups before the experiment

reg businessage  treat_class_b treat_mentor_b if wave == 0
reg sec0_b treat_class_b treat_mentor_b if wave == 0
reg sec1_b treat_class_b treat_mentor_b if wave == 0
reg sec2_b  treat_class_b treat_mentor_b if wave == 0
reg sec3_b  treat_class_b treat_mentor_b if wave == 0
reg sec4_b treat_class_b treat_mentor_b if wave == 0
reg secondaryedu_b treat_class_b treat_mentor_b if wave == 0
reg formalaccount_b treat_class_b treat_mentor_b if wave == 0
reg bankaccount_b treat_class_b treat_mentor_b if wave == 0
reg loan_b treat_class_b treat_mentor_b if wave == 0
reg I_emp_b treat_class_b treat_mentor_b if wave == 0
reg emp_b treat_class_b treat_mentor_b if wave == 0
```

```
reg tprofits_b treat_class_b treat_mentor_b if wave == 0


generate treat2b = 2 if control_b ==1
replace treat2b = 4 if treat_class_b == 1 | treat_mentor_b == 1

// Generating tables and carrying out t-tests to check whether the control and treatment
groups are balanced

estpost ttest tprofits_b businessage sec0_b sec1_b sec2_b sec3_b sec4_b secondaryedu_b
formalaccount_b bankaccount_b loan_b I_emp_b emp_b, by(treat2b)

eststo est4

// Creating lateX tables
esttab est4 using reg_table_1nd.tex, replace cells("mu_1 mu_2 b p count") ///
title("Balancing tests in mentee's selection") unstack r2  ///
label eqlabels(none) se



// Distribution of the microentreprises's profits at the beginning of the program
// and during the waves for the control and treatement groups.


use "rct_kenya.dta", clear
twoway(kdensity tprofits if control_b == 1 & wave == 0)/*
*/(kdensity tprofits if treat_class_b == 1 & wave == 0,lpattern(--))/*
*/(kdensity tprofits if treat_mentor_b == 1 & wave == 0)

// distribution on the wave 3
twoway(kdensity tprofits if control_b == 1 & wave == 3)/*
*/(kdensity tprofits if treat_class_b == 1 & wave == 3,lpattern(--))/*
*/(kdensity tprofits if treat_mentor_b == 1 & wave == 3),

//distribution on the wave 6
twoway(kdensity tprofits if control_b == 1 & wave == 6)/*
*/(kdensity tprofits if treat_class_b == 1 & wave == 6,lpattern(--))/*
*/(kdensity tprofits if treat_mentor_b == 1 & wave == 6),




 // add controls, and clustering across individuals

reg tprofits i.wave treat_class_f treat_mentor_f tprofits_b secondaryedu_b lage_b sec0_b
sec1_b sec2_b sec3_b sec4_b I_emp_b, vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]


// estimating with robust errors for each wave of the program

reg tprofits treat_class_f treat_mentor_f  secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 1, vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 2, vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 3 ,vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
```

```
sec3_b sec4_b I_emp_b tprofits_b if wave == 4  ,vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 5 ,vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 6 ,vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]

reg tprofits treat_class_f treat_mentor_f secondaryedu_b lage_b sec0_b sec1_b sec2_b
sec3_b sec4_b I_emp_b tprofits_b if wave == 7 ,vce(cluster id)
test _b[treat_class_f] = _b[treat_mentor_f]


/** Estimation of the effect of treatment on selected outcome variables  **/


// Generating a second treatment dummmy variable

generate treat = 2 if control_f == 1
replace treat = 4 if treat_class_f == 1
replace treat = 3 if treat_mentor_f == 1

/**checking the change on the variable supplier_switch in th fifth wave for treatment
and control group**/

probit supplierswitch i.treat if wave==5, cluster(id)
margins, dydx(*)

estimates store probit


/* Estimation of the effect of treatment on keeps_some_records*/

probit keeps_some_records i.treat if wave==6, cluster(id)
margins, dydx(*)

estimates store probit


*-------------
*-- installing additional required packages
ssc install psmatch2
*------------


// REGRESSION DISCONTINUITY design and test

/* Balance test for the control variables*/


save "firstdataset.dta", replace
clear
use "rd_mentors.dta",clear

// T-tests and tables for testing the balance on the control and treatment groups
// for the mentors

estpost ttest profit businessage employees employeesnumber credit bankaccount loan
account marketing secondaryedu, by(treat)

eststo est4
```

```
esttab est4 using reg_table_1nd.tex, replace cells("mu_1 mu_2 b p count") ///
title("Balancing tests in mentor's selection") unstack r2  ///
label eqlabels(none) se


*------------- for variable profit

rd tprofit_endline ce_std, mbw(100 150 200)
eststo t

esttab t using "table1.tex", replace title('Profit') unstack r2 label eqlabels(none) se

*------------- for variable inventory


rd tinventory_endline ce_std, mbw(100 150 200)
eststo i

esttab i using "table2.tex", replace title('Profit') unstack r2 label eqlabels(none) se

*------------- For  variable arketing


rd marketing_endline ce_std, mbw(100 150 200)
eststo ma
esttab ma using "table_3.tex", replace title('Profit') unstack r2 label eqlabels(none)
se


*------------- For variable Keeps_Some_Records

eststo keeps_some

esttab keeps_some using "table4.tex", replace title('Profit') unstack r2 label
eqlabels(none) se



/* Graphs to compare the effect of the bins selection*/


*----- Greater bandwith, less bins


binscatter tprofit_endline ce_std, rd(0.25) n(100) line(qfit)


*----- Smaller Bandwidth, more bins

binscatter tprofit_endline ce_std, rd(0.25) n(200) line(qfit)


*-------------- Placebo Tests

*--- This will shift the ce_std 0.025 so that the units that were actually on zero will
be above

*--- shiftting the cutoff to 0.25 +


use "rd_mentors.dta", clear

rdrobust tprofit_endline ce_std if treat == 1, c(0.25) p(1)
rdrobust tinventory_endline ce_std if treat == 1, c(0.25) p(1)
rdrobust marketing_endline ce_std if treat == 1, c(0.25) p(1)
```

```
rdrobust keeps_some_records_endline ce_std if treat == 1, c(0.25) p(1)

binscatter tprofit_endline ce_std, rd(0.25) n(60) line(qfit)
```

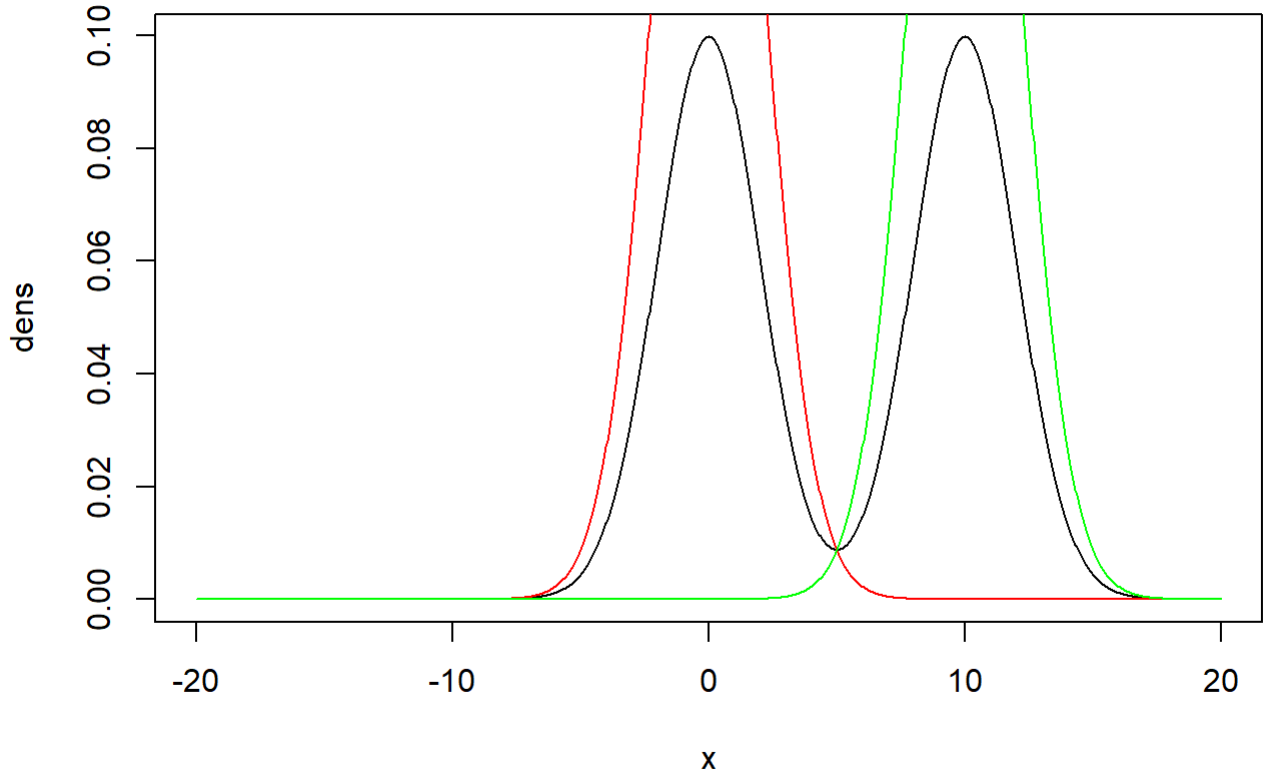
R CODING EXAMPLE

Johan Santiago Ruiz Moreno

Data Mining Course

Plotting univariate gaussian densities

```
library("mvtnorm")
ngrid <- 1001

ngrid <- 1001
x <- seq(-20, 20, length = ngrid)
dens1 <- dnorm(x, mean = 0, sd = 2)
dens2 <- dnorm(x, mean = 10, sd = 2)
#dens<-1/3*dens1 + 1/3*dens2 + 1/3*dens3
dens <-1/2*dens1 + 1/2*dens2

# The new density by clusters is a weighted sum of
# two different densities.
plot(x, dens, type = "l")
lines(x, dens1, col = "red")
lines(x, dens2, col = "green")
```



Using the “GMM” package for clustering detection

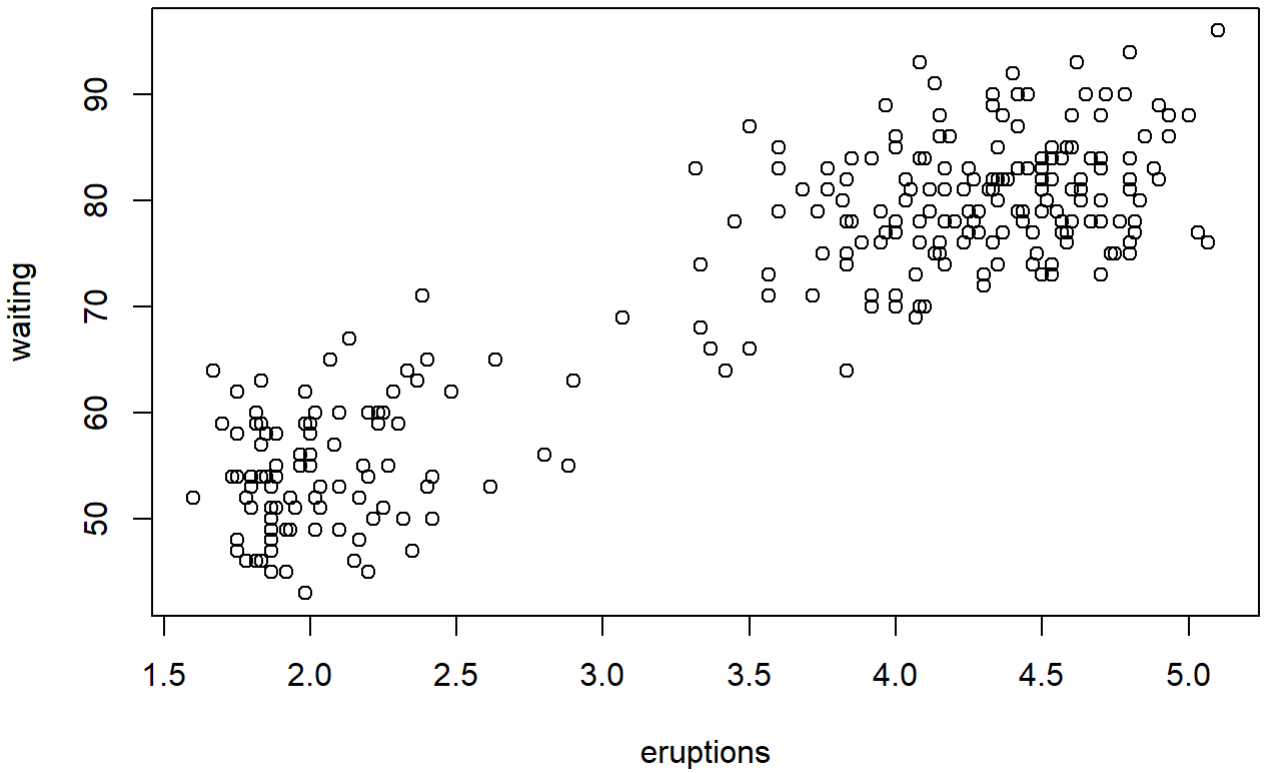
```
# Documentation of mclust
# https://cran.r-project.org/web/packages/mclust/mclust.pdf
library(mclust)
```

```
## Package 'mclust' version 5.4.7
## Type 'citation("mclust")' for citing this R package in publications.
```

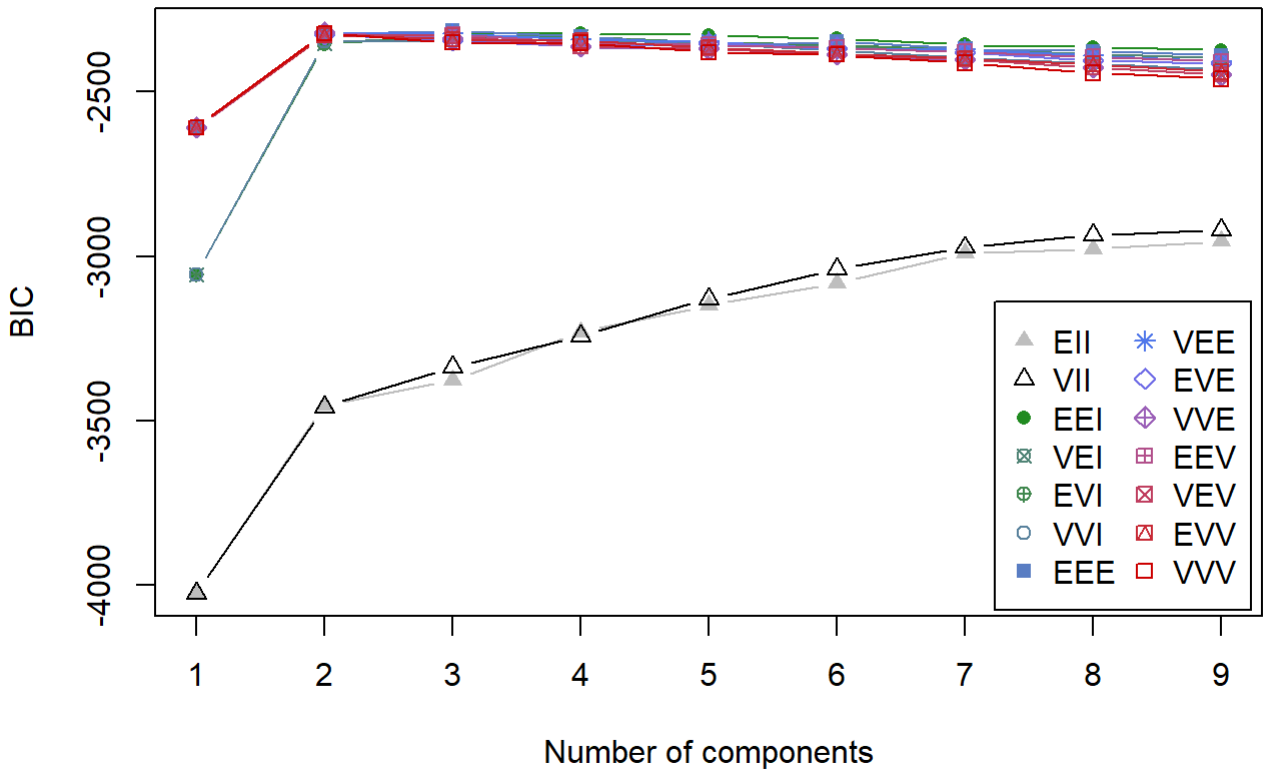
```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:mvtnorm':
##
##     dmvnorm
```

```
# The choice of the number of clusters can be done
# by optimizing the
# BIC "(Bayesian Information Criteria)"
# BIC=ln(n)k- 2 ln(L)
plot(faithful)
```



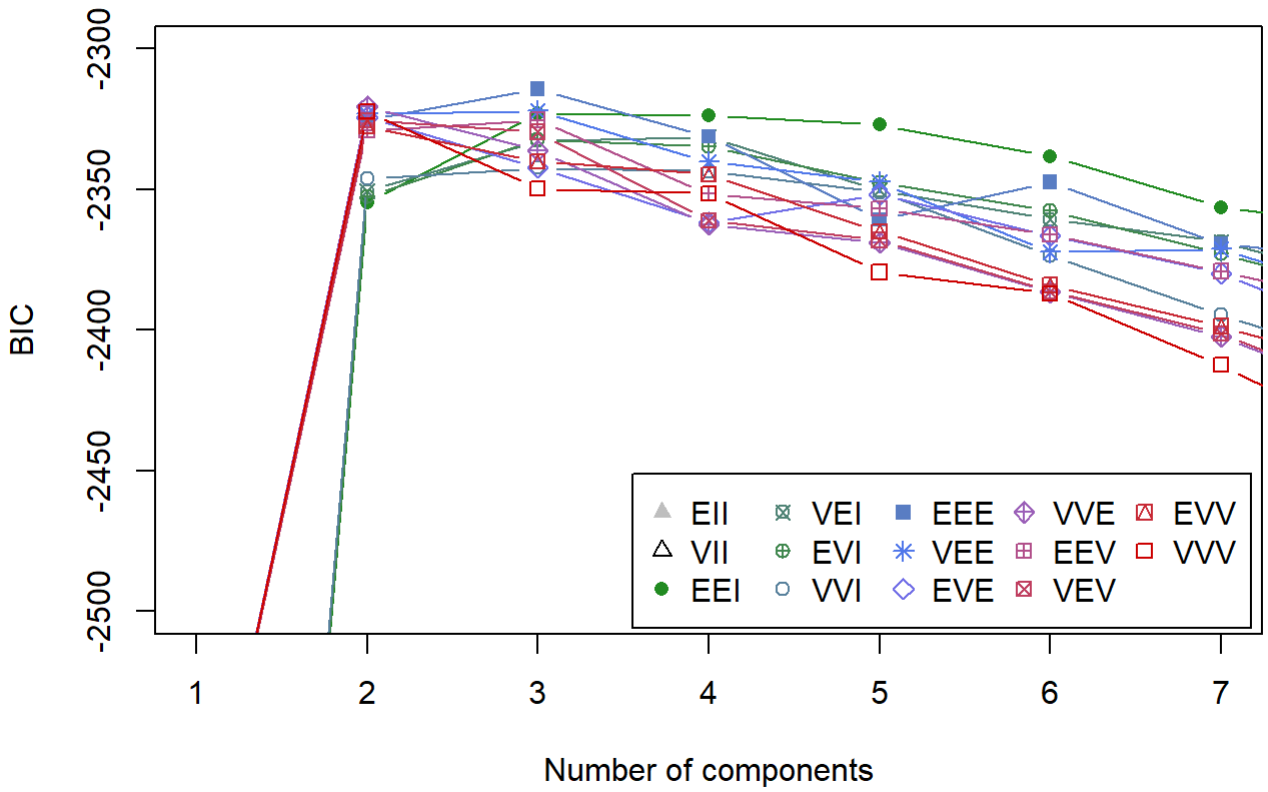

```
faithfulBIC <- mclustBIC(faithful)
# The mclustBIC for parameterized Gaussian mixture models
# fitted by EM algorithm initialized by model-based
# hierarchical clustering

# The plot fithfulBIC provides the BIC according to
# the covariance matrix and the number of clusters
plot(faithfulBIC)
```




```
# The optimal number of clusters is 2,
# according to the BIC plots, which actually
# tested different covariance matrix

# Summarizing the value of BIC, for the 3 best models
faithfulSummary <- summary(faithfulBIC,data = faithful)
faithfulSummary
```

```
## Best BIC values:
##              EEE,3        VVE,2        VEE,3
## BIC      -2314.316 -2320.432980 -2322.103490
## BIC diff     0.000    -6.116684    -7.787194
##
## Classification table for model (EEE,3):
##
##   1   2   3
##  40  97 135
```

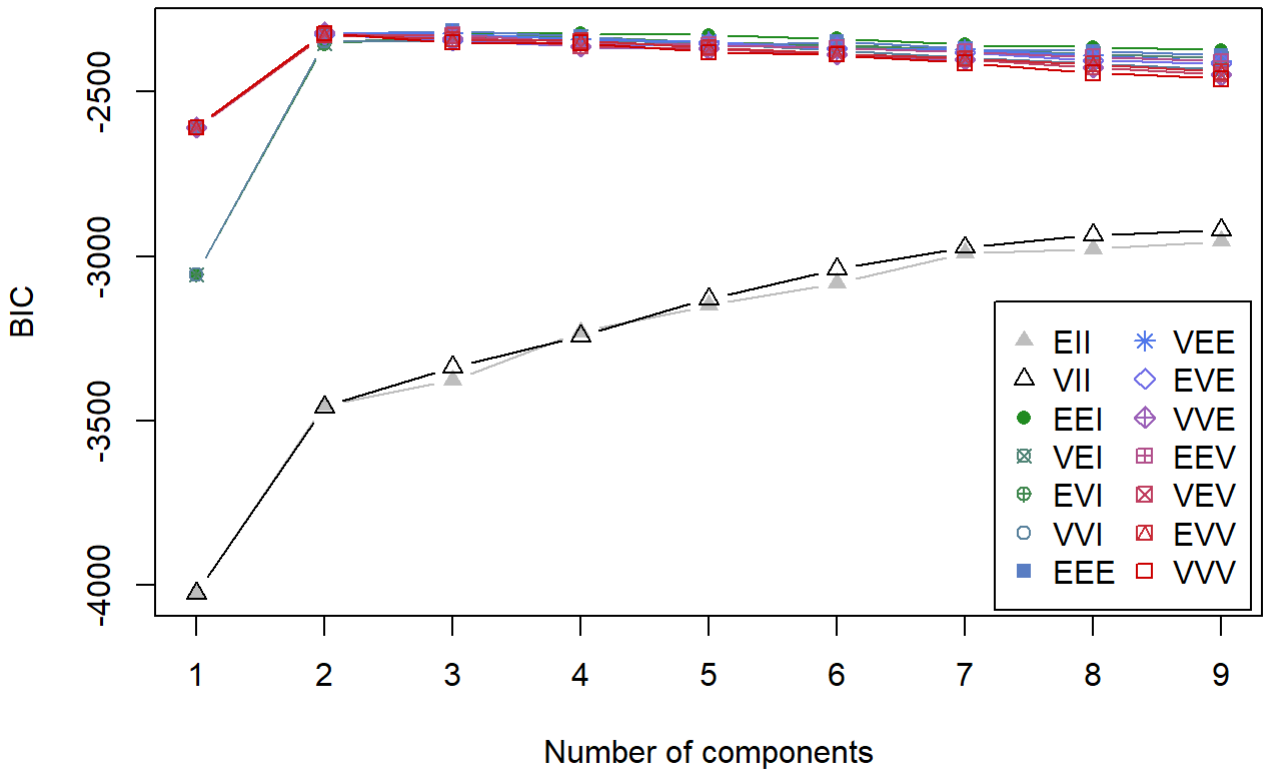
```
# BIC values for the number o clusters from 1 to 7
plot(faithfulBIC, G = 1:7, ylim = c(-2500,-2300), legendArgs = list(x = "bottomright", ncol = 5))
```

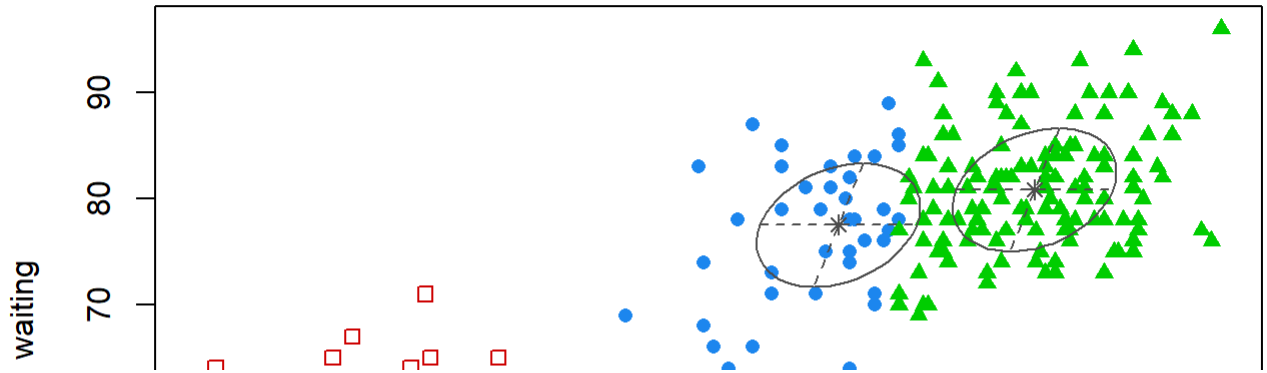


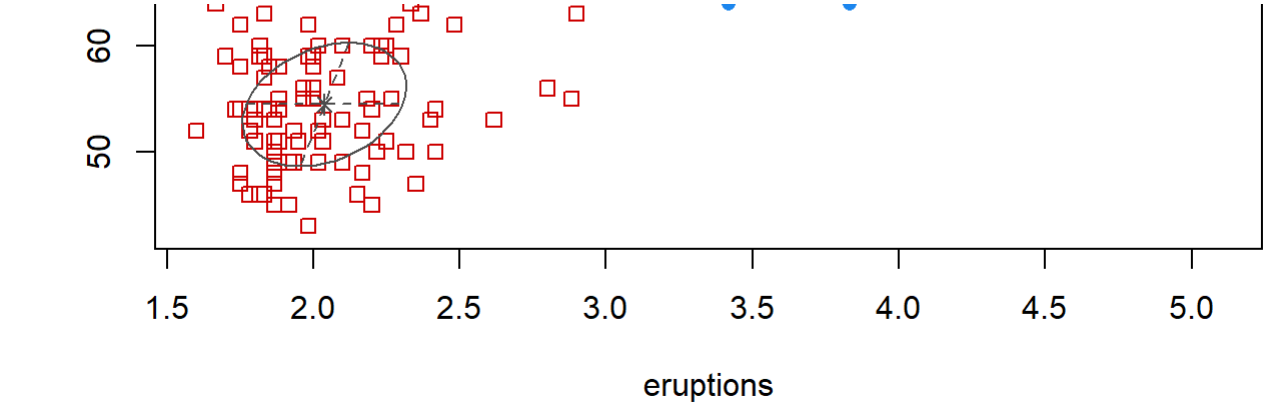

```
# fitting the model by using the number of parameters provided by the mclustBIC function
# It automatically selects the best model from the mclustBIC function
faithfulMclust<-Mclust(faithful,x=faithfulBIC)
# Printing results:
# According to the results, the functions used found
# three different clusters in the data.
summary(faithfulMclust,parameters=TRUE)
```

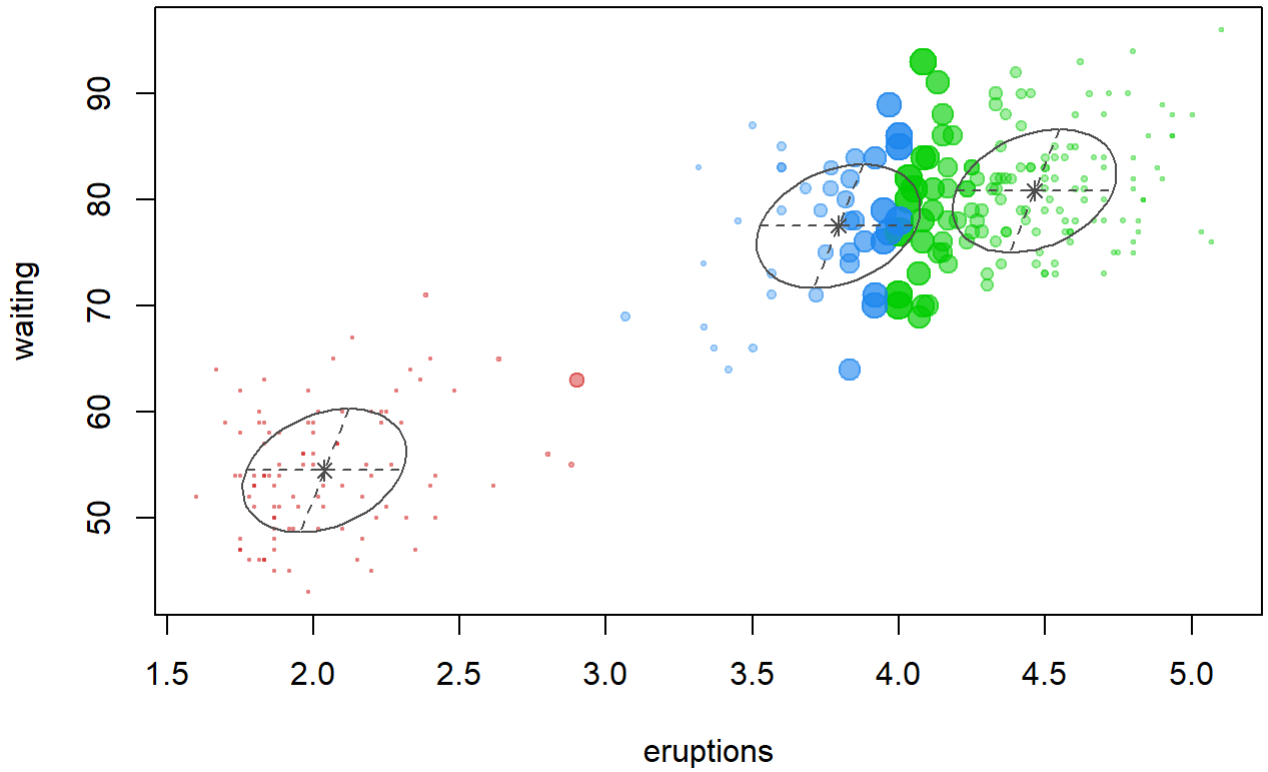
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 3
## components:
##
##  log-likelihood   n df       BIC       ICL
##       -1126.326 272 11 -2314.316 -2357.824
##
## Clustering table:
##   1   2   3
##  40  97 135
##
## Mixing probabilities:
##         1         2         3
## 0.1656784 0.3563696 0.4779520
##
## Means:
##                [,1]      [,2]      [,3]
## eruptions  3.793066  2.037596  4.463245
## waiting   77.521051 54.491158 80.833439
##
## Variances:
## [,,1]
##            eruptions    waiting
## eruptions 0.07825448  0.4801979
## waiting   0.48019785 33.7671464
## [,,2]
##            eruptions    waiting
## eruptions 0.07825448  0.4801979
## waiting   0.48019785 33.7671464
## [,,3]
##            eruptions    waiting
## eruptions 0.07825448  0.4801979
## waiting   0.48019785 33.7671464
```

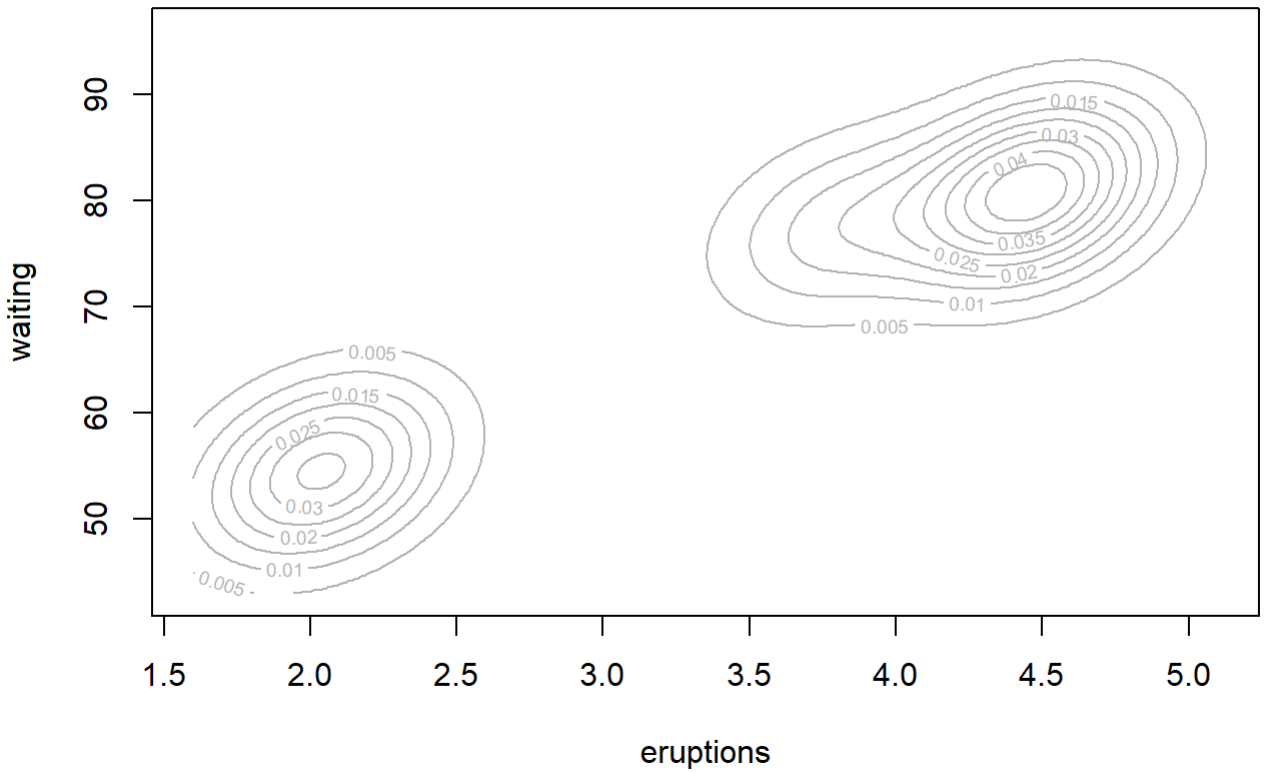
```
# It willl plot the densities, then one can obtain the models that fit the best the data, and the iso probabilities and
plot(faithfulMclust)
```


BIC
-2500
-3000
-3500
-4000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV


waiting
90
80
70


60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions


waiting
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions




```
# The object yielded by the function enables us to obtain directly a value
# saying to which cluster belongs each observation.
faithfulMclust$classification
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   2   1   2   3   2   3   1   2   3   2   1   3   2   3   2   2   3   2   3
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   1   1   3   1   2   3   1   3   3   3   1   3   1   2   2   3   2   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   3   2   3   2   3   1   1   2   3   2   3   3   2   3   2   3   1   2   3   3
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   3   2   3   2   3   3   3   2   3   3   2   3   3   2   3   2   3   1   1
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   3   3   2   3   3   1   3   2   1   2   3   2   3   2   3   3   1   2   3
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   3   2   3   3   2   3   2   3   1   3   2   3   3   2   3   2   3   2   3
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   3   3   2   3   1   2   3   2   3   2   3   2   3   2   3   2   3   2   1
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   3   2   3   3   3   2   3   2   3   2   3   3   2   3   1   3   3   3   2   1
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   3   2   1   1   3   2   3   2   3   2   2   3   1   3   3   3   2   1   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   2   3   3   1   2   3   3   2   3   2   3   2   3   3   1   3   1   3   2   3
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   2   3   3   2   3   2   3   1   2   3   2   3   2   1   1   3   2   3   2   3
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   2   3   2   3   1   3   3   3   1   3   3   2   3   2   3   2   2   3   1   2
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   3   2   3   2   3   1   2   3   2   3   2   3   1   3   3   1   1   3   2   3
## 261 262 263 264 265 266 267 268 269 270 271 272
##   3   3   2   3   2   2   3   3   2   3   2   3
```

```
# That's how one can get the uncertainty of each value
faithfulMclust$uncertainty
```

```
##            1            2            3            4            5            6
## 2.819635e-02 8.604228e-13 3.260091e-03 3.140301e-07 1.168409e-02 3.093050e-03
##            7            8            9           10           11           12
## 2.985953e-03 2.387092e-02 5.227374e-12 5.522795e-02 1.662226e-12 2.862230e-01
##           13           14           15           16           17           18
## 1.507334e-01 1.976197e-14 2.590469e-03 5.899479e-10 7.554180e-12 1.114522e-03
##           19           20           21           22           23           24
## 7.105427e-15 1.055467e-01 2.620126e-13 1.976197e-14 7.990382e-03 3.130850e-02
##           25           26           27           28           29           30
## 8.567092e-03 2.523553e-02 3.580736e-11 3.177812e-01 2.094909e-01 2.330837e-02
##           31           32           33           34           35           36
## 6.038527e-02 1.647497e-02 5.722475e-03 4.468733e-01 2.038357e-01 2.957812e-11
##           37           38           39           40           41           42
## 3.039791e-13 7.457201e-04 1.205480e-11 1.533348e-03 4.824795e-02 2.199596e-11
##           43           44           45           46           47           48
## 8.465780e-03 1.548095e-12 8.328420e-03 2.182189e-03 2.539722e-01 2.303091e-10
##           49           50           51           52           53           54
## 4.511302e-03 3.375327e-10 8.626203e-04 2.749461e-03 1.662226e-12 7.457201e-04
##           55           56           57           58           59           60
## 2.260414e-13 5.249817e-04 9.194328e-02 3.185230e-12 6.945520e-03 6.505602e-02
##           61           62           63           64           65           66
## 3.527008e-08 1.509693e-02 2.953193e-14 1.052845e-03 1.301959e-11 4.407861e-02
##           67           68           69           70           71           72
## 1.914459e-01 2.247248e-03 1.385095e-08 1.949413e-03 4.609991e-01 5.321898e-11
##           73           74           75           76           77           78
## 1.311848e-02 4.547362e-01 7.893755e-10 8.625443e-05 7.042484e-10 7.144847e-03
##           79           80           81           82           83           84
## 2.723389e-01 2.523553e-02 2.263256e-01 5.860821e-02 2.528565e-01 1.111078e-03
##           85           86           87           88           89           90
## 3.296582e-01 3.912239e-04 4.018847e-01 1.165256e-02 1.209026e-10 4.388383e-01
##           91           92           93           94           95           96
## 2.713534e-08 7.252584e-02 6.712408e-13 8.100026e-04 5.882117e-11 5.878859e-02
##           97           98           99          100          101          102
## 3.552273e-03 1.075611e-01 9.978685e-13 4.398691e-04 1.698596e-05 5.203187e-02
##          103          104          105          106          107          108
## 4.719913e-11 1.467904e-02 4.174397e-01 2.045031e-13 2.665148e-03 2.775558e-13
##          109          110          111          112          113          114
## 7.626731e-04 5.355313e-02 1.547597e-03 1.342776e-07 5.369192e-04 2.671072e-02
##          115          116          117          118          119          120
```

```
## 8.484324e-13 4.385121e-03 5.326828e-09 6.542032e-03 8.760104e-12 3.332180e-02
##          121          122          123          124          125          126
## 6.957309e-06 3.049762e-01 1.002861e-01 5.321898e-11 7.121726e-03 1.054207e-01
##          127          128          129          130          131          132
## 2.511324e-13 1.427254e-02 1.424491e-08 4.882705e-03 9.259260e-14 2.144739e-01
##          133          134          135          136          137          138
## 8.793720e-04 7.063228e-02 6.972201e-14 3.867279e-02 1.373124e-12 3.695592e-04
##          139          140          141          142          143          144
## 6.049428e-11 8.483812e-02 1.265745e-01 5.242048e-08 1.073705e-02 7.872649e-04
##          145          146          147          148          149          150
## 4.985685e-02 2.404363e-10 4.262454e-03 9.009238e-12 1.143155e-04 5.788703e-13
##          151          152          153          154          155          156
## 1.194282e-04 4.973550e-01 1.064458e-05 5.841405e-03 2.659201e-02 4.476804e-01
##          157          158          159          160          161          162
## 1.387715e-02 4.305601e-01 5.788703e-13 3.498765e-01 7.114065e-11 2.565069e-01
##          163          164          165          166          167          168
## 2.271021e-10 1.859608e-01 1.726170e-02 5.876348e-03 2.494465e-06 2.179539e-04
##          169          170          171          172          173          174
## 5.534240e-12 7.079430e-03 1.225020e-12 8.005225e-10 6.045178e-03 4.073319e-03
##          175          176          177          178          179          180
## 2.050287e-01 5.705566e-02 1.107969e-02 3.917705e-08 4.458673e-01 1.744195e-01
##          181          182          183          184          185          186
## 6.699752e-12 6.045178e-03 1.168011e-01 1.001656e-01 2.738587e-11 2.266841e-02
##          187          188          189          190          191          192
## 3.691129e-01 6.972201e-14 2.983931e-02 2.665306e-09 1.023298e-03 5.457190e-12
##          193          194          195          196          197          198
## 8.875324e-04 3.352576e-01 4.288923e-01 1.265745e-01 9.552900e-03 3.889645e-02
##          199          200          201          202          203          204
## 2.079535e-09 2.995688e-03 3.689545e-09 5.093338e-02 3.157763e-01 2.204015e-12
##          205          206          207          208          209          210
## 5.365363e-03 2.575717e-14 3.857125e-02 1.825793e-01 1.685763e-12 1.467904e-02
##          211          212          213          214          215          216
## 8.172391e-05 2.378727e-03 4.516387e-13 1.992503e-01 9.128001e-03 1.116429e-01
##          217          218          219          220          221          222
## 9.162192e-08 1.481435e-03 6.917311e-11 2.060082e-01 6.712408e-13 9.974611e-02
##          223          224          225          226          227          228
## 3.173017e-13 1.357264e-02 4.955213e-01 2.737949e-01 3.302628e-01 8.996764e-02
##          229          230          231          232          233          234
## 3.740539e-01 8.516287e-03 2.819197e-01 1.911655e-07 2.054747e-01 7.242791e-10
```

```
##          235          236          237          238          239          240
## 2.737454e-02 4.507728e-12 2.333467e-12 7.711069e-02 3.815193e-01 1.881226e-06
##          241          242          243          244          245          246
## 2.013863e-01 3.134375e-09 3.695592e-04 9.396093e-02 7.581242e-03 1.693200e-01
##          247          248          249          250          251          252
## 8.005225e-10 4.421637e-02 1.141826e-07 4.097625e-02 2.517493e-09 2.253588e-02
##          253          254          255          256          257          258
## 2.515564e-02 1.107969e-02 2.675256e-01 1.579973e-01 3.674064e-01 2.253588e-02
##          259          260          261          262          263          264
## 1.028093e-10 8.126541e-02 1.252990e-03 1.135958e-02 1.138623e-11 1.168011e-01
##          265          266          267          268          269          270
## 4.241052e-13 7.358963e-08 1.334346e-03 2.852713e-01 3.898792e-11 3.618804e-02
##          271          272
## 5.062617e-14 1.514555e-02
```

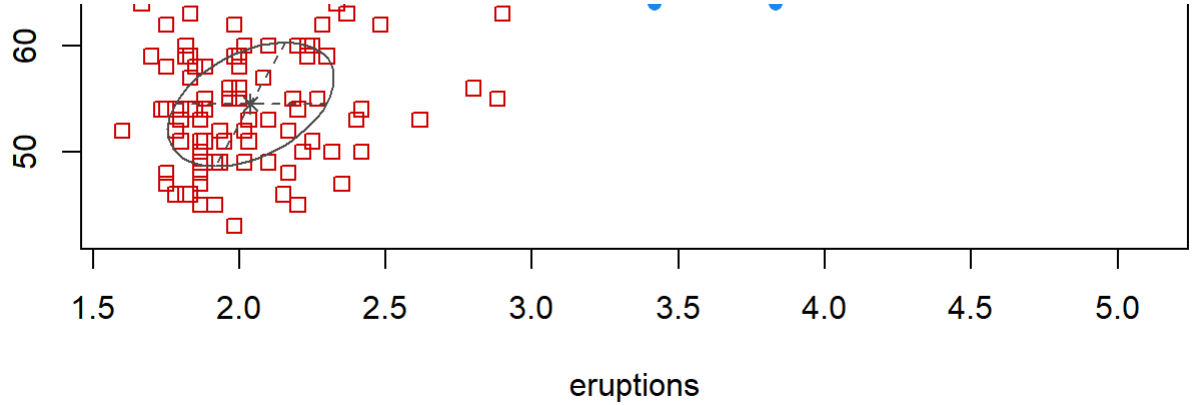
```
# Replicating the problem and only using two clusters this time
# by using the option G = 2

faithfulVVE<-Mclust(faithful,G=2,modelNames="VVE")
summary(faithfulVVE,parameters=TRUE)
```

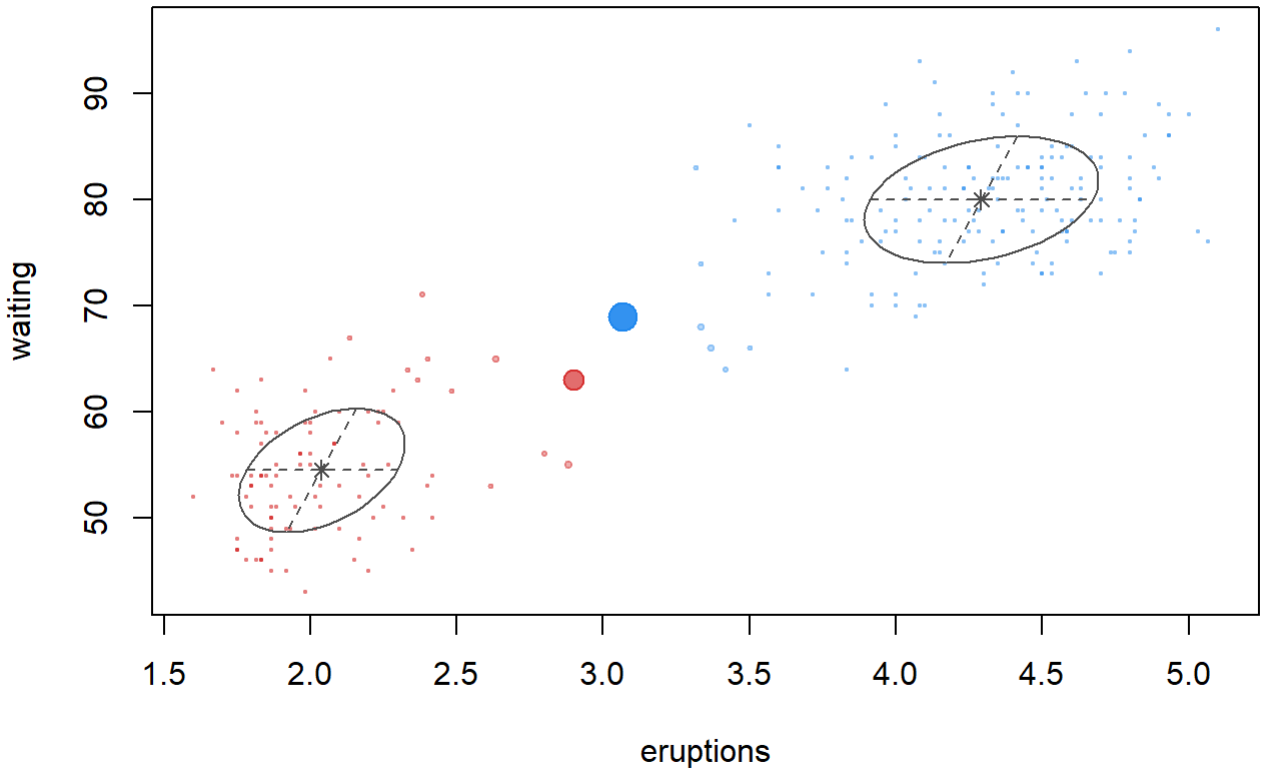
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1132.187 272 10 -2320.433 -2320.763
##
## Clustering table:
##   1   2
## 175  97
##
## Mixing probabilities:
##         1         2
## 0.6431778 0.3568222
##
## Means:
##                [,1]      [,2]
## eruptions  4.291633  2.038831
## waiting   79.990255 54.506423
##
## Variances:
## [,,1]
##           eruptions    waiting
## eruptions 0.1600175  0.7272673
## waiting   0.7272673 35.7585543
## [,,2]
##            eruptions    waiting
## eruptions 0.08069286  0.6912995
## waiting   0.69129952 33.9186657
```

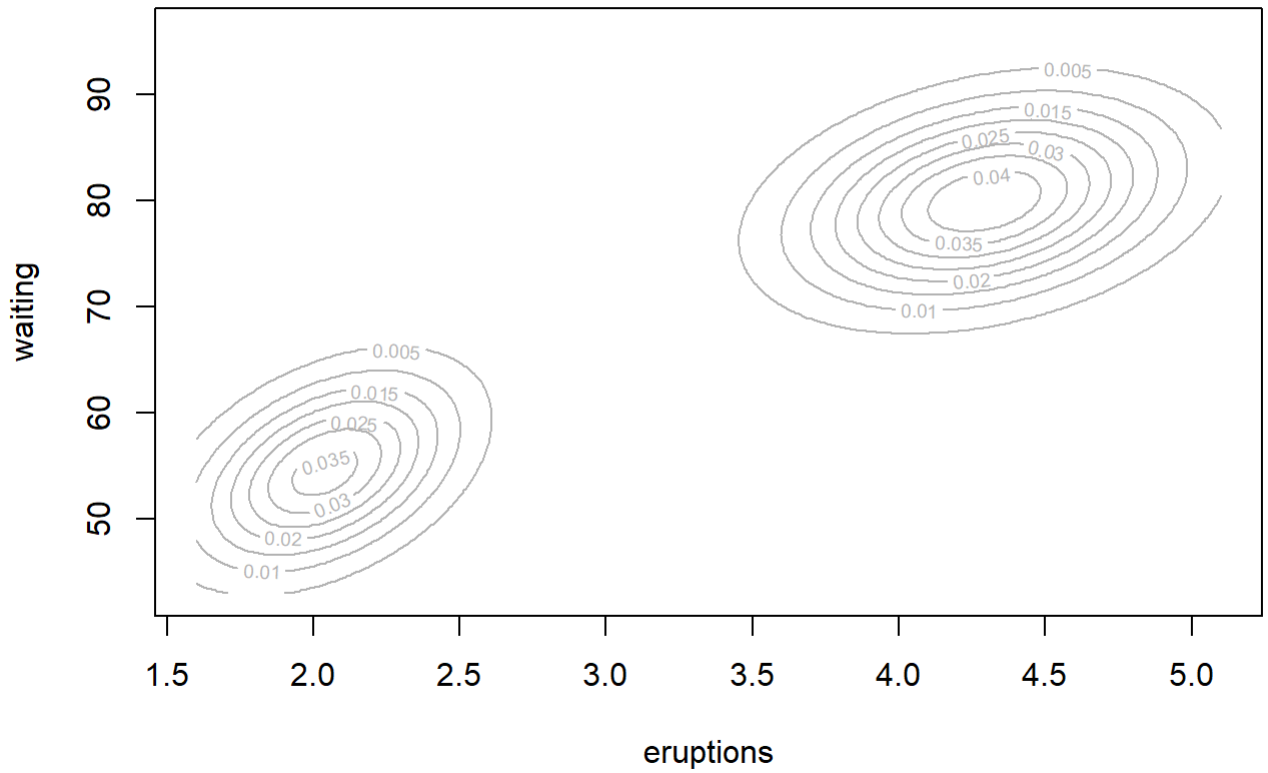
```
plot(faithfulVVE)
```

BIC
-1500
-2000
-2500
-3000
VVE
2
Number of components

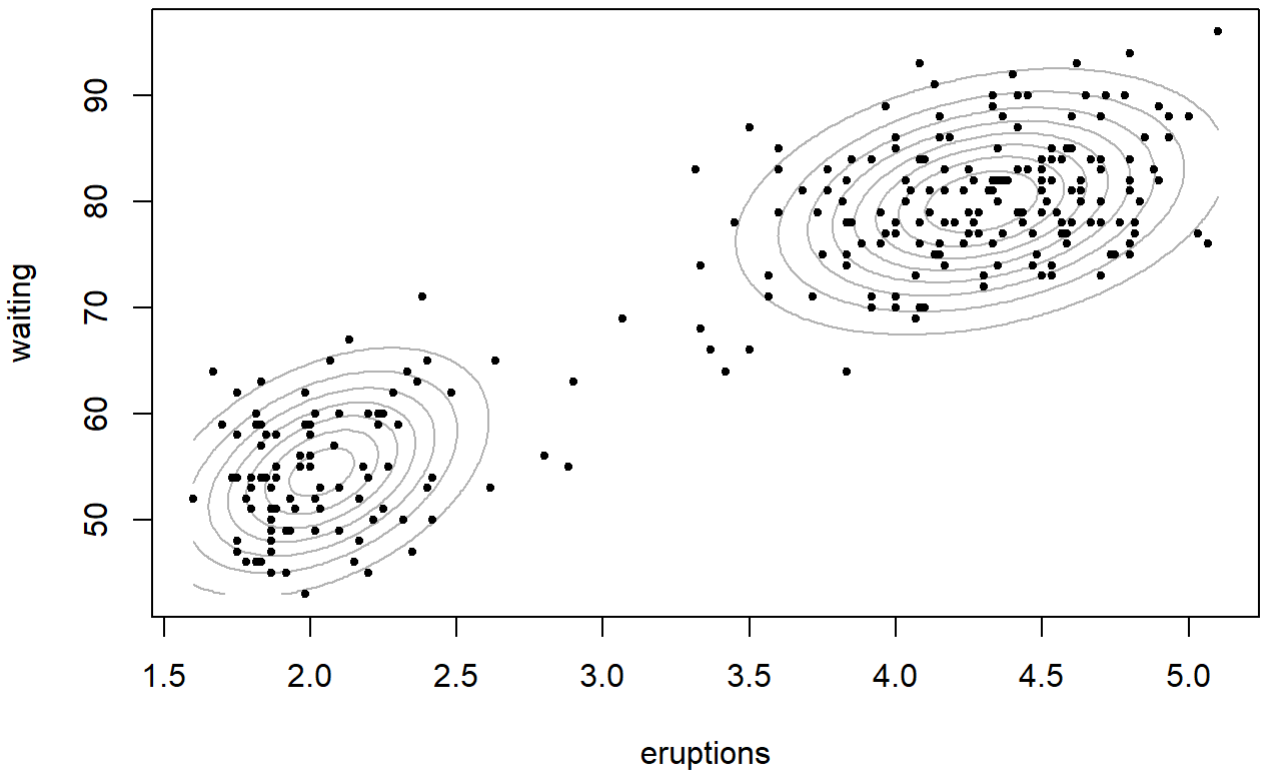
waiting
90
80
70


60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions

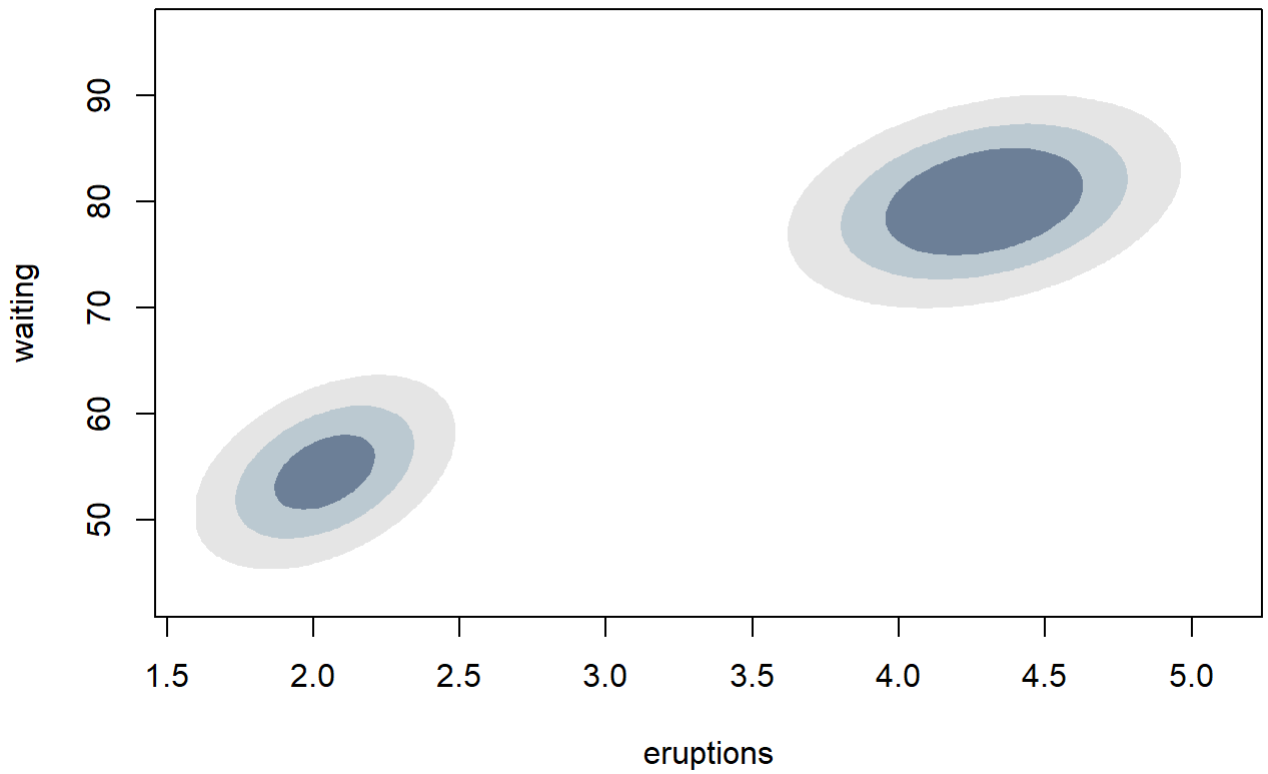

waiting
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions



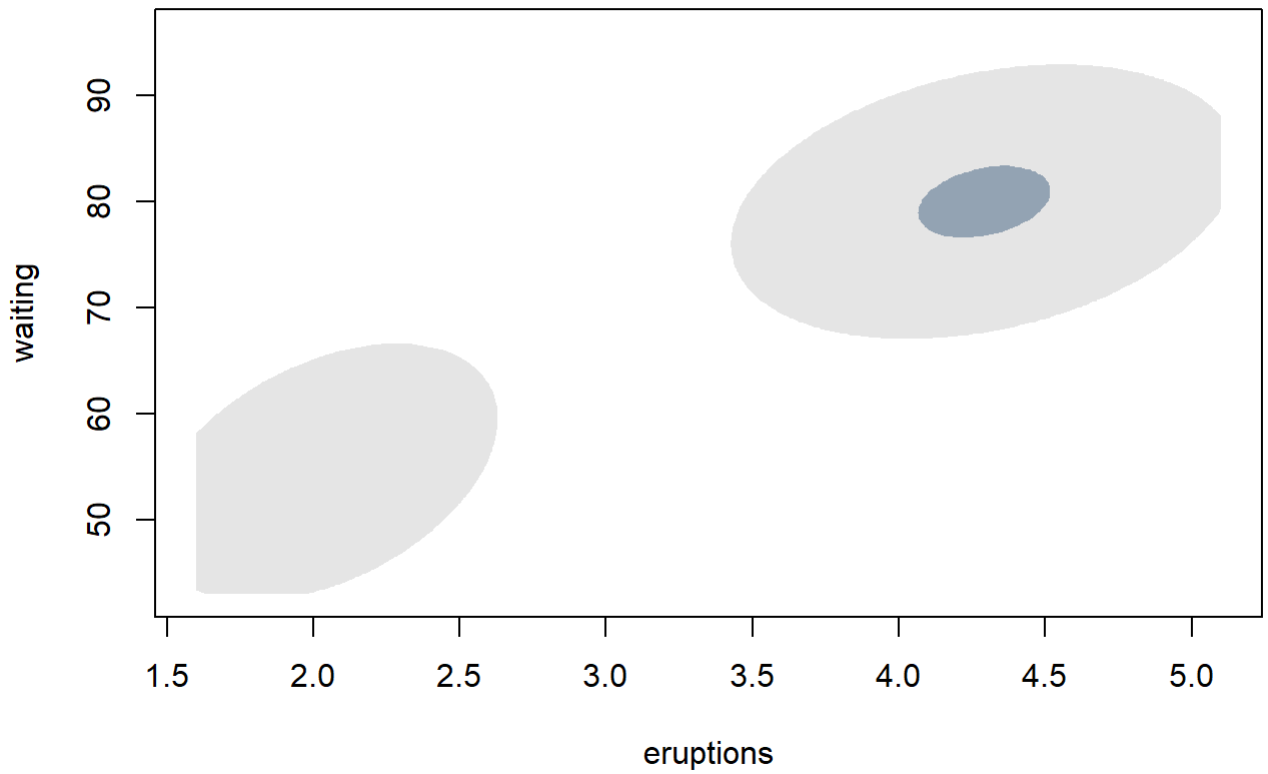

```
# Trying with different kind of plots
dens<-densityMclust(faithful,G=2,modelNames="VVE")
plot(dens, what = "density", data = faithful,drawlabels = FALSE, points.pch = 20)
```



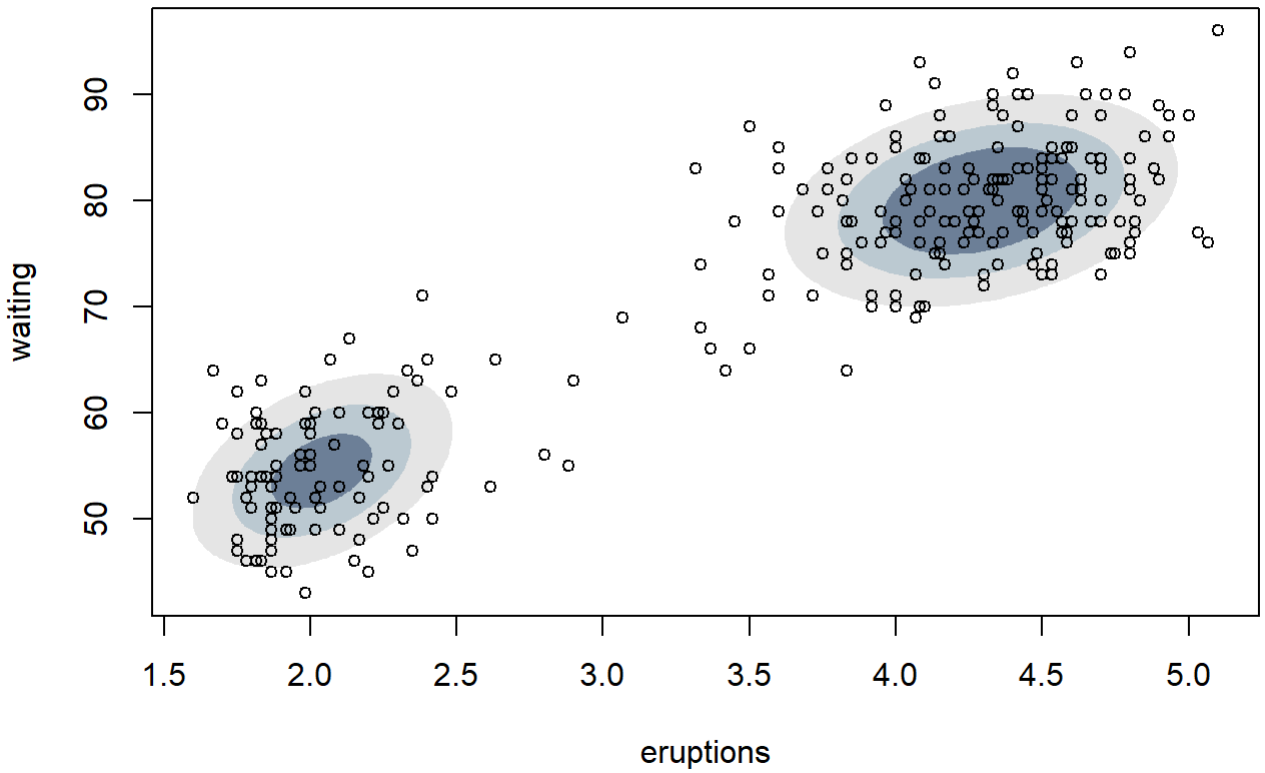

```
plot(dens, what = "density", type = "hdr")
```



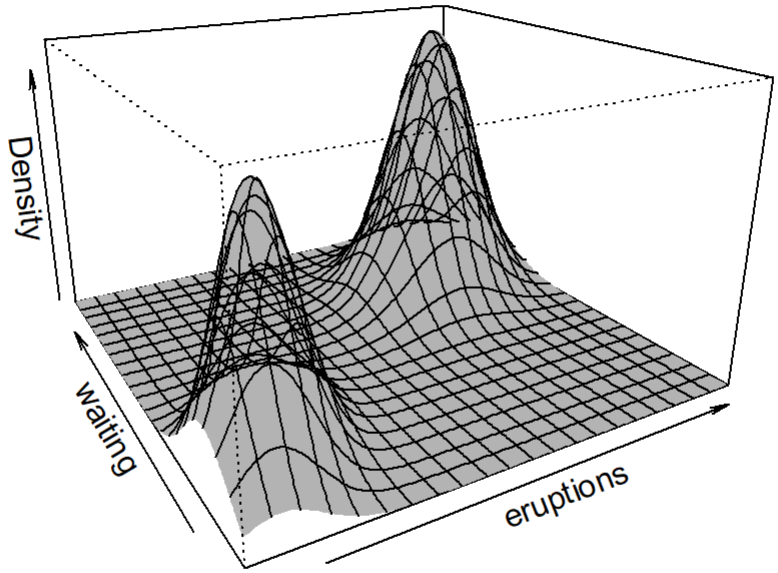

```
plot(dens, what = "density", type = "hdr", prob = c(0.1, 0.9))
```




```
plot(dens, what = "density", type = "hdr", data = faithful)
```




```
plot(dens, what = "density", type = "persp")
```



Using the EM algorithm

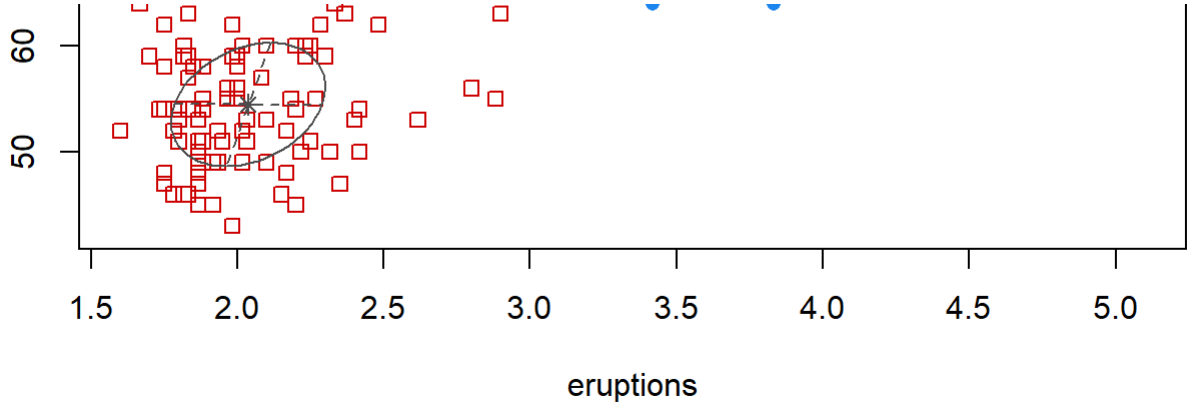
```
# Allowing for three clusters with a flexible covariance matrix
faithfulVVV<- Mclust(faithful,G=2,modelNames="VVV")
# summary of our new model
summary(faithfulVVE)
```

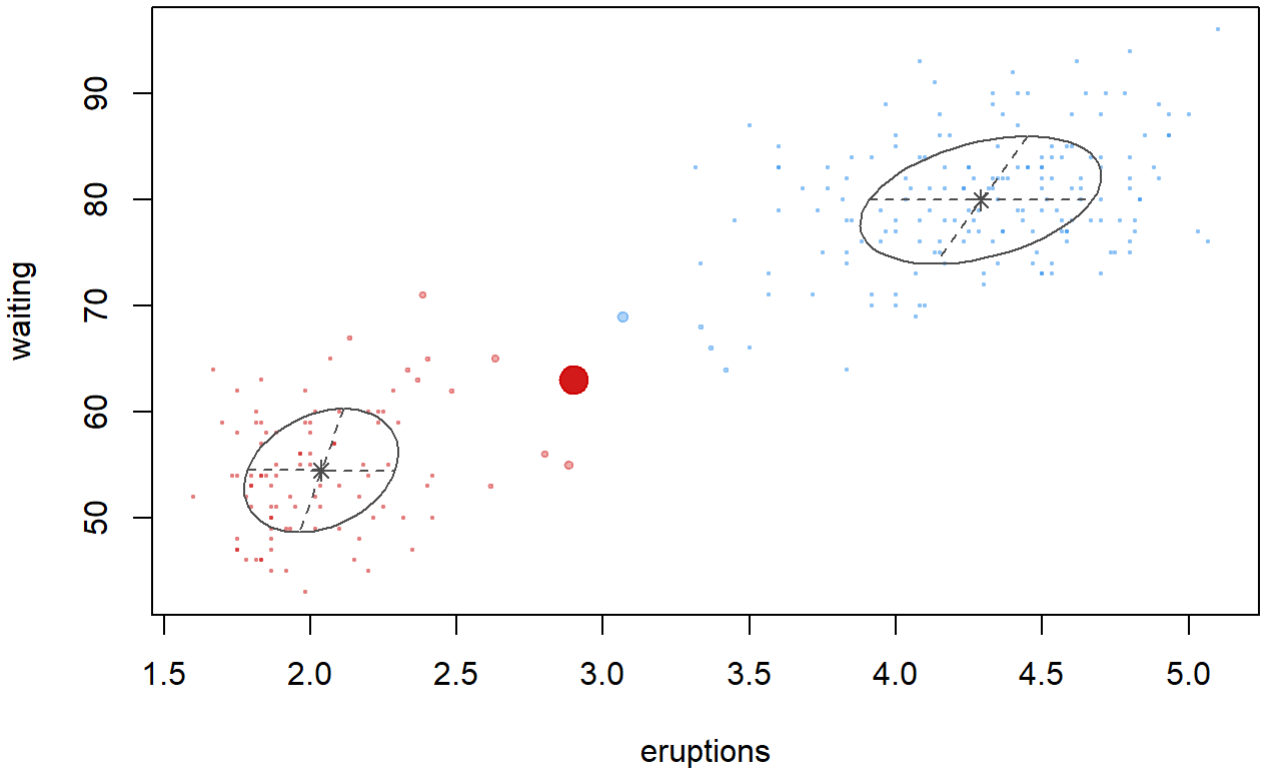
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVE (ellipsoidal, equal orientation) model with 2 components:
##
##  log-likelihood   n df       BIC       ICL
##       -1132.187 272 10 -2320.433 -2320.763
##
## Clustering table:
##   1   2
## 175  97
```

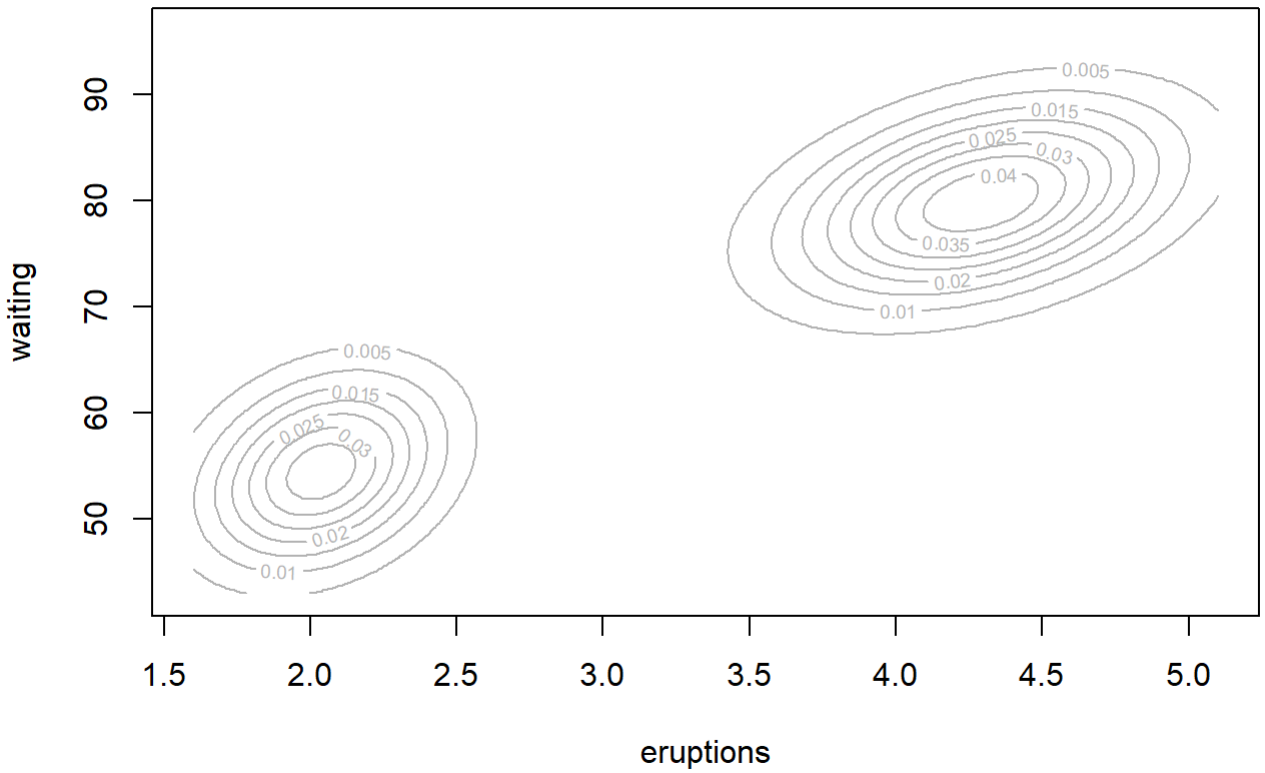
```
plot(faithfulVVV)
```

BIC
-1500
-2000
-2500
-3000
VVV
2
Number of components

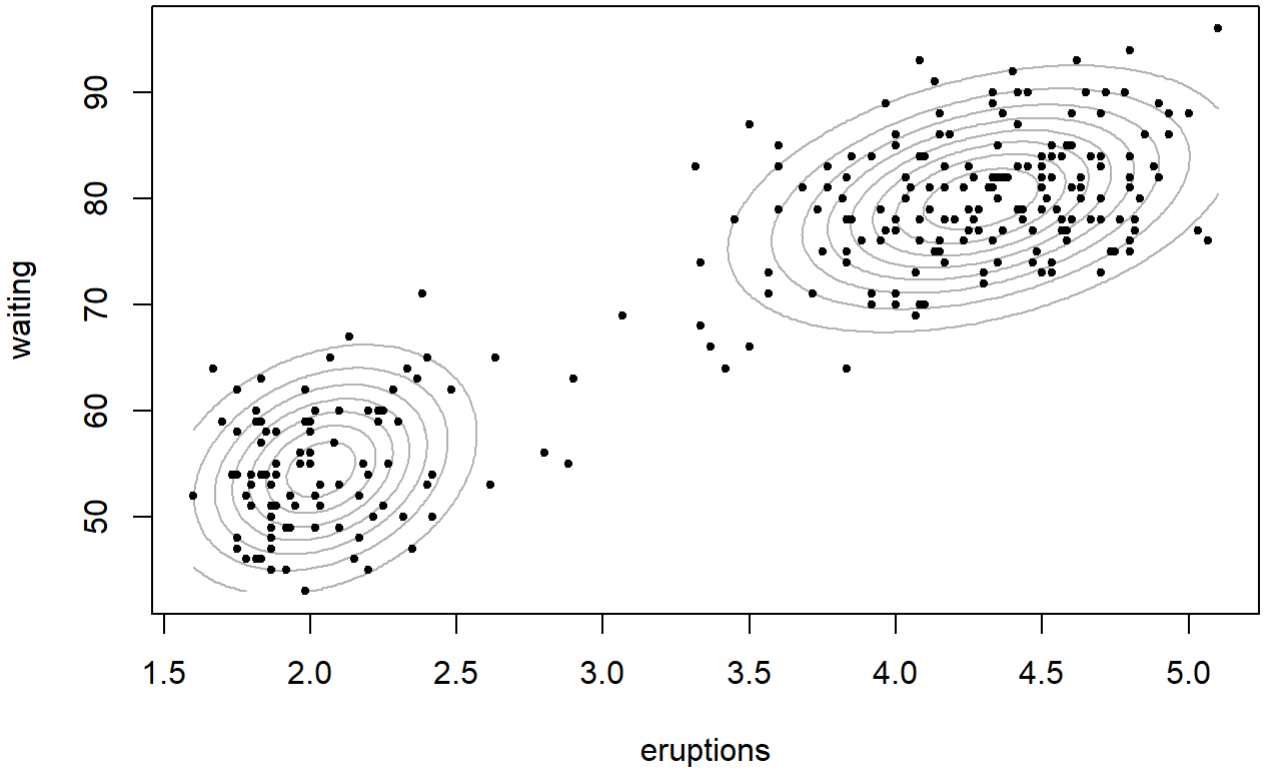
waiting
90
80
70


60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions


waiting
90
80
70
60
50
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
eruptions




```
dens<-densityMclust(faithful,G=2,modelNames="VVV")
# New estimation with the newly defined parameters,
plot(dens, what = "density", data = faithful,drawlabels = FALSE, points.pch = 20)
```



Application with the Iris data

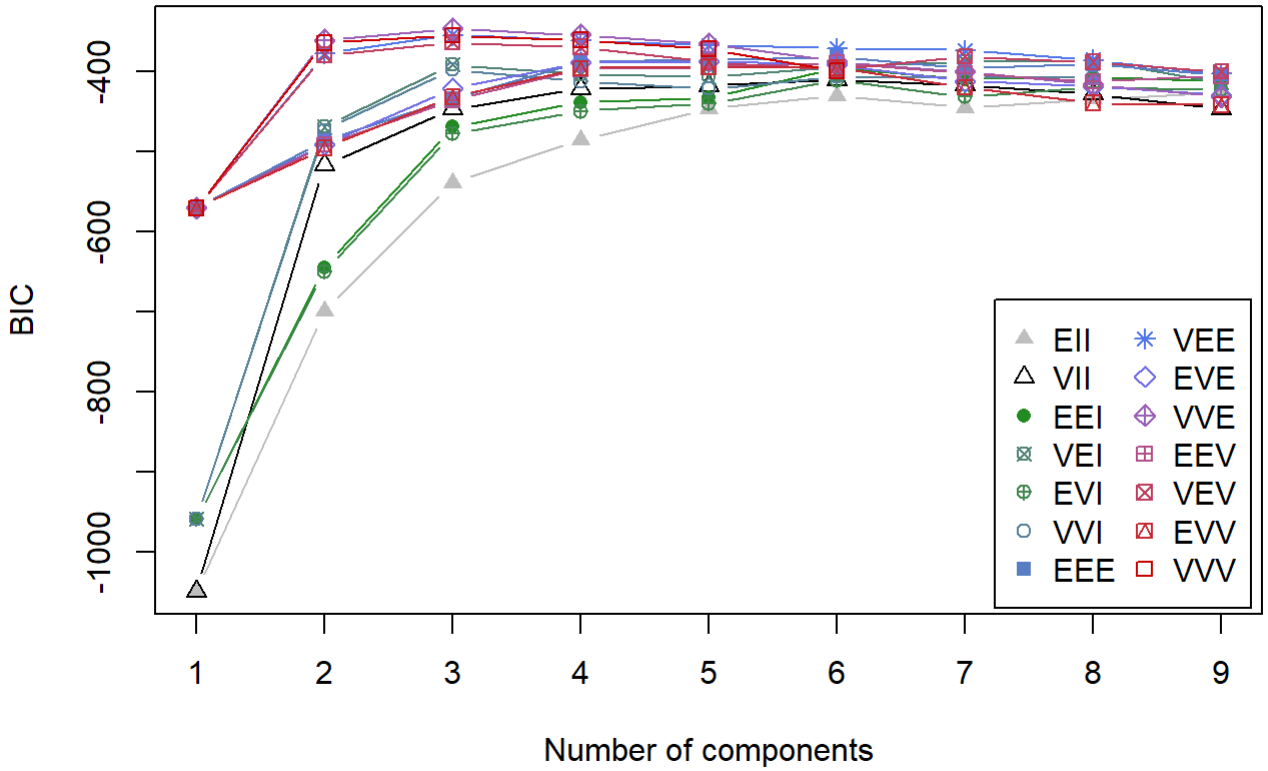
```
train_data <- (iris)
summary(iris)
```

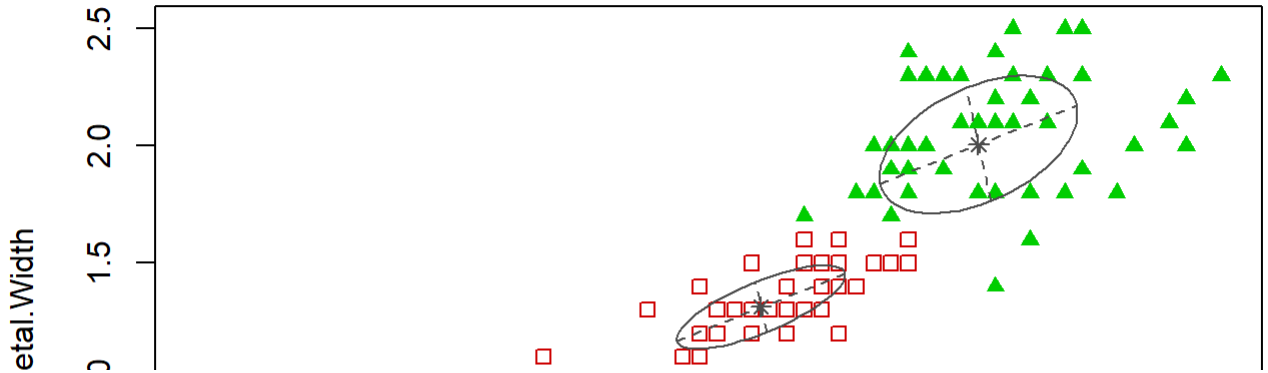
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

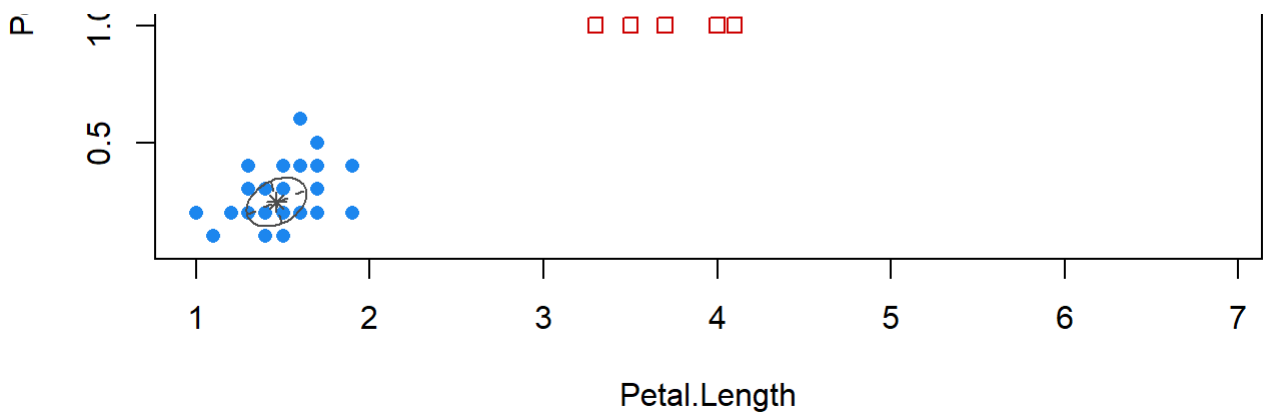
```
# clustering the data for the variables sepal.width and PetaLength into three different clusters

data =  iris[,c(3,4)]

# Keeping only the lentgh and the Width to cluster for the 3 different categories

dataBIC <- mclustBIC(data)
#Plotting the Bayesian Information Criteria
plot(dataBIC)
# According to the BIC the best models would have between 3 and 2 clusters

dataSummary <- summary(dataBIC,data = data)
# the model yields three categories with 50 values each, same as the original data


dataMclust<-Mclust(data,x=dataBIC)
# Here below is the classification provided by the model estimated
dataMclust$classification
```

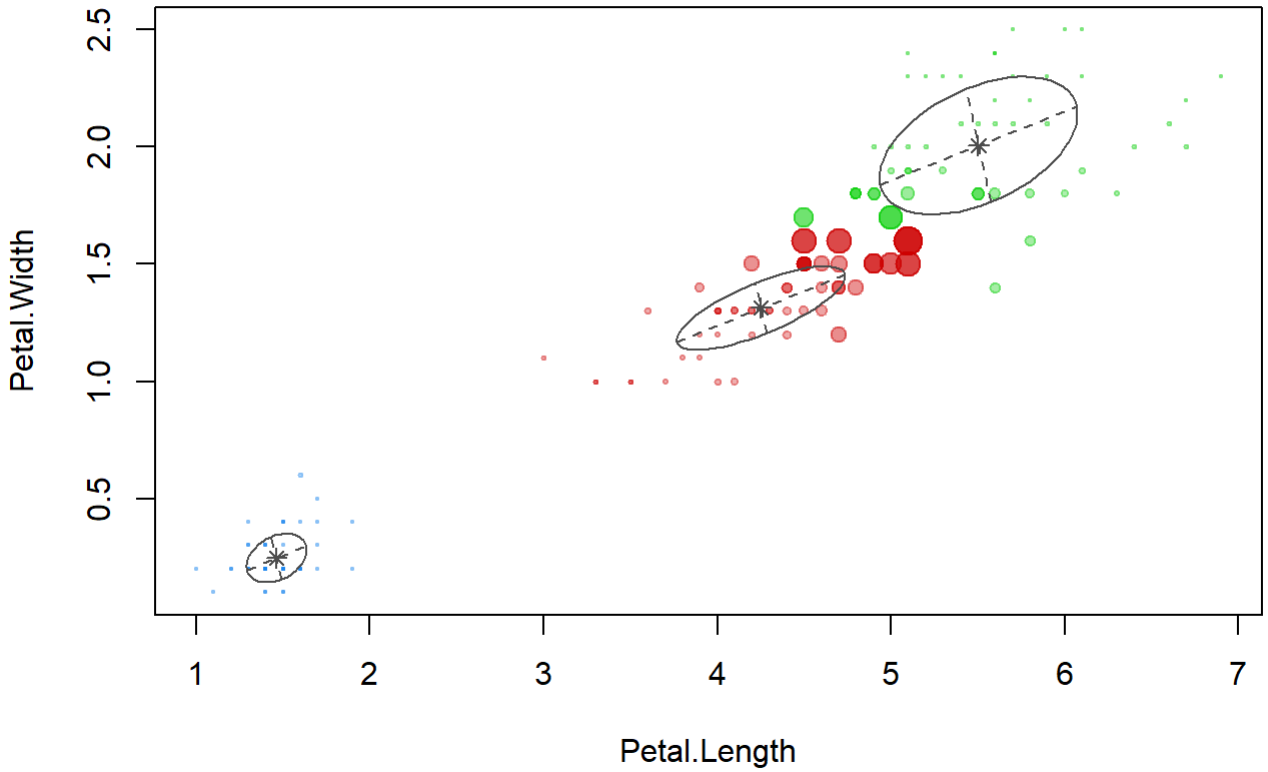
```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2
##  [75] 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3
## [149] 3 3
```

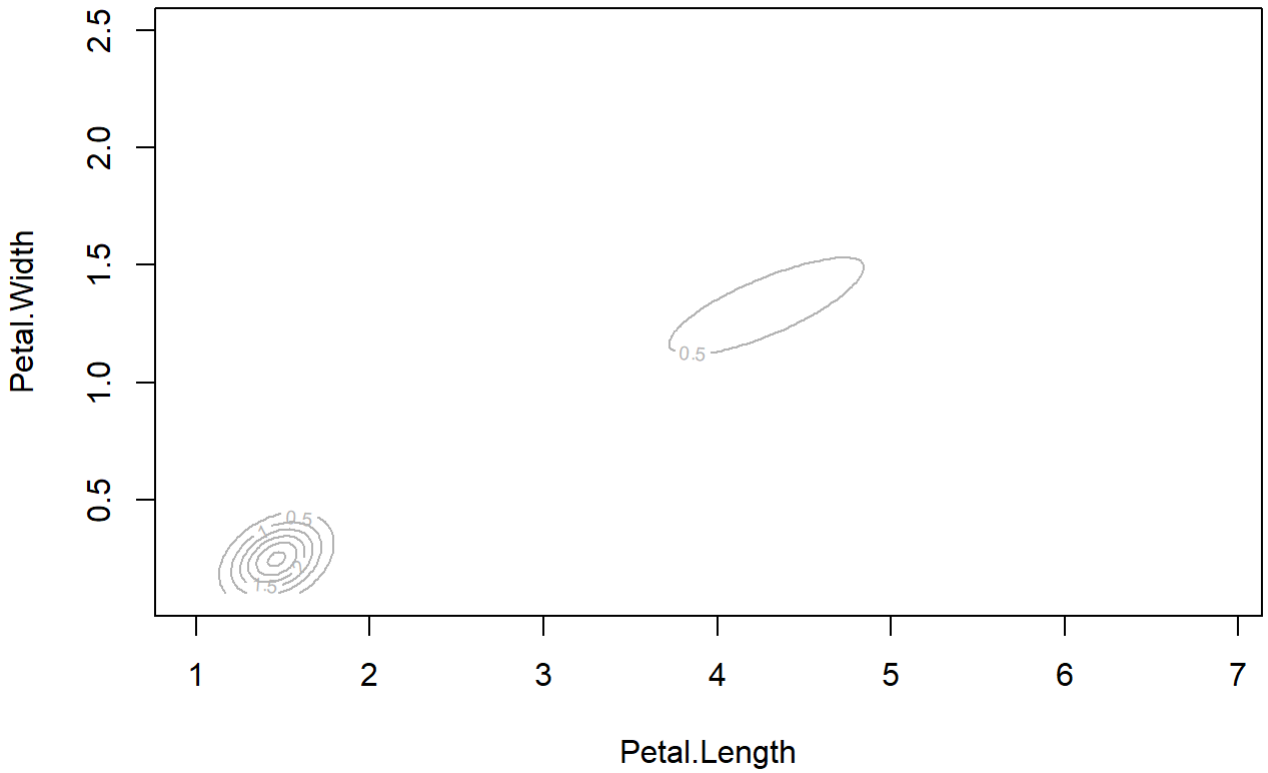
```
plot(dataMclust)
```


BIC
-400
-600
-800
-1000
1
2
3
4
5
6
7
8
9
Number of components
EII
VII
EEI
VEI
EVI
VVI
EEE
VEE
EVE
VVE
EEV
VEV
EVV
VVV

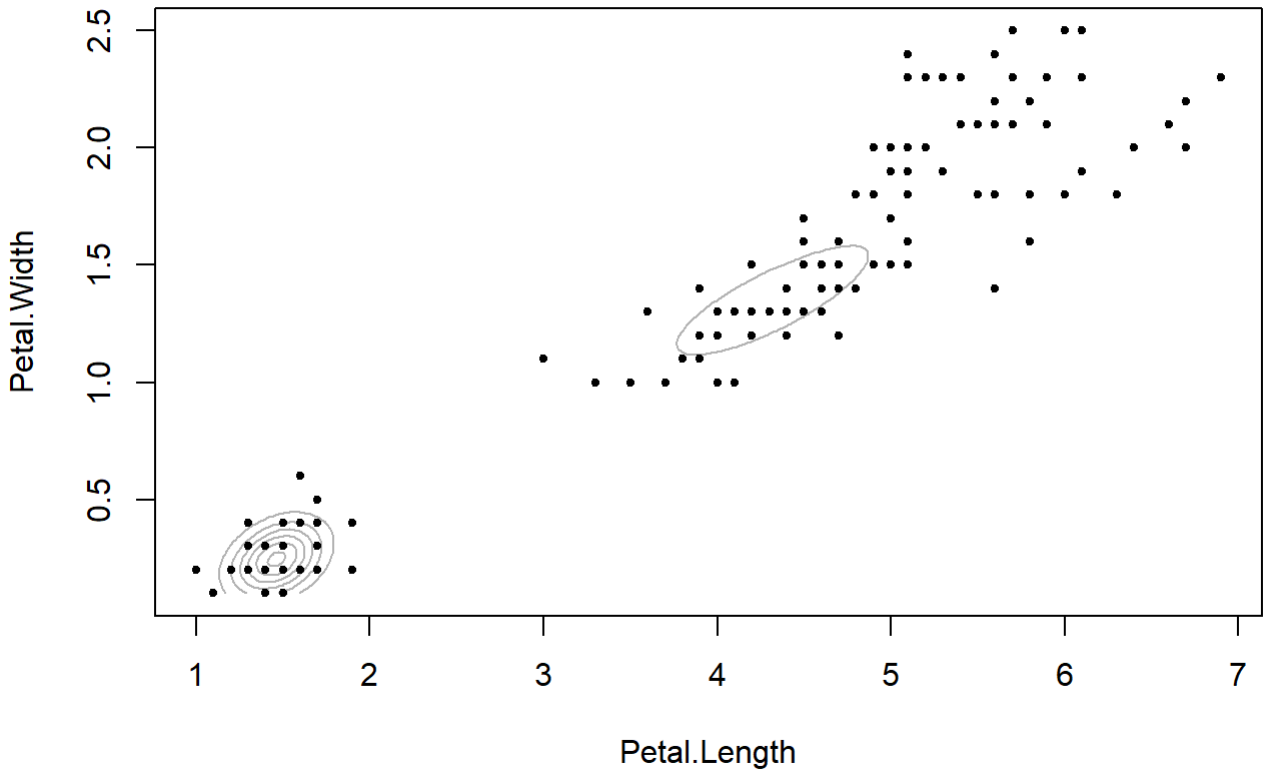

etal.Width
2.5
2.0
1.5


0.5
1
2
3
4
5
6
7
Petal.Length


Petal.Width
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7
Petal.Length




```
# Creating a density for our model
dens<-densityMclust(data,G=3,modelNames="VVV")
# ploting that density for our given values
plot(dens, what = "density", data = data,drawlabels = FALSE, points.pch = 20)
```




```
#rmarkdown::render('JRM_DM_2.Rmd', 'html_document')
```