Introduction:

Geographically Weighted Regression (GWR) is a spatial analysis that aims to identify whether spatial relationships vary according to observation location. GWR relies on location-specific, weighted measures of spatial autocorrelation based on localized regressions. While it is easy to determine whether a relationship between variables differs over space, determining if these relationships are due to error or bias proves more difficult. The function GWR_Analysis seeks to address this issue by running a series of inferential tests to identify potential threats in our regression model and to ensure the precision and accuracy of our estimated coefficients.

As GWR is based on a "global" multiple regression model, the function first determines whether the global model is statistically significant, and then whether multicollinearity and outliers are a significant threat to our model. Both multicollinearity and outliers are important to address as they have the potential to distort local estimates, which then can subsequently distort the surface estimates. The function further allows the user to establish the power of our local model by essentially estimating GWR model fit relative to the global model. Lastly, the function maps GWR coefficients alongside independent variables onto a choropleth map to visualize and identify patterns of spatial autocorrelation. This also allows users to appreciate the extent to which global patterns are determined by the localized regression. Our function is designed to work with census data, and thus strategically utilizes an adaptive kernel window to accommodate for the variable size of census zones.

Function Design:

The function has been designed to work with any shapefile and corresponding dataset, so long as the shapefile and data have been merged beforehand and the working directory has been set. This is true for any dataset which utilizes OA, LSOA, or other zone codes. After the installation of necessary packages in R, the user can input their variables of interest alongside the filename to which they wish the maps to be saved, and the function will run. The steps of the function can be followed in the flowchart below.

Essentially, the first step of the function is to test the relationship between the dependent variable and two independent variables of choice in our global model. The function has been programmed to test for significance at a p-value threshold of 0.05. Based on the results of the model, the function will either print "Significant regression model" or "Non-significant regression model" for the user. The second test run on the model looks for multicollinearity, and software from the "car" package will calculate the Variance Inflation Factor (VIF) on each independent variable to then tell the user whether there is a risk for multicollinearity. A VIF value of 1 would determine no collinearity, a value between 1 and 5 would be considered moderate, and any value above 5 would be considered a serious risk (Fotheringham, 2003). The third test utilizes the "MASS" package and runs tests on studentized residuals as an appropriate tool for detecting outliers. Studentized residual rests compare the absolute value to the threshold of three, so if points are above this, they are considered outliers (Zhang, 2016). The function will return an output of either "no significant outlier present", if the value is below 3, or the opposite if it is greater. Next, the adaptive kernel bandwidth is calculated and the GWR model is run. The function determines whether a GWR model best fits the data in comparison to the global regression model by measuring the Akaike Information Criterion (AIC) value of both. The AIC value is essentially an estimated prediction error of

the models relative to one another, where AIC decreases with model accuracy, hence why a lower AIC value is more desirable (Fotheringham, 2013) . The function returns both AIC values and will print either “GWR model is better fit according to AIC value" or the opposite for the user, based on which value is lower. It is important to highlight, however, that AIC is strictly a measure of best model fit relative to other candidate models and not a measure of whether the model describes the data accurately or not.

The final step in the function is to map the distribution of the independent variables and their GWR coefficients. The function assigns GWR coefficients to a quantile range, represented in the choropleth returning maps of the same color to maps of the same variable. This utilizes packages “tmap”, “grid”, and “gridExtra” to customize and print the maps.

Figure 1: Flowchart for Function GWR_Analysis(Dataframe, Dependent, First, Second)

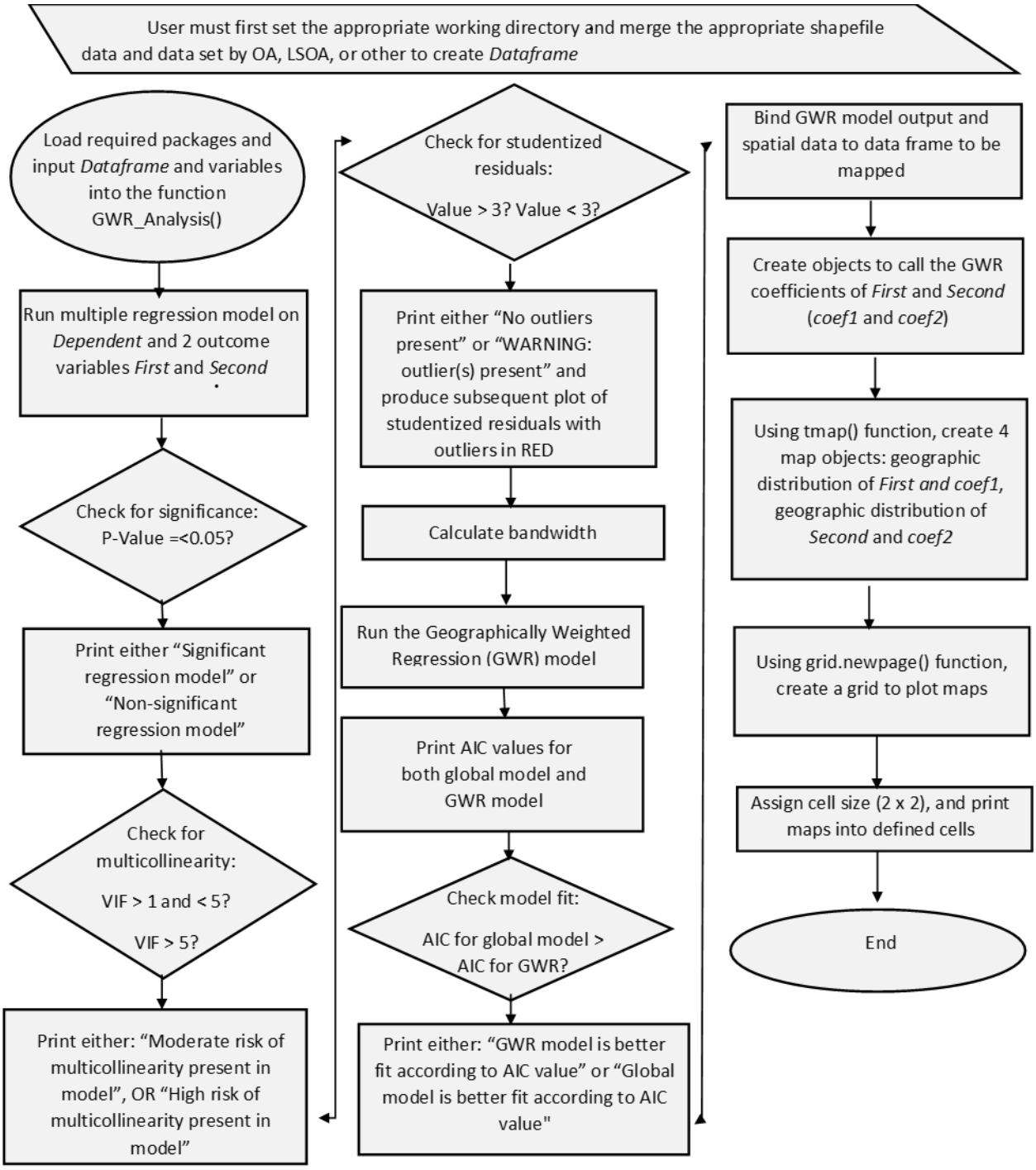
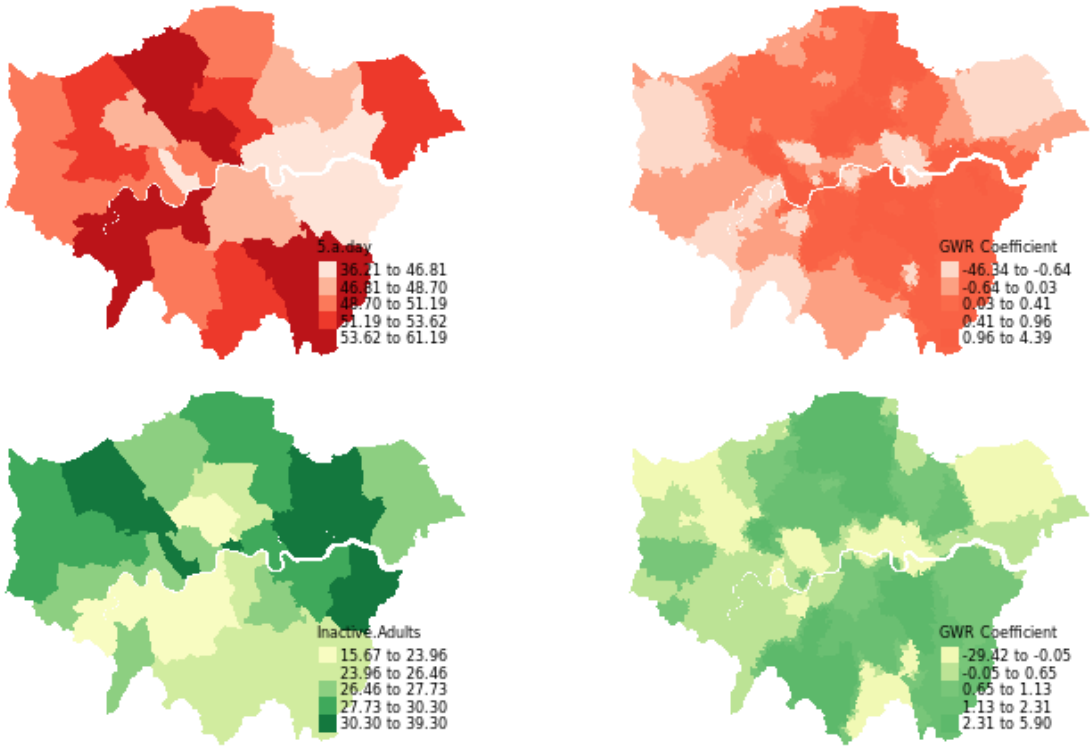


Figure 2: Explanation of Required Variables and Packages

| Required Variables<br><br>GWR_Analysis(Dataframe, Dependent, First, Second) | |
|---|---|
| Dataframe | A SpatialPolygonsDataFrame consisting of two merged datasets by the user, one SpatialPolygonsDataFrame and one dataframe which both address the same geographic area and geographic code**<br><br>**This will require manipulation by the user to call the same data, using merge() function<br><br>Example:<br>Southwark.OA.Census <-<br>merge(Southwark.Output.Areas, Southwark.Data,<br>by.x="OA11CD", by.y="OA")<br><br>Definitions:<br>~ Southwark.OA.Census = SpatialPolygonsDataFrame<br>~ Southwark.Output.Areas<br>=SpatialPolygonsDataFrame<br>~ Southwark.Data = dataframe<br>~ OA11CD = column name in SpatialPlygonsDataFrame<br>~OA = column name in dataframe with Output Area values |
| Dependent | Dependent variable (*from: Dataframe*) |
| First | First explanatory variable (*from: Dataframe*) |
| Second | Second explanatory variable (*from: Dataframe*) |
| Required Packages | |
| library("tmap") | Used to create maps<br>~ tm_shape()<br>~ tm_fill()<br>~ tm_layout() |
| library("car") | Used to find VIF value<br>~ vif() |
| library("spgwr") | Used to compute GWR<br>~ gwr.sel()<br>~ gwr() |
| library("grid")<br>library("gridExtra") | Used to create and assign grids to plot multiple map objects in one panel<br>~ grind.newpage()<br>~ pushViewport() |
| library("MASS") | Used to calculate AIC value and studentized residuals<br>~ AIC()<br>~ studres() |

Function Application:

The function is applied to answer the question: is there a relationship between eating the recommended 5 a day and inactivity on excess weight in Adults in London? How confident can we be in our model, and how does this relationship vary across the city? This data was provided by Public Health England (now known as UK Health Security Agency) from the Public Health Outcomes Framework published in November 2014 (Public Health England, 2014). The dependent variable in this situation is "Excess Weight in Adults", while our two independent variables are "5 a day" and "Inactivity". According to the dataset, excess weight in adults refers to "Percentage of adults classified as overweight or obese". Five a day refers to the "Proportion of the population who, when surveyed, reported that they had eaten the recommended 5 portions of fruit and vegetables on the previous day", and Inactivity as "The number of respondents aged 16 and over doing less than 30 'equivalent' minutes of at least moderate intensity physical activity per week in the previous 28 days expressed as a percentage of the total number of respondents aged 16 and over". The rationale for this research question is the importance of physical activity, diet, and obesity as predictors for the onset of several chronic diseases and premature deaths across the globe. This dataset uses codes Lower Layer Super Output Area (LSOA) codes, which are small areas designed to be of a similar population size, with an average of approximately 1,500 residents or 650 households




Once the data is run through the GWR_Analysis function, the output confirms that we are dealing with a "Significant regression model" and identifies that there is a "Moderate risk of multicollinearity in the model". The output also returns that there are "No outliers present", and by looking at both AIC values the user can confirm that the GWR model is a better fit than the global model. The map output explains how Excess Weight in Adults is influenced by our independent variables across London. When looking at 5 a day, areas with the strongest positive impact on Excess Weight in Adults are shaded dark red.

Inactivity in Adults also displays a positive relationship. In areas of the borough shaded dark green this relationship is the strongest, and where shades of green is lightest, Inactivity in Adults has little to no impact on Excess Weight in Adults.

The same function was applied to census data for Camden, looking at qualification as the dependent variable, and White British and unemployment as the independent. The relevant output and maps are included below, and the full code can be found in the appendix.

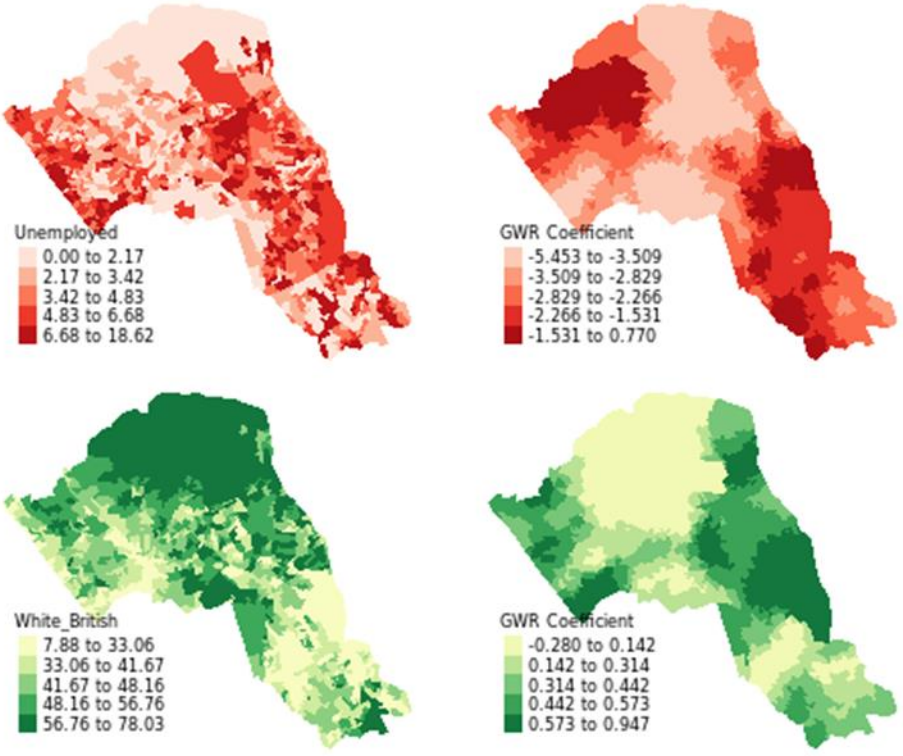
```
GWR_Analysis(OA.Census, "Qualification", First = "Unemployed", Second = "Whit
e_British")
```

```
## [1] "Significant regression model"
```

```
## [1] "Moderate risk of multicolinearity present in model"
```

```
## [1] "No outliers present"
```

```
## [1] 5936.169
## [1] 5508.777
## [1] "GWR model is better fit according to AIC value"
```




Looking at the output of the function, we are dealing with a significant regression model with moderate risk of multicollinearity and no significant outliers present. We can also be confident that the GWR model is better fit, according to AIC value. In the series of maps printed, we can predict how

qualification is influenced by our independent variables across Camden. When looking at White British, areas with the strongest positive impact on Qualification are shaded dark green. Unemployment, on the other hand, displays a negative relationship. In middle areas of the borough shaded dark red this relationship is the strongest, and where shades of red is lightest, Unemployment has little to no impact on Qualification.

Potential Applications and Limitations of function

The advantage of our function is that users are not only able to visualize and understand the correlation between a given social phenomena and its variation across space, but they can also be confident in the precision and accuracy of their models. This confidence is essential in the social science field, as policies and decisions must be made on reliable data. In the event that a model is subject to a particular threat, the function allows users to identify the specific cause of concern based on the output. From this information, the user can then make decisions on whether to include or remove certain variables, for example, or whether to address these issues as limitations in their own further analysis.

Regarding the maps, the function has pre-defined settings for color palette, legend names, and breaks. This is a potential limitation as it allows for very minimal customization on the user's behalf. The pre-defined settings were designed, however, to be compatible with all independent variables and GWR output. The assignment of breaks, color palette, and titles were pre-selected so that output is consistent and easy to interpret every time. Another potential limitation is the fact that users must merge the shapefiles themselves.

Appendix:

```
GWR_Analysis <- function(Dataframe, Dependent, First, Second) {

Dep <- as.data.frame(Dataframe)[, Dependent]

variable_1 <- as.data.frame(Dataframe)[, First]
variable_2 <- as.data.frame(Dataframe)[, Second]

#run the global model in a multiple regression
global_model <- lm(Dep ~ variable_1 + variable_2)

#check p-value to determine model's significance
 if (coef(summary(global_model))[,"Pr(>|t|)"] < 0.05) {print("Significant reg
ression model")
  } else {print("Non-significant regression model") }



object<-vif(global_model)

#check for multicolinearity in the model
  if (object > 1 & object < 5) {print("Moderate risk of multicolinearity pres
ent in model")
  } else {print("High risk of multicolinearity present in model") }


#check for studentised residuals,
studresiduals<-studres(global_model)
DF <- as.data.frame(studresiduals)

#if studentised residuals are > 3, plot outliers

  if (studresiduals > 3) {print("WARNING: outlier(s) present")


        DF$Resid<-studres(global_model)

        DF$Outs<-ifelse(abs(DF$Resid)>3, 1, 0)

        plot(DF$Resid, col=DF$Outs+1, pch=16,ylim=c(-3,3))

        abline(0,0)

        DF2<-DF[!DF$Outs,]

        nrow(DF2)

        plot(DF2$Resid, col=DF2$Outs+1,pch=16, ylim=c(-3,3))
```

```
        abline(0,0)


  } else {print("No outliers present") }


#calculate bandwidth
GWRbandwidth <- gwr.sel(Dep ~ variable_1 + variable_2, data=Dataframe,adapt=T
)

#set gwr model
gwr.model = gwr(Dep ~ variable_1 + variable_2, data = Dataframe, adapt=GWRban
dwidth, hatmatrix=TRUE, se.fit=TRUE)


#print the AIC value for the global model and our local gwr model and determi
ne which is the best fit
global_model_AIC <- AIC(global_model)
GWR_model_AIC <- gwr.model[["results"]][["AICh"]]

print(AIC(global_model))
print(gwr.model[["results"]][["AICh"]])

if (global_model_AIC > GWR_model_AIC) {print("GWR model is better fit accordi
ng to AIC value")
  } else {print("Global model is better fit according to AIC value") }

#bind gwr model outputs to our OA.Census polygon so we can map
results <- as.data.frame(gwr.model$SDF)

#call the coefficients of our variables of interest
coef1<- results[[3]]
Dataframe@data$coef1 <- coef1
coef2 <- results[[4]]
Dataframe@data$coef2 <- coef2


#assign gwr.map (a non-spatial object) to OA.Census (a spatial object, so we
can map it)
gwr.map <- Dataframe


#bind spatial data to the matrix results
gwr.map@data <- cbind(Dataframe@data, as.matrix(results))


#create tmap objects
map1 <- tm_shape(gwr.map) + tm_fill(First, palette = "Reds", n = 5, style = "
quantile")  + tm_layout(frame = FALSE, legend.text.size = 0.5, legend.title.s
ize = 0.6)
```

```
map2 <- tm_shape(gwr.map) + tm_fill("coef1", midpoint = NA, palette = "Reds",
n = 5, style = "quantile", title = "GWR Coefficient") + tm_layout(frame = FAL
SE, legend.text.size = 0.5, legend.title.size = 0.6)
map3 <- tm_shape(gwr.map) + tm_fill(Second, palette = "YlGn", n = 5, style =
"quantile") + tm_layout(frame = FALSE, legend.text.size = 0.5, legend.title.s
ize = 0.6)
map4 <- tm_shape(gwr.map) + tm_fill("coef2", midpoint = NA, palette = "YlGn",
n = 5, style = "quantile", title = "GWR Coefficient") + tm_layout(frame = FAL
SE, legend.text.size = 0.5, legend.title.size = 0.6)


#create a clear grid to plot maps
grid.newpage()


# assigns the cell size of the grid, in this case 2 by 2
pushViewport(viewport(layout=grid.layout(2,2)))



#print maps into defined cells
print(map1, vp=viewport(layout.pos.col = 1, layout.pos.row =1))
print(map2, vp=viewport(layout.pos.col = 2, layout.pos.row =1))
print(map3, vp=viewport(layout.pos.col = 1, layout.pos.row =2))
print(map4, vp=viewport(layout.pos.col = 2, layout.pos.row =2))


}
```

References:

Fotheringham, A.S., Brunsdon, C. and Charlton, M., 2003. *Geographically weighted regression: the analysis of spatially varying relationships*. John Wiley & Sons.

Zhang, Z., 2016. Residuals and regression diagnostics: focusing on logistic regression. *Annals of translational medicine*, *4*(10).

Public Health England. 2014. Public Health Outcomes Framework Indicators. [Online]. [Accessed 20 November 2022]. Available from: https://data.london.gov.uk/dataset/public-health-outcomes-framework-indicators