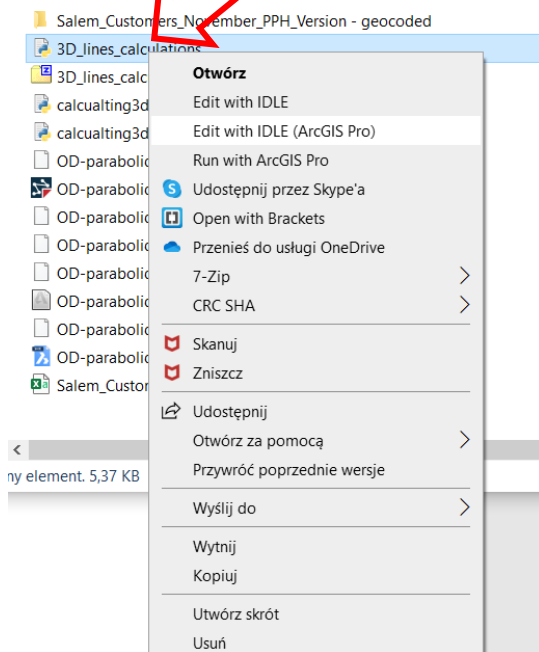
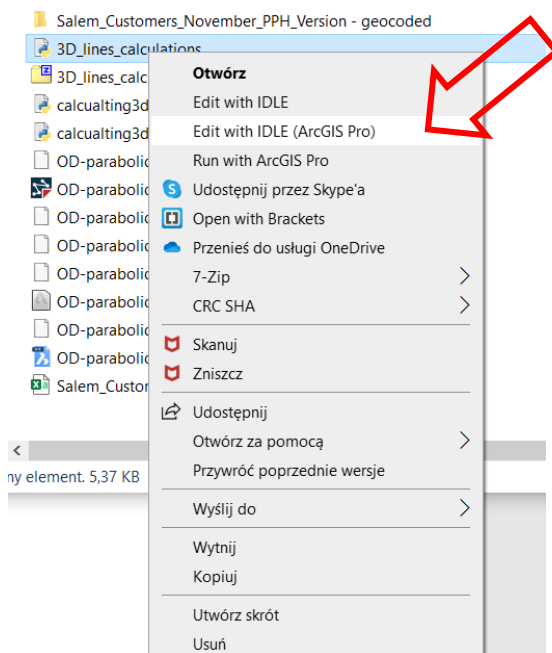


1. Right click on script file



2. On the list click Edit with IDLE (ArcGIS Pro)



### 3. The python script will be opened in IDLE

```
import os, arcpy

## INPUTS - part 1
customerPoints = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Customers') ## paste path for customer feature class or shapefile
storePoint = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Store_location') ## paste path for customer feature class or shapefile
folder_for_parabolicLines = os.path.join(r'D:\Origin-Destination Lines\New folder') ## paste path for folder where 3D lines should be saved

## INPUTS - part 2
parabolicMaxElevation_as_fixed_elevation = 5000 ## for defining fixed elevation type number in meters
parabolicMaxElevation_as_line_length_in_percentage= 0 ## for defining elevation by percentage of line length type number here and set parabolicMaxElevation_as_fixed_elevation to zero (

##-----##
## Reading store coordinates
idx_store_sc = 0
spatial_reference = arcpy.SpatialReference(4326)
with arcpy.da.SearchCursor(storePoint, ['SHAPE@']) as store_sc:
    for store_row in store_sc:
        if idx_store_sc < 1:
            point_geometry = store_row[0]
            point_WGS84 = point_geometry.projectAs(spatial_reference)
            idx_store_sc += 1
        else:
            break

## Creating parabolic lines feature class
parabolicLinesName = 'parabolicLines.shp'
parabolicLines = os.path.join(folder_for_parabolicLines, parabolicLinesName)

idx = 1
while os.path.exists(parabolicLines):
    parabolicLinesName = 'parabolicLines_{x}.shp'.format(x = idx)
    parabolicLines = os.path.join(folder_for_parabolicLines, 'parabolicLines_{x}.shp'.format(x = idx))
    idx +=1
```

### 4. There are 3 path to be set

customerPoints - path for customer points shapefile or feature class  
storePoint - path for store point shapefile or feature class  
folder\_for\_parabolicLines - path to the folder where result 3D lines will be saved.

```
import os, arcpy

## INPUTS - part 1
customerPoints = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Customers') ## paste path for customer feature class or shapefile
storePoint = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Store_location') ## paste path for customer feature class or shapefile
folder_for_parabolicLines = os.path.join(r'D:\Origin-Destination Lines\New folder') ## paste path for folder where 3D lines should be saved

## INPUTS - part 2
parabolicMaxElevation_as_fixed_elevation = 5000 ## for defining fixed elevation type number in meters
parabolicMaxElevation_as_line_length_in_percentage= 0 ## for defining elevation by percentage of line length type number here and set parabolicMaxElevation_as_fixed_elevation to zero (

##-----##
## Reading store coordinates
idx_store_sc = 0
spatial_reference = arcpy.SpatialReference(4326)
with arcpy.da.SearchCursor(storePoint, ['SHAPE@']) as store_sc:
    for store_row in store_sc:
        if idx_store_sc < 1:
            point_geometry = store_row[0]
            point_WGS84 = point_geometry.projectAs(spatial_reference)
            idx_store_sc += 1
        else:
            break

## Creating parabolic lines feature class
parabolicLinesName = 'parabolicLines.shp'
parabolicLines = os.path.join(folder_for_parabolicLines, parabolicLinesName)

idx = 1
while os.path.exists(parabolicLines):
    parabolicLinesName = 'parabolicLines_{x}.shp'.format(x = idx)
    parabolicLines = os.path.join(folder_for_parabolicLines, 'parabolicLines_{x}.shp'.format(x = idx))
    idx +=1
```

Replace the path that are currently typed in the script with path on you computer. Please remember to replac the path with valid path on you computer also please place new path between those single quotation marks.

## 5. There is one more parameter for max line elevation

```
import os, arcpy

## INPUTS - part 1
customerPoints = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Customers') ## paste path for customer feature class or shapefile
storePoint = os.path.join(r'D:\Origin-Destination Lines\Origin-Destination Lines.gdb\Store_location') ## paste path for customer feature class or shapefile
folder_for_parabolicLines = os.path.join(r'D:\Origin-Destination Lines\New folder') ## paste path for folder where 3D lines should be saved

## INPUTS - part 2
parabolicMaxElevation_as_fixed_elevation = 5000 ## for defining fixed elevation type number in meters
parabolicMaxElevation_as_line_length_in_percentage = 0 ## for defining elevation by percentage of line length type number here and set parabolicMaxElevation_as_fixed_elevation to zero (

##-----##
## Reading store coordinates
idx_store_sc = 0
spatial_reference = arcpy.SpatialReference(4326)
with arcpy.da.SearchCursor(storePoint, ['SHAPE@']) as store_sc:
    for store_row in store_sc:
        if idx_store_sc < 1:
            point_geometry = store_row[0]
            point_WGS84 = point_geometry.projectAs(spatial_reference)
            idx_store_sc += 1
        else:
            break

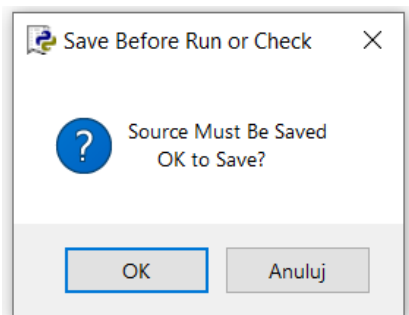
## Creating parabolic lines feature class
parabolicLinesName = 'parabolicLines.shp'
parabolicLines = os.path.join(folder_for_parabolicLines, parabolicLinesName)

idx = 1
while os.path.exists(parabolicLines):
    parabolicLinesName = 'parabolicLines_{x}.shp'.format(x = idx)
    parabolicLines = os.path.join(folder_for_parabolicLines, 'parabolicLines_{x}.shp'.format(x = idx))
    idx +=1
```

Currently it is 5000 meters but you can change it but keep in mind those are meters

## 6. OK and now to run the script click F5 bottom

The will be question if to save script, click OK



If all parameter have been set properly script will process data and will create and save shapefile in the folder you specified as one of parameters, it can take few minutes to finish. There will some calculation visible in python window.