FIT3179 DATA VISUALISATION

Tutorial Week 9

Maps with Custom Projections in Tableau

Semester 2, 2019

Overview

In this week's tutorial, you will do the following activities:

1. Overview of Spatial Data
2. Transforming between Projections with QGIS
3. Blending with Tabular Data

Overview of Spatial Data

a. GeoJSON format

One of the most popular spatial data formats is GeoJSON. GeoJSON is a JSON text file containing specialised attributes for geospatial data storage. The GeoJSON standard format is specified under IETF 's (Internet Engineering Task Force) RFC 7946. The GeoJSON format consists of several geometry objects (figure 1).

- **Point**: a single geospatial position.
- **LineString:** is an array of two or more positions.
- **Polygon:** an array of linear ring coordinate arrays.
- **MultiPoint:** an array of positions.
- **MultiLineString**: an array of LineString coordinate arrays.
- **MultiPolygon**: an array of Polygon coordinate arrays.
- **GeometryCollections**: collection of geometry objects.

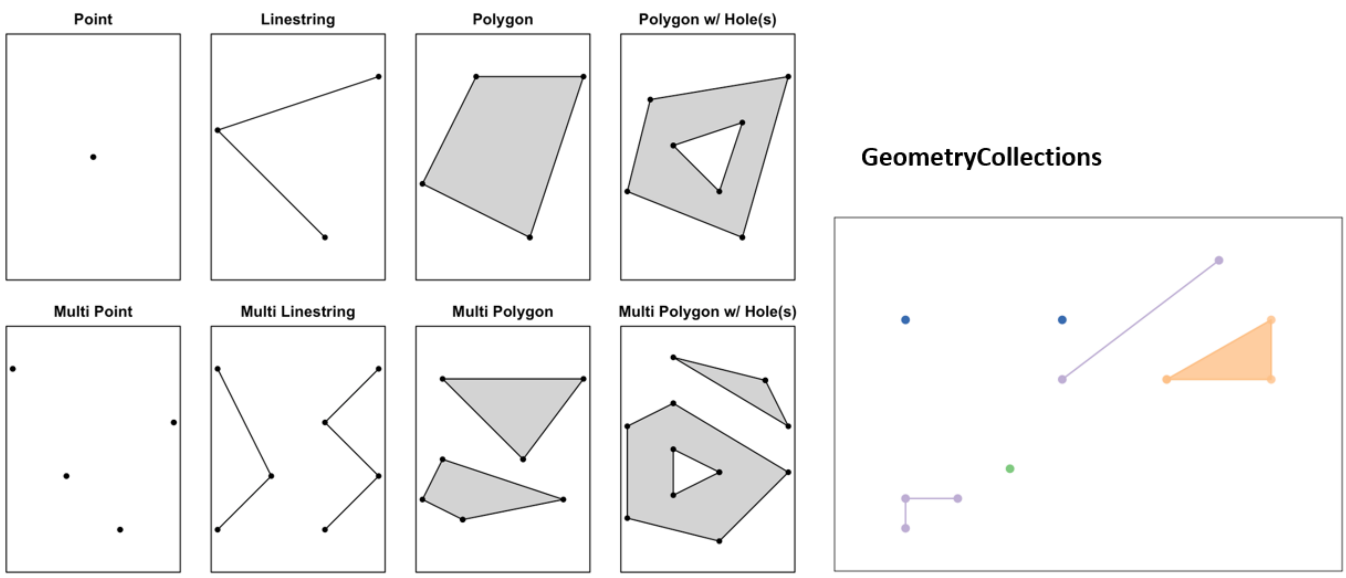


Figure 1. Geometry objects. Source.

b. Shapefile

Shapefile is another spatial data format. This data format is developed and regulated by Esri, the leading company in Geographic Information System software. ArcGIS (a well-know Geographic Information System) is one of their products. The shapefile format is actually a group of files stored in a single directory. These files are **.shp, .shx,** and **.dbf**.

- .**shp**: contains geometry objects.
- .**shx**: contains index.
- .**dbf**: contains attribute of dBase format.

The shape file format also has other files such as **.prj, .cpg,** or **.qpg**.

There are also other spatial data types supported by Tableau. We only briefly introduce you to GeoJSON and shapefile because we will be using those two file types in this tutorial.

Transforming Between Projections with QGIS

Visualising geospatial data with Tableau is generally easy to do. Tableau has basic mappings functionalities to create a wide range of geovisualisation idioms.

In the lecture, you have learnt basic principles of map projections. The goal of this tutorial is to create a map with a custom projection. Unfortunately, Tableau does not support map projection transformations. Thus, we need external software. We will use QGIS, which is a multi-platform open-source geographic information system.

QGIS is installed on the lab computers. If you want to use QGIS on your own device, download and install it from the official website. It could take a while; be patient.

a. Obtaining GeoJSON Data

The first step is obtaining the GeoJSON data. There are many online sources available to download GeoJSON files for free. The spatial data that can be used for your project is not limited to GeoJSON because QGIS supports a large variety of spatial data formats, including CSV and shapefile. In this tutorial, we will be focusing on the GeoJSON and shapefile formats.

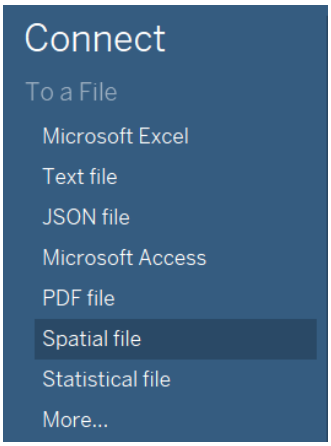
As a starting point, we will extract a GeoJSON file of a specific area using https://geojson-maps.ash.ms.

1. Go to the website. Select high resolution ("110 metre, low resolution") and check North and South America. (Note: This website uses data from https://www.naturalearthdata.com, which provides additional data layers. "110 metre" is an apparent misunderstanding of the authors of this website, because the data is optimised for a map at a scale of 1:110 million.)

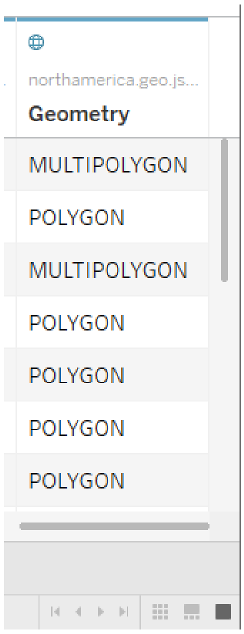
2. Click Build Custom GeoJSON. It will generate and download a **custom.geo.json** file.
3. To make it easy to remember, let’s rename that file to **america.geo.json**.
4. That’s it; you now have the GeoJSON file of the two American continents. Next we will try to see how it looks in Tableau.

b. Show it in Tableau

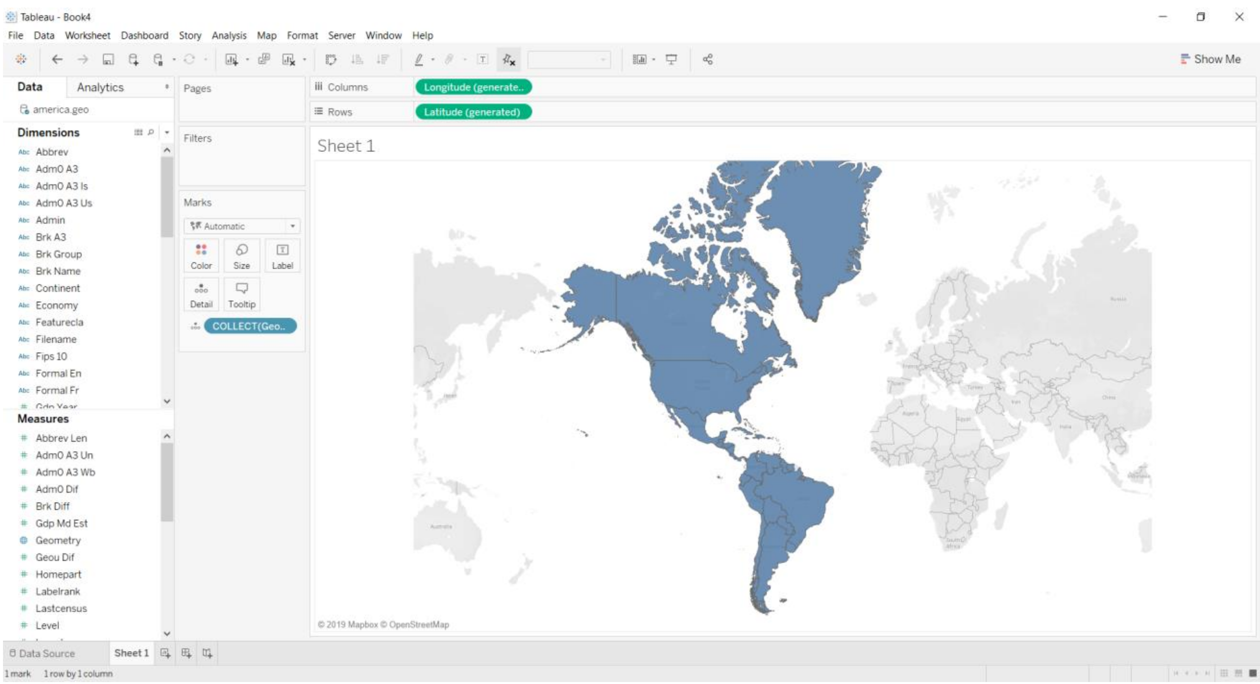
1. Open Tableau and import the GeoJSON file. Use **Spatial file connection**.



2. Once you successfully imported the GeoJSON file, you will see a Geometry column on the table. Look at the values and relate to what we discussed in the overview to geospatial data section.



3. Create a new sheet and double click on Geometry to show the Americas on a map.
   You can see that other countries are still visible on the map. You may or may not want that to happen. If you do not need them, we can remove those countries later.
   By default, Tableau gets maps data from [OpenStreetMap](https://www.openstreetmap.org) using [Mapbox's](https://www.mapbox.com) API. You can see the labels on the bottom left corner of the map.



Before moving on to the next step, let's see if you can answer these following questions.

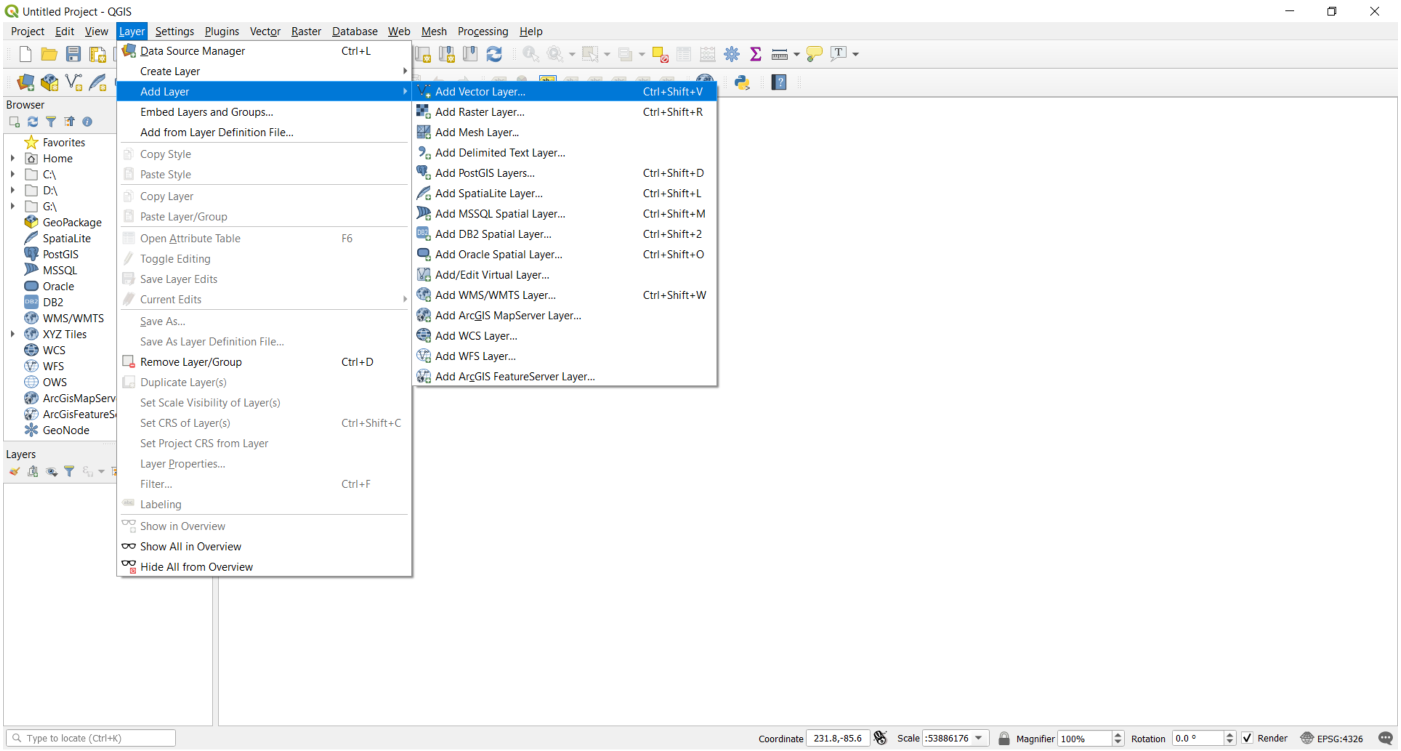
Question
What is the projection used on the map you just created? **What is that projection useful for?** **What is the drawback of that projection?**

Once you answered those questions, we can move on to the next step.

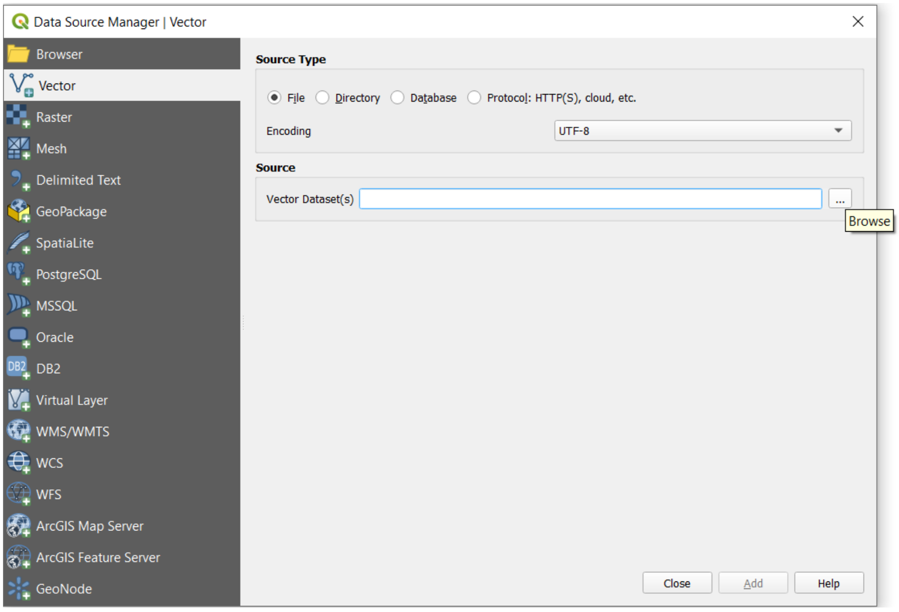
c. Importing and exporting data in QGIS

It is time to use QGIS. All steps in this part are very important. Please follow them carefully.

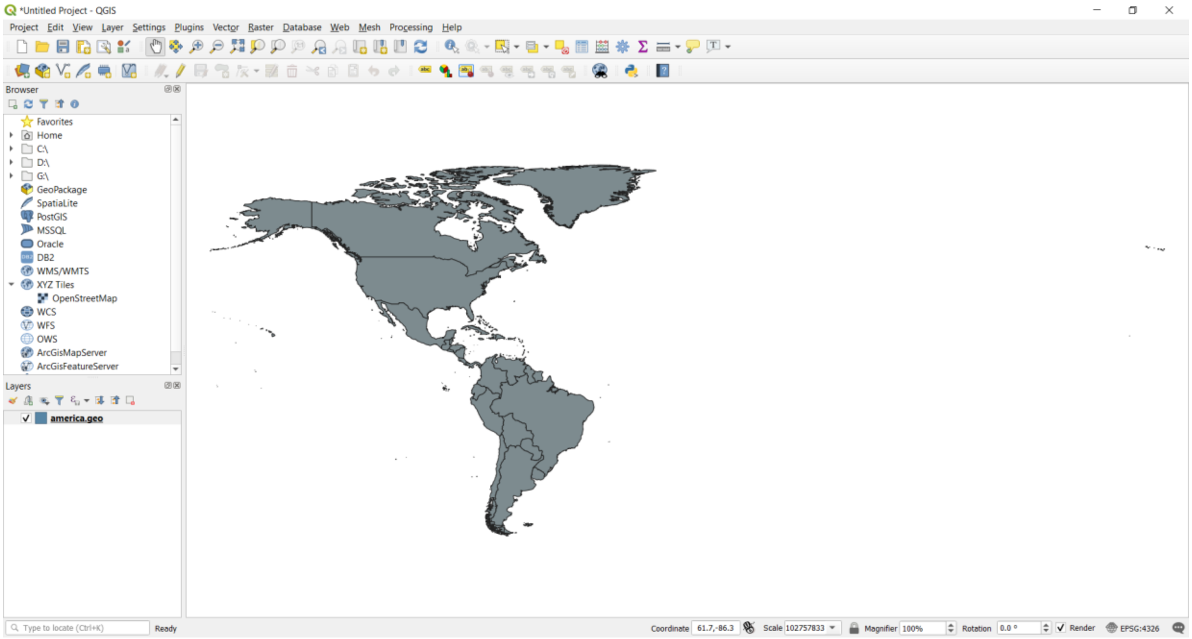
1. Open QGIS.
2. First, let's import the america.geo.json file. Click Layer → Add Layer → Add Vector Layer.



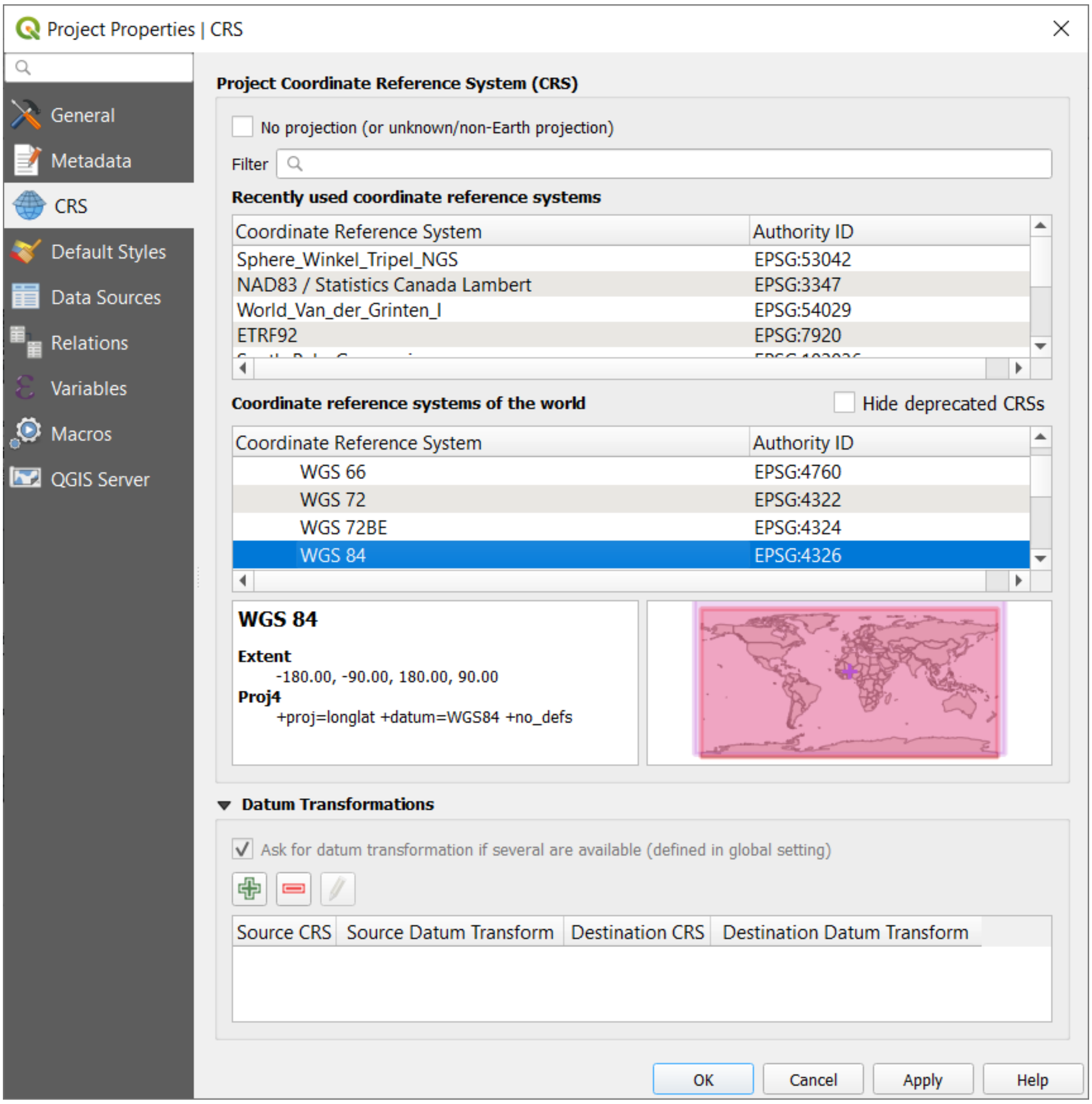
3. Click browse button to locate the file. Then click Add.



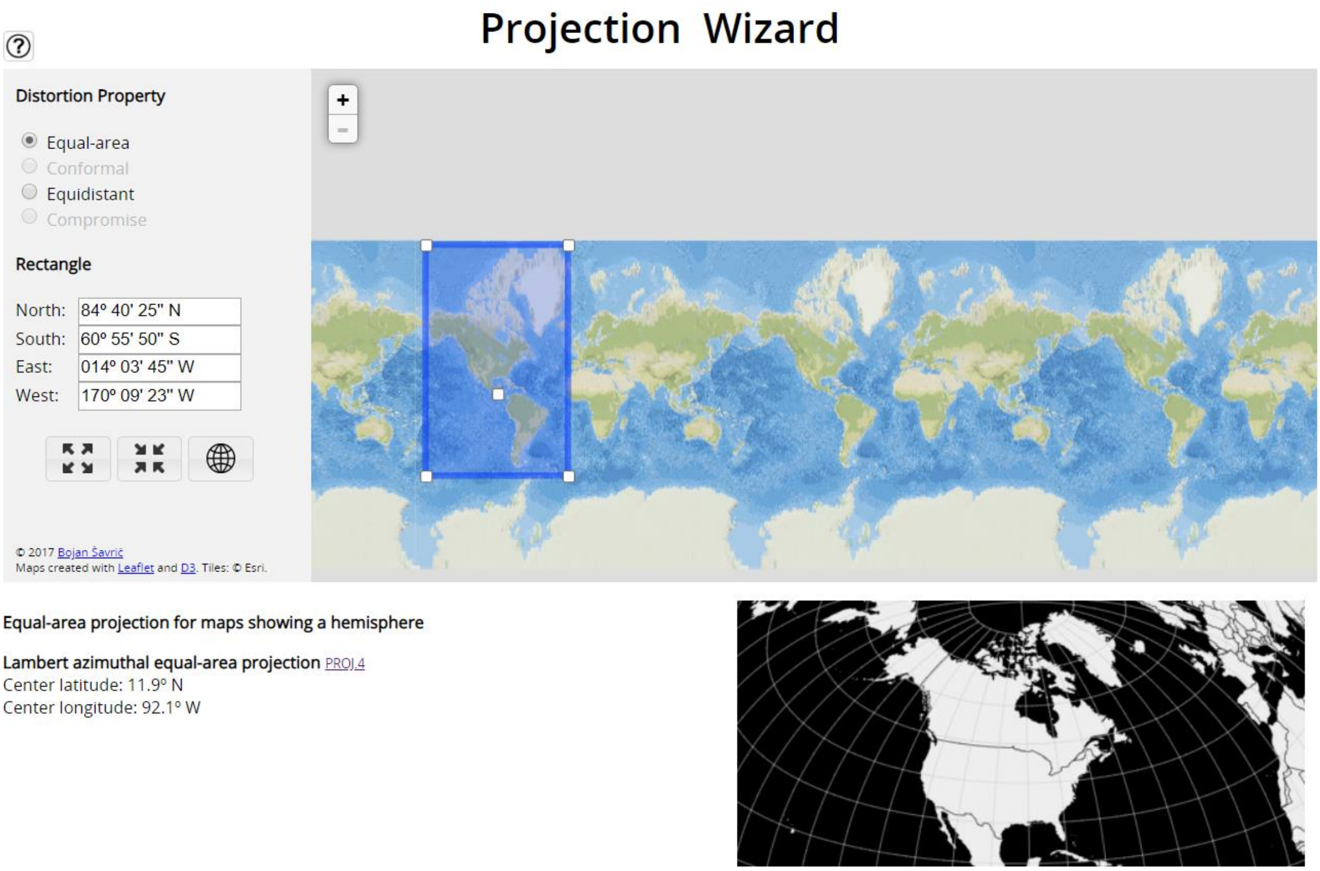
4. Once the GeoJSON file is imported successfully, you should see a map of the Americas.



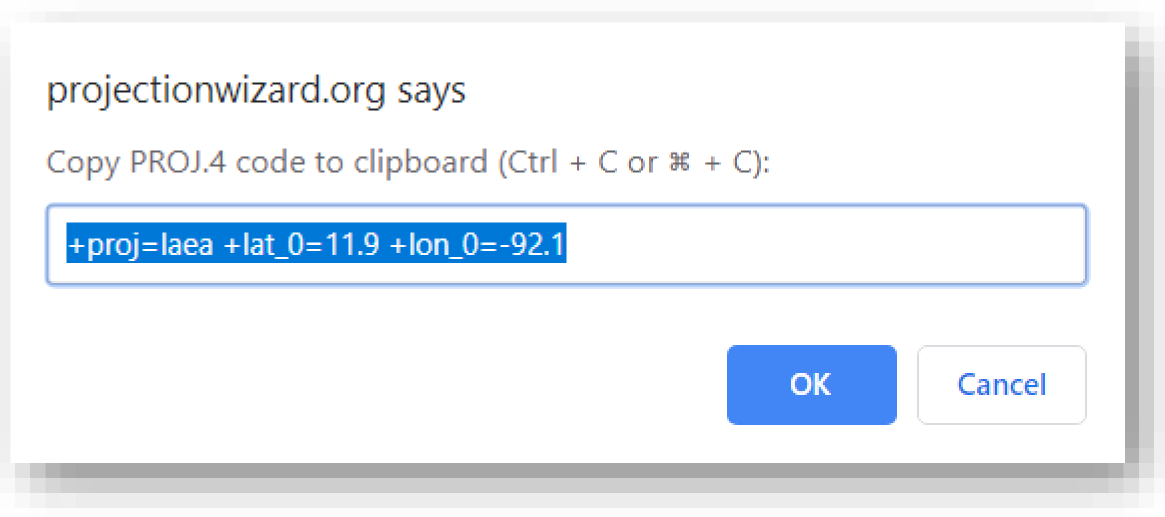
5. It looks different from the map on Tableau, doesn't it? If you want to know why, we can check the current map projection by clicking **Project → Properties**. You should see the Coordinate Reference System (which includes a projection) is set to WGS 84, which is different from Web Mercator used in OpenStreetMap.
   Ok, let's not change any setting and leave it as it is for now. Press Cancel.



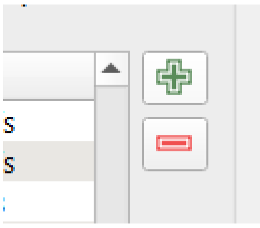
6. Now we want to change the projection of the American continents. The question is, what is a good projection for showing North and South America? To find out the answer, we will be using https://projectionwizard.org/, an online tool that can give you suggestion of appropriate projections for a specific area.
7. Go to that website and select the Americas by adjusting the blue rectangle.
8. At the bottom, you can see the suggested projection and how the result will look like. In this case, the suggested projection is the Lambert azimuthal equal-area projection.



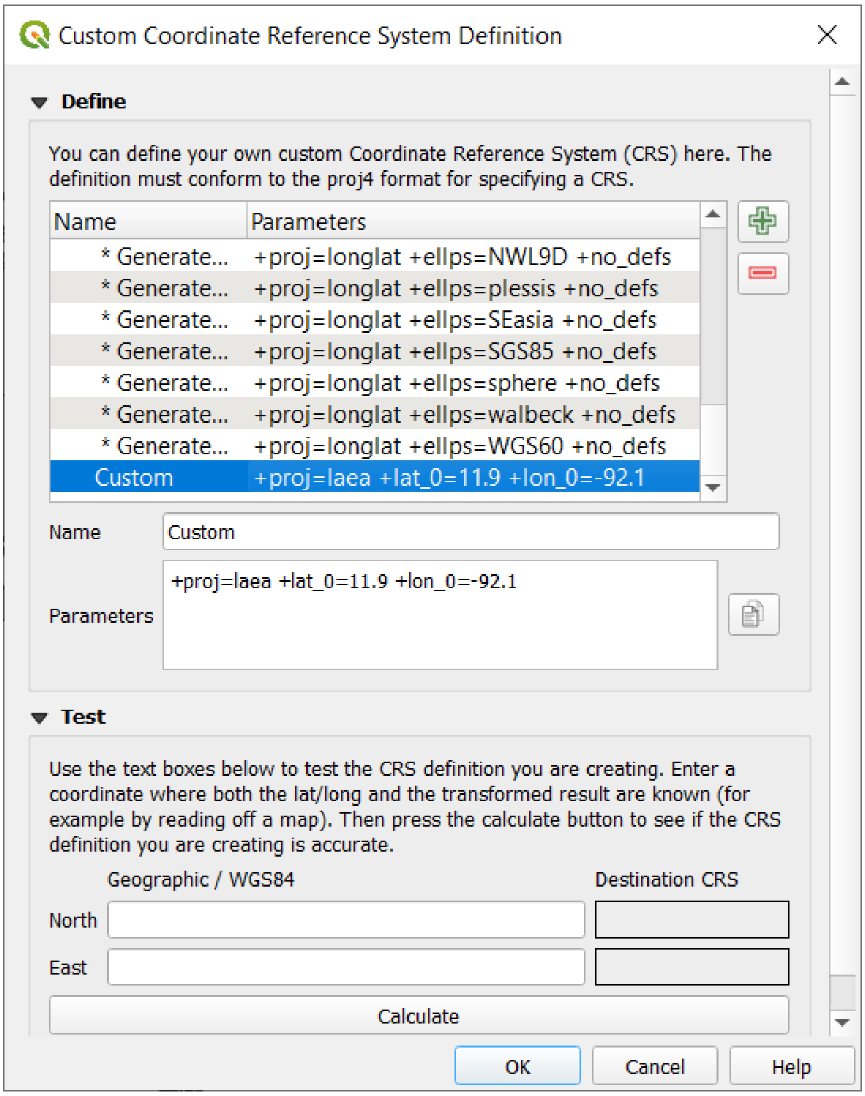
9. We need to get the projection parameters and import it to QGIS. To do that, click the **PROJ4** link. It should pop out a text dialog. Copy it (yours will be slightly different, which is okay).



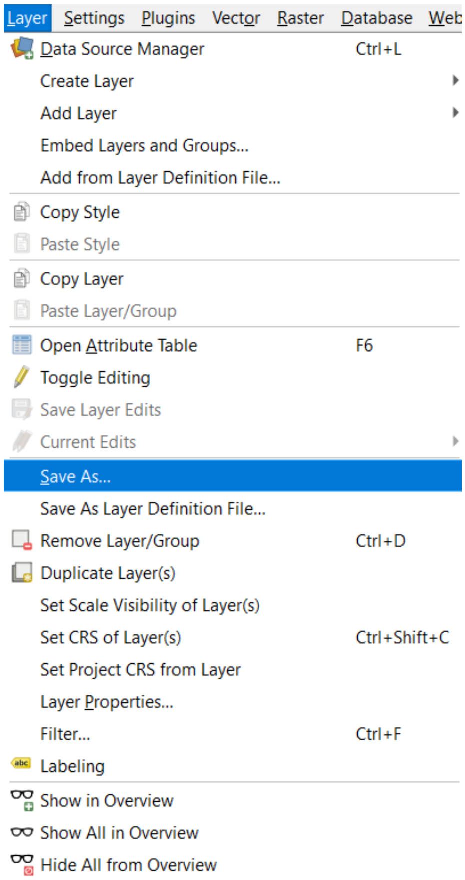
10. Now back to QGIS.
11. Now we will use the copied parameters to create a custom projection. In QGIS, click **Settings → Custom Projection**.
12. Click the plus icon to add a new projection.



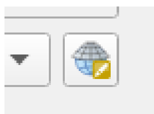
13. Add name and parameters that you copied from projectionwizard.org. Name the projection as Custom (the naming is arbitrary).



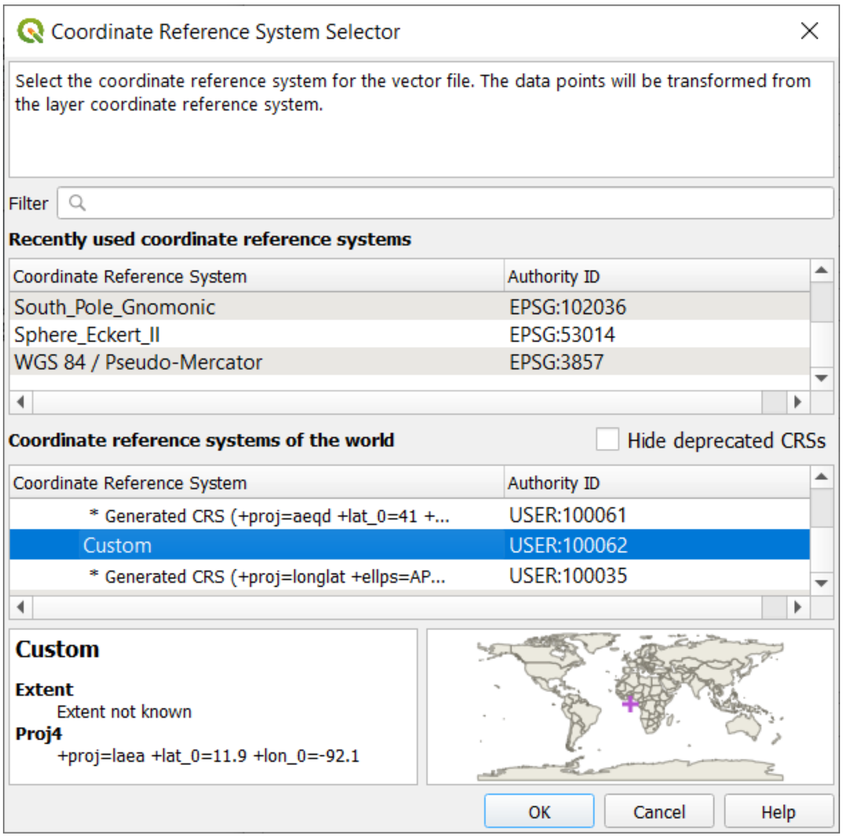
14. If you do not see the effect on the map right now, it is expected. Let's not worry too much about it now and export the layer. Click **Layer → Save as**.



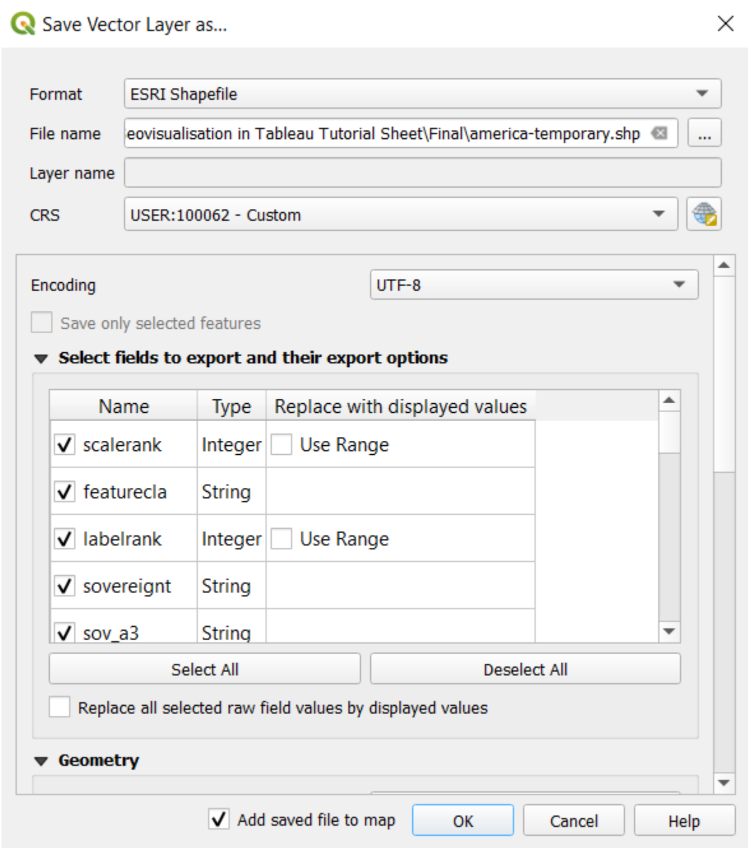
15. The setting window should appear. Do not export just yet. We need to change the Coordinate Reference System (CRS). Click the CRS button to change the projection.



16. Select the projection we just created: **Custom**. You need to scroll down in the Coordinate Reference System table. Your custom projection is located under **User Defined Coordinate Systems.** Once you found it, select the projection and click OK.



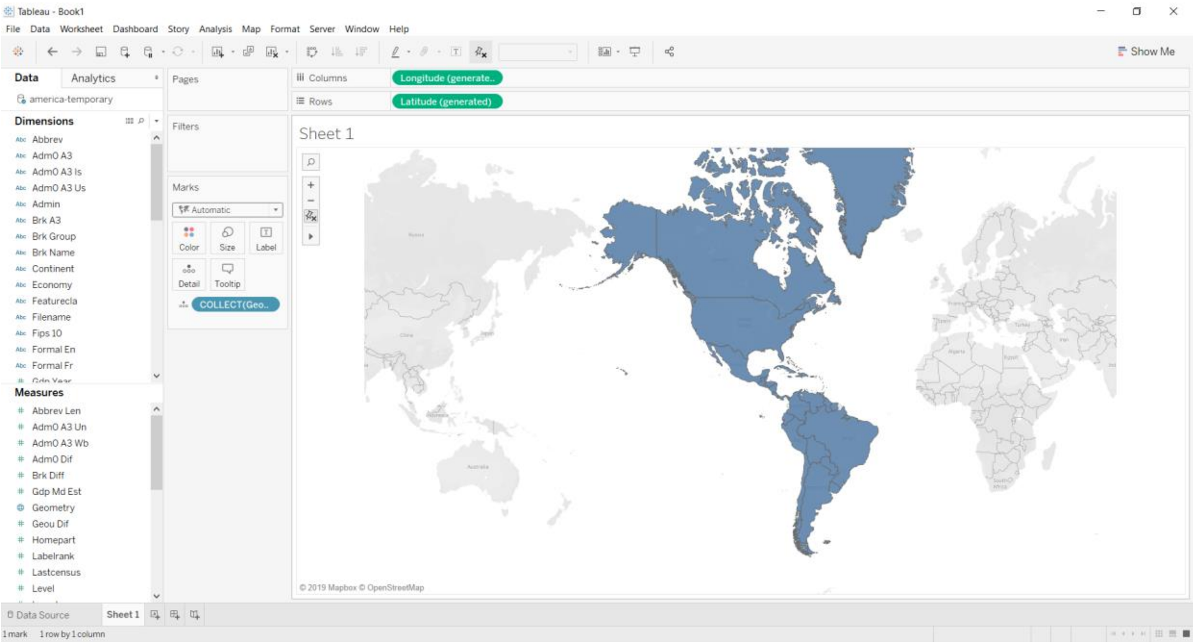
17. Set Format to **ESRI Shapefile**. Do not forget to set the name too. Let's set the filename to **america-temporary**. Why temporary? You will find out the later. Remember the file location because we need it later.



18. When you finished setting up the **Format**, **Filename**, and **CRS**, click OK to export the shapefile.
19. Once the file is successfully exported, close the project by clicking **Project** → **Close**.

d. Import the Shapefile into Tableau

Go back to Tableau and import the shapefile. You will notice that it is still using Mercator projection.



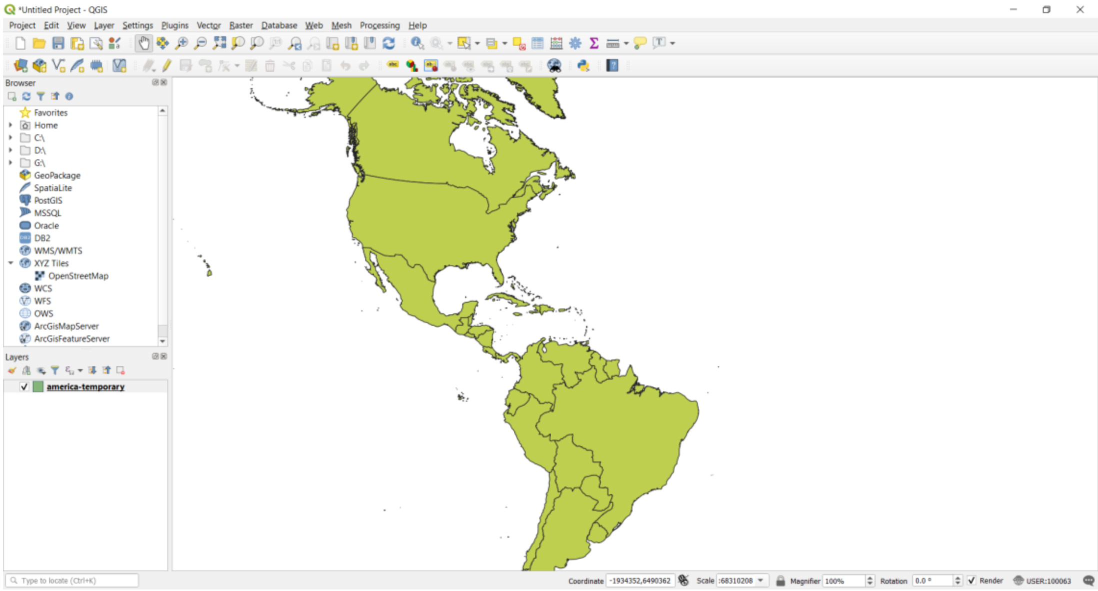
This happens because Tableau does not support other projections than Web Mercator. Is there any workaround? Yes, thanks to Sarah Battersby's post in the Tableau community forum, we know that we can "cheat" Tableau to think that our custom projection is a standard Mercator projection.

Please note that this is an obscure hack and not an elegant way, but it works for our purpose. Follow the next steps carefully to get the intended result.

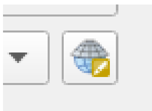
e. Cheating the Projection in QGIS

Go back to QGIS.

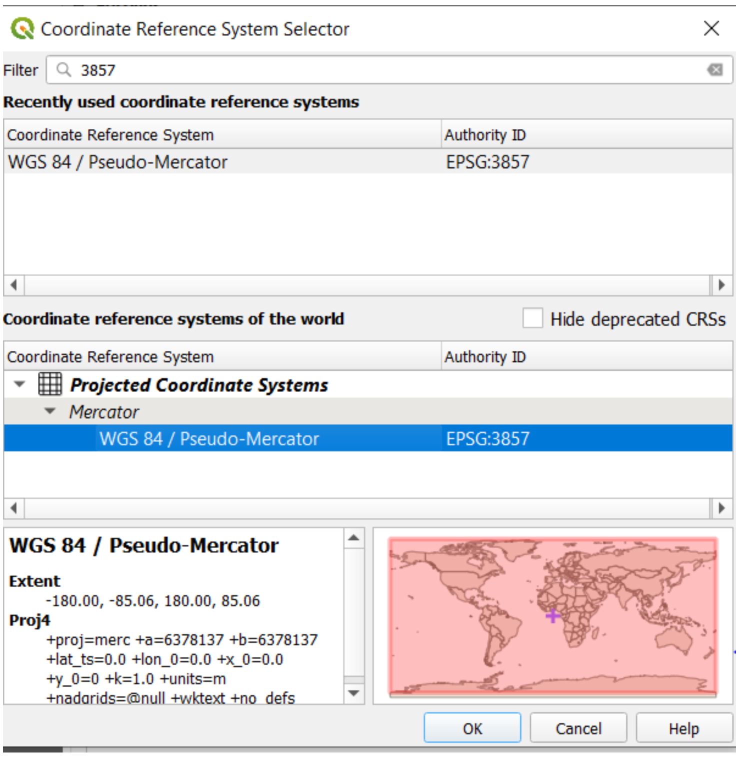
1. In a new project, import the **america-temporary** shapefile that we just created. Click **Layer → Add Layer → Add Vector Layer**.
2. Once QGIS imported the shapefile successfully, you should see the America continent in the correct projection.



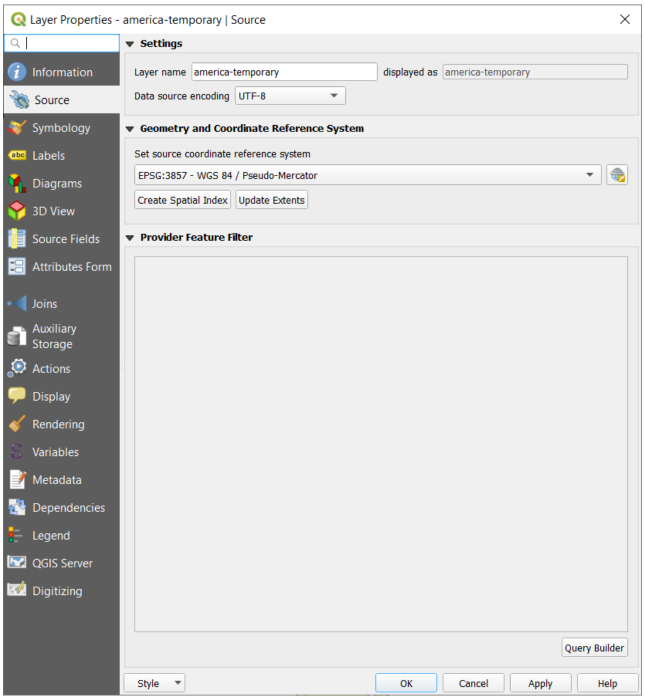
3. Here is where the critical step is taking place. Now we want to change the layer CRS (Coordinate Reference System) to the Pseudo-Mercator so that Tableau will recognise it as the standard projection. This setting does not change the original Custom projection that we had. Pseudo-Mercator is a variation of the Mercator projection used by Google Maps, Tableau and most other web mapping applications.
4. To do it, select america-temporary layer then right-click **Layer** → **Layer Properties**.
5. On the Source tab, click again the select CRS button.



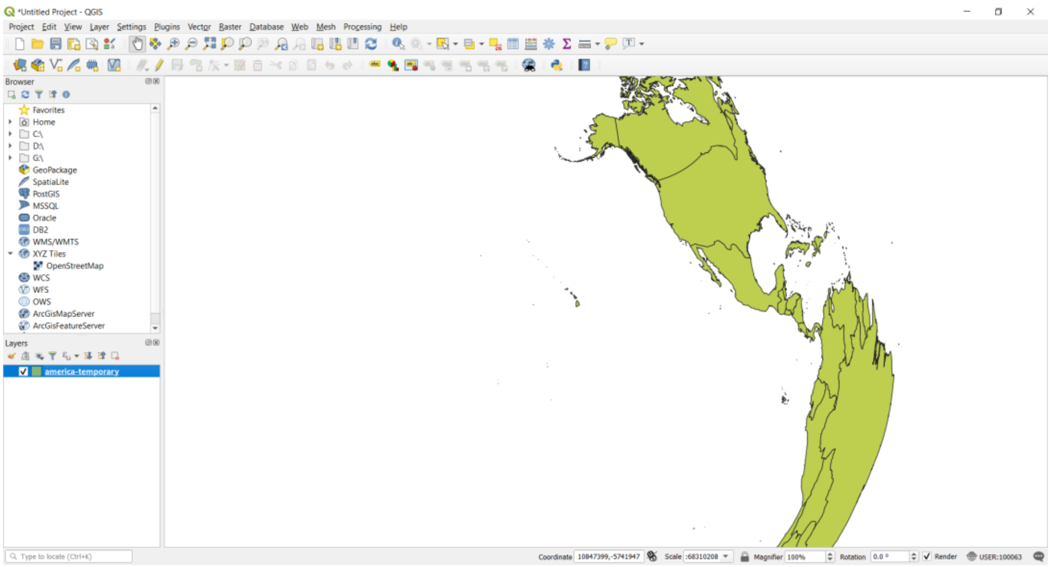
6. Search the Pseudo-Mercator by typing **3857** in **Filter** text field. Select WGS 84 / Pseudo-Mercator.



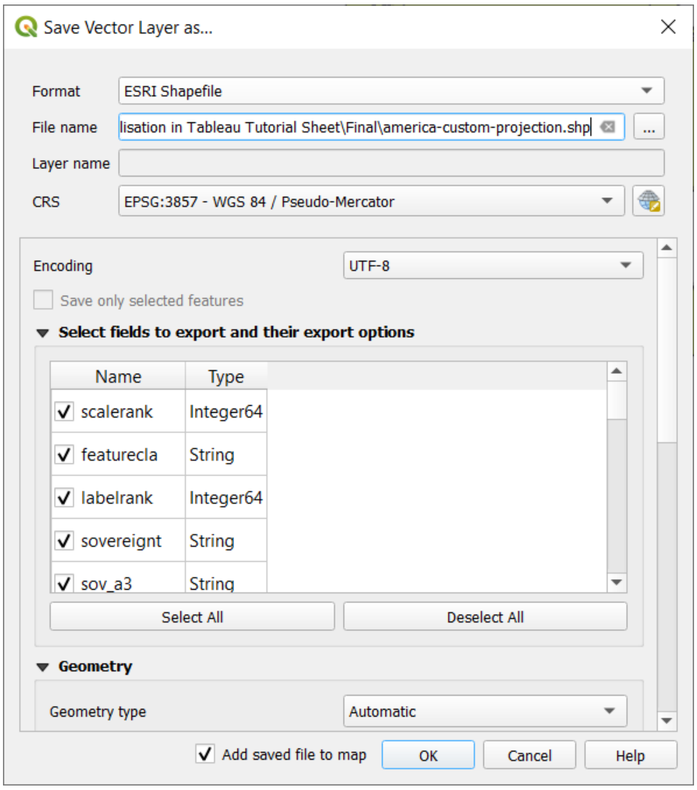
7. Here is how the layer properties look like once you have changed the coordinate reference system.



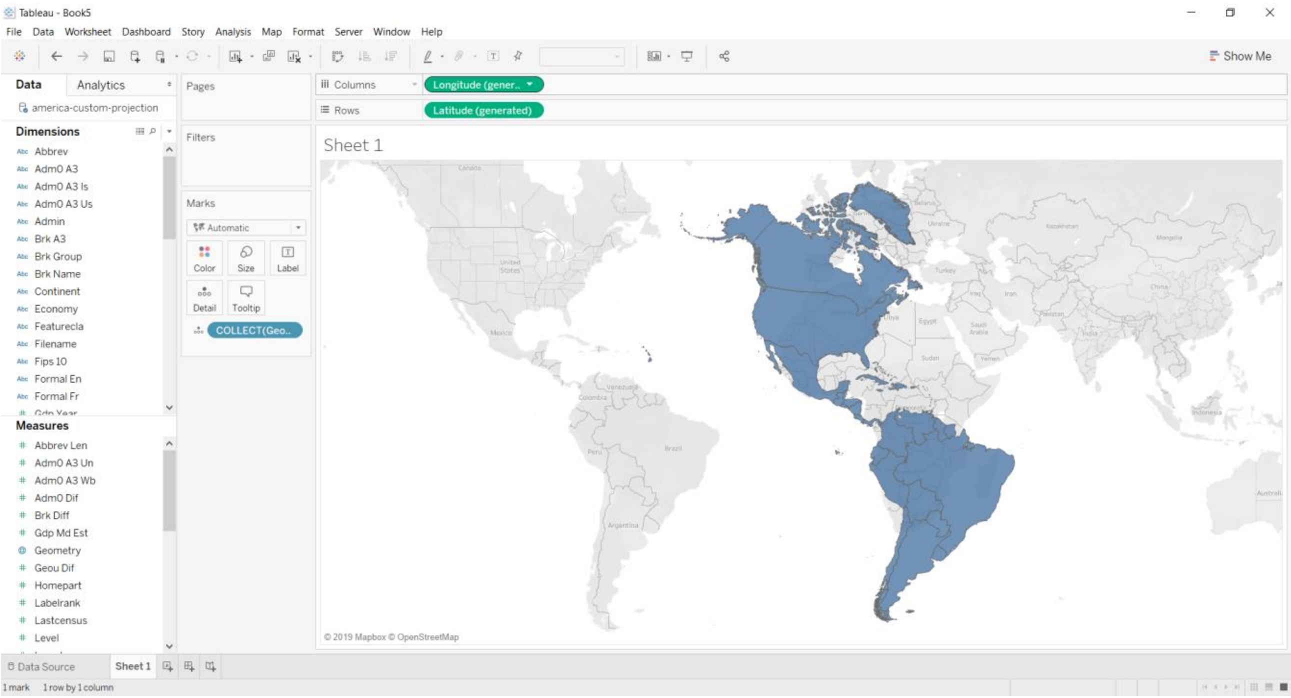
8. Click OK. You will see the result. This may look bizarre, but it is fine. The next step is exporting the layer to shapefile again.



9. Export the layer using the same way you did previously.
   Select the layer, then click **Layer** → **Save feature as**. Name it as "**america-custom-projection.shp**".
   When exporting, keep the CRS to EPSG:3957 – WGS 84 / Pseudo-Mercator.



10. Double check the Format, File name, and CRS.
11. Click OK to export.
12. Import the **america-custom-projection.shp** into Tableau. If you performed the previous section correctly, you should be able to see the following result. You notice that the imported shapefile does not align with the Mercator map in Tableau. When comparing the size of Greenland on the two maps, you also notice that the imported Americas are using the Lambert azimuthal projection, which is an area-preserving projection. Tableau now thinks that the imported shapefile already is in the Pseudo-Mercator projection and therefore does not apply any projection transformation when importing the shapefile.



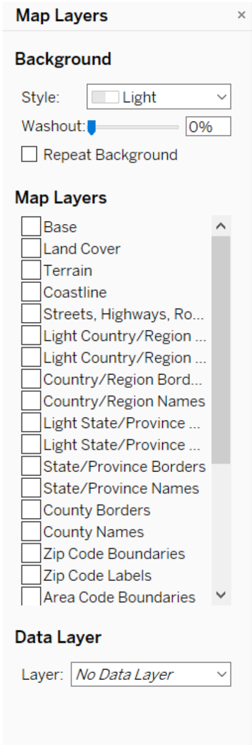
Blending with Tabular Data

In this section we will be using our custom map to create maps. The first step is to remove the base map. It does not make sense to keep it because it is not aligned with our map.

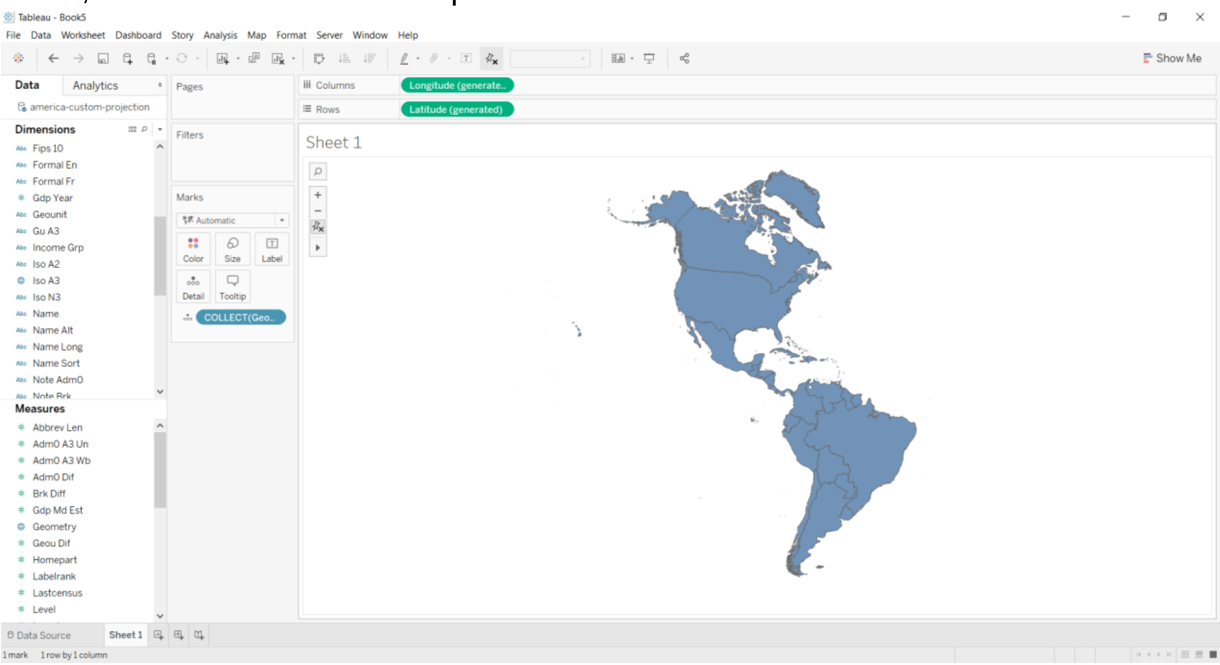
a. Removing the Background Map

As you learned earlier, Tableau gets the map from Mapbox's OpenStreetMap. OpenStreetMap usually contains multiple layers. We can control the visibility of those layers (i.e. roads, terrain, country border, etc.) in Tableau.

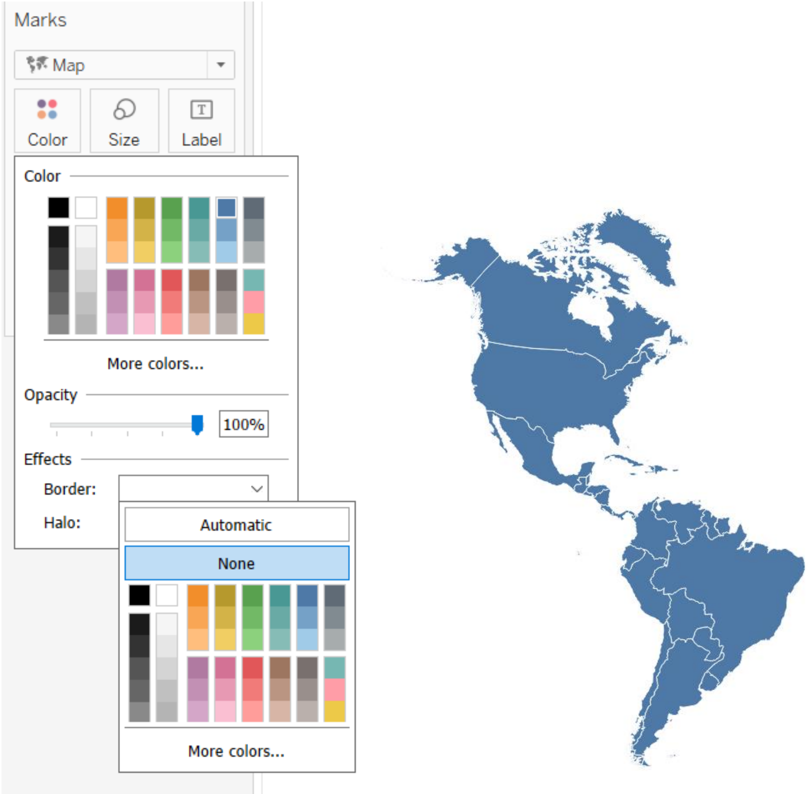
1. To show the setting, click **Maps → Map Layers.**
2. Uncheck all map layers in the **Map Layers** panel. This will hide all map layers.



3. Great, now we have a clean map.



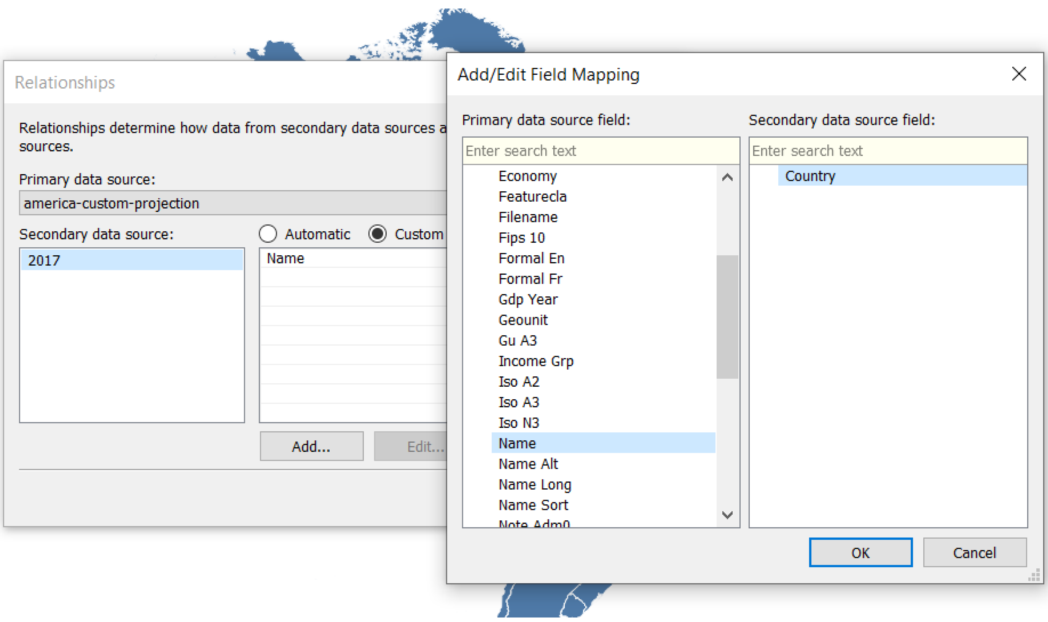
4. You can also change the shape and border colour by clicking **Color Mark**.



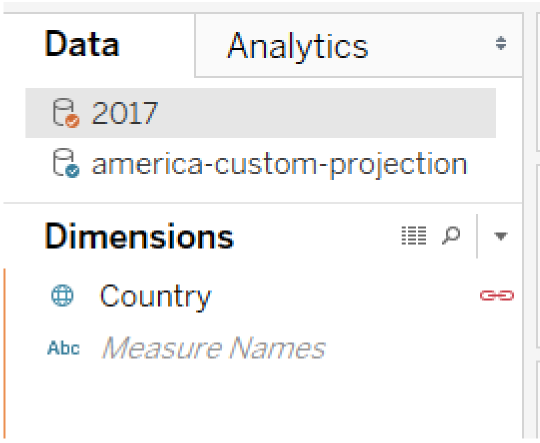
b. Blending Datasets and Creating a Choropleth Map

In this last section, we will blend our shapefile with tabular CSV data. We will be using the geometry from the shapefile and quantitative values from a CSV file to create a choropleth map and a proportional symbol map.

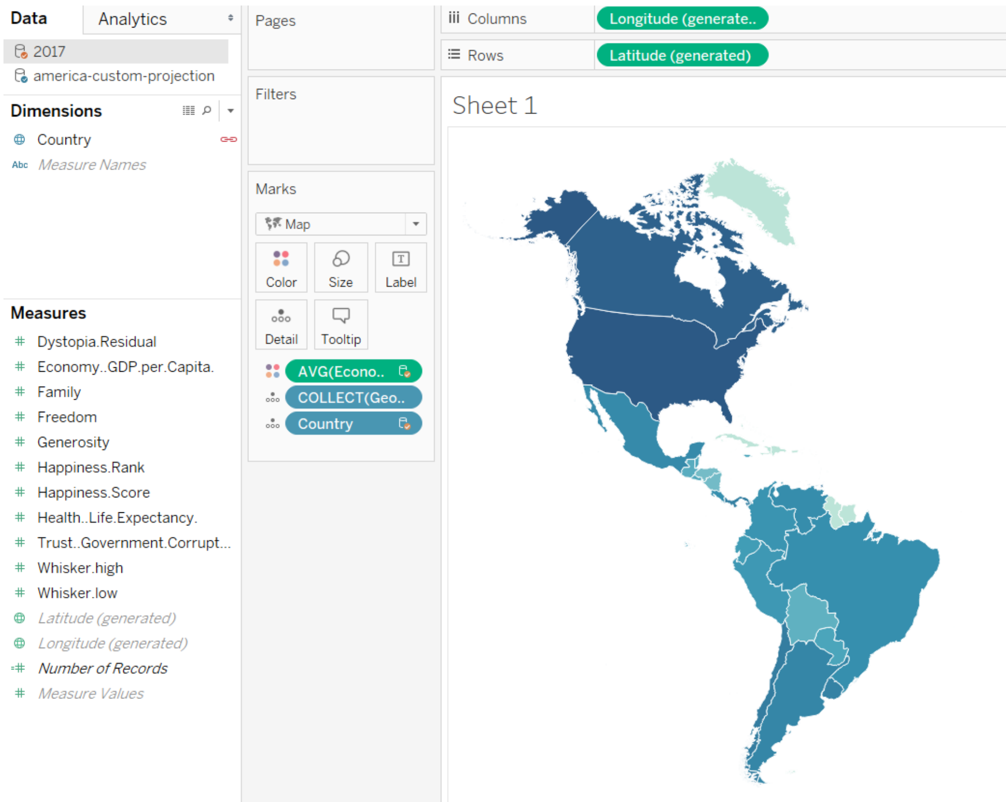
1. First, let's download and import the **world happiness dataset** from Kaggle https://www.kaggle.com/unsdsn/world-happiness/downloads/world-happiness-report.zip/2.
2. Go to Sheet 1 and connect our two datasets. Click **Data → Edit Relationship.**
3. Select **custom** radio button then click **Add**. We will connect these datasets using **Name** attribute on shapefile and **Country** on world happiness dataset.



4. A successful linking will reveal a link icon on the Country Dimension.



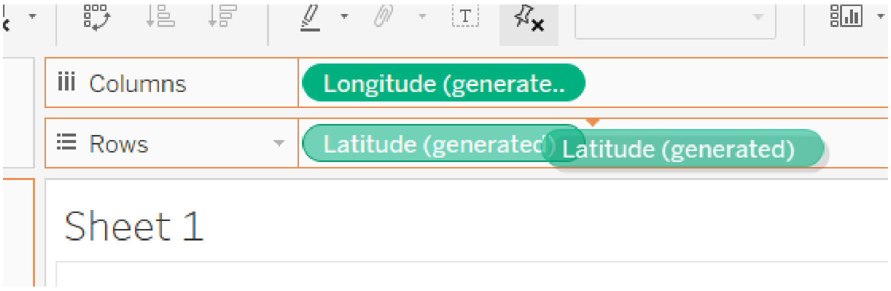
5. Now we can start creating the choropleth map. Drag **Economy GDP per Capita** to **Color Mark** and set it to **AVG**.
6. Then drop Country to Detail mark.
7. This is how your map should look like.



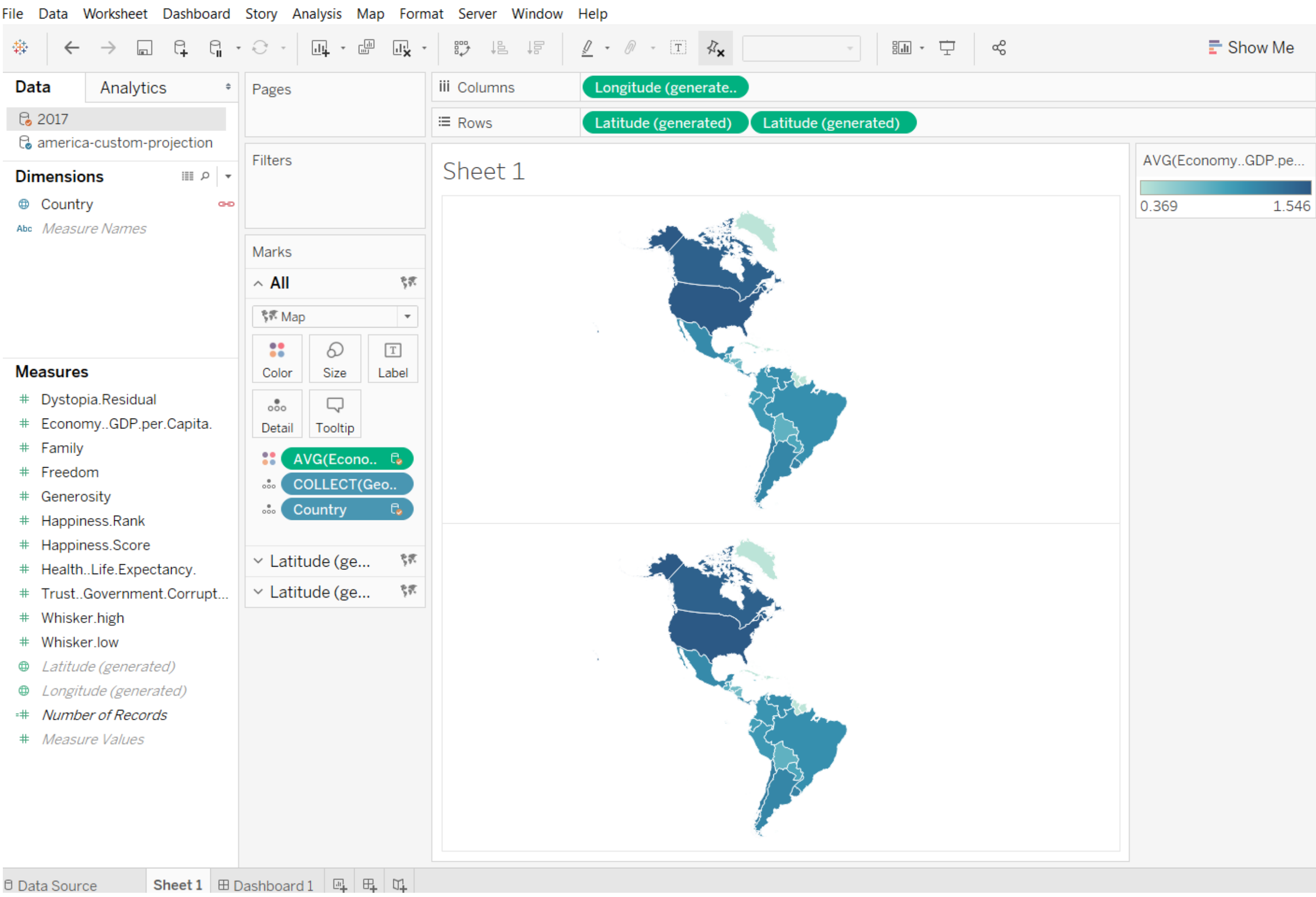
8. To make it even more informative, we can overlay a proportional symbol map on top of it.

c. Creating Proportional Symbol Map

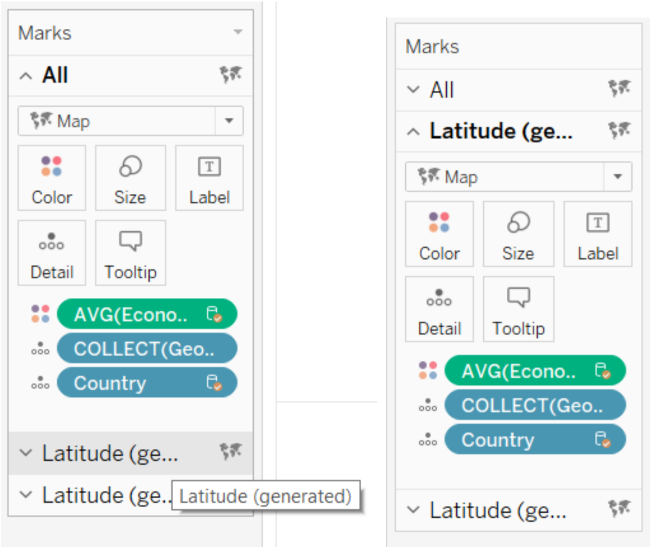
1. First, duplicate the **Latitude (generated)** measure in **Rows**. Press Alt + Ctrl (Windows) or Alt + Command (Mac) and Drag the **Latitude (generated)** to the left.



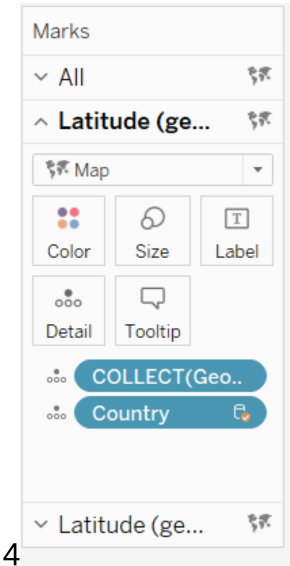
As a result, you will see two identical maps. If you recall our previous tutorials, you might notice that we are using the same technique that we used to create a dual axes visualisation.



2. Before combining the axes, we need to create the proportional symbol map. Expand the second **Mark** panel.

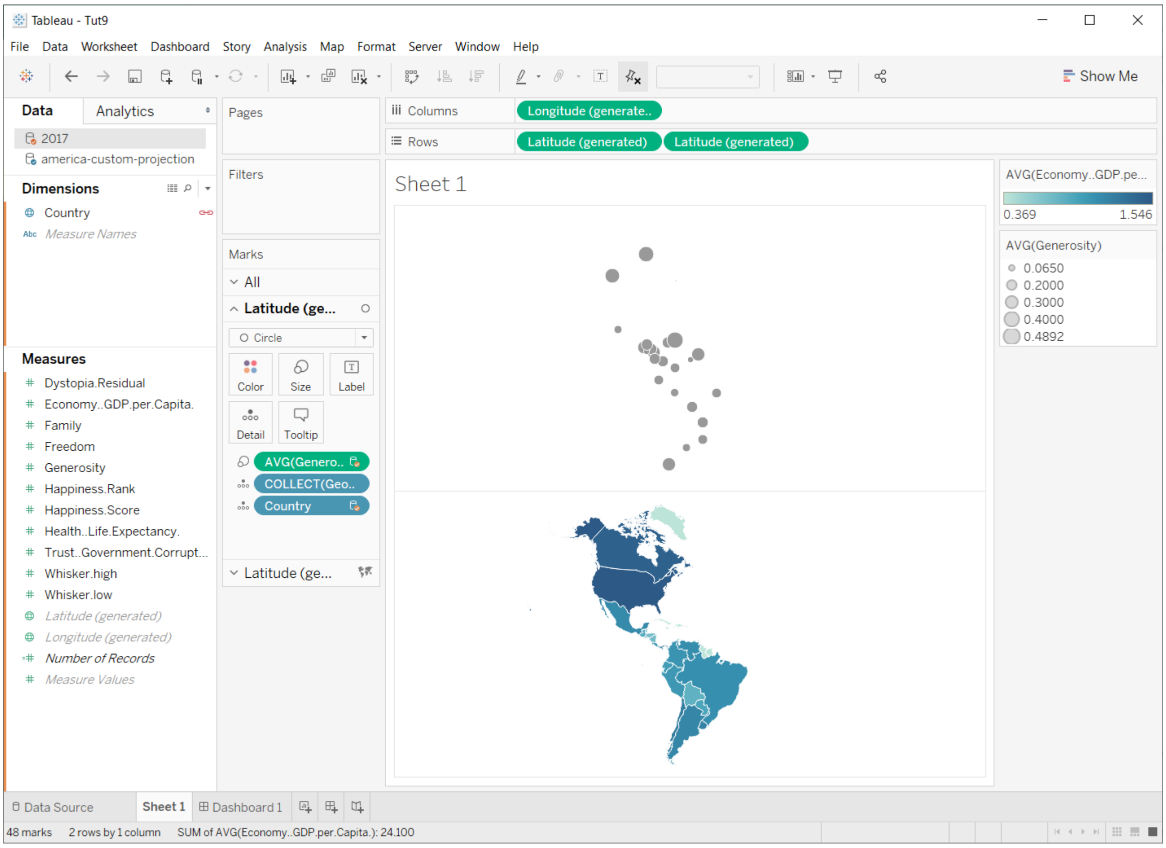


3. Remove **AVG(Economy GPD per Capita).**

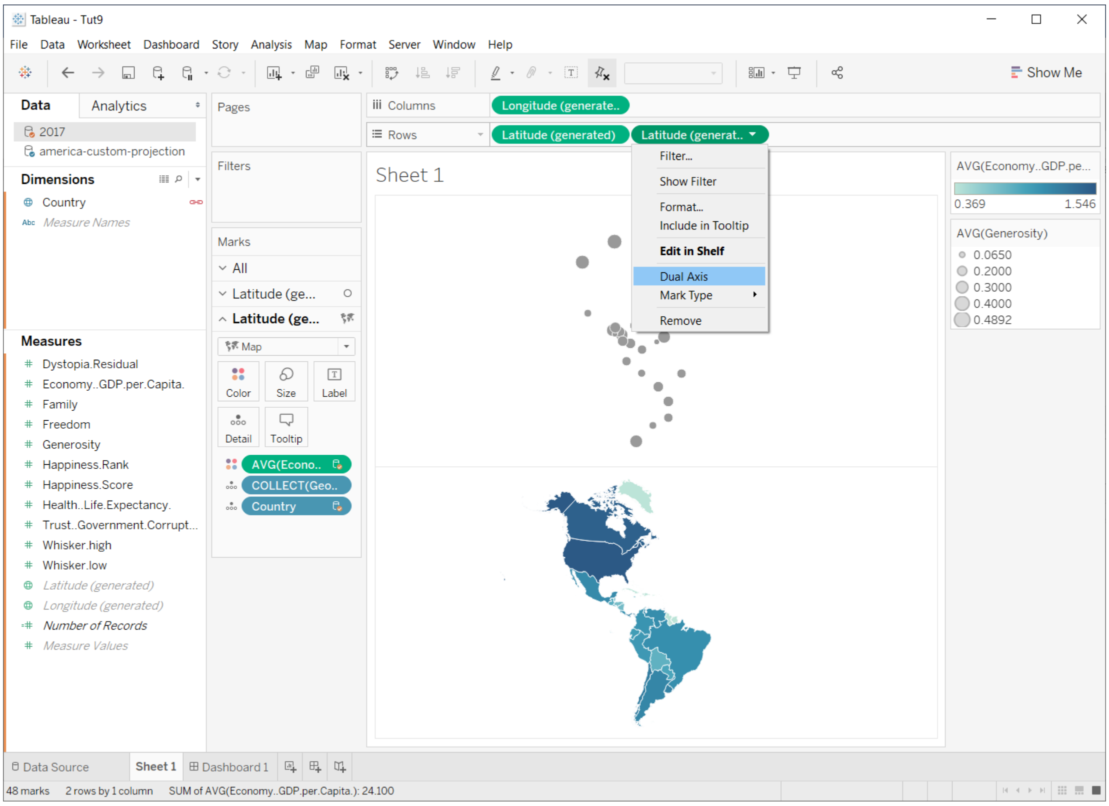


4

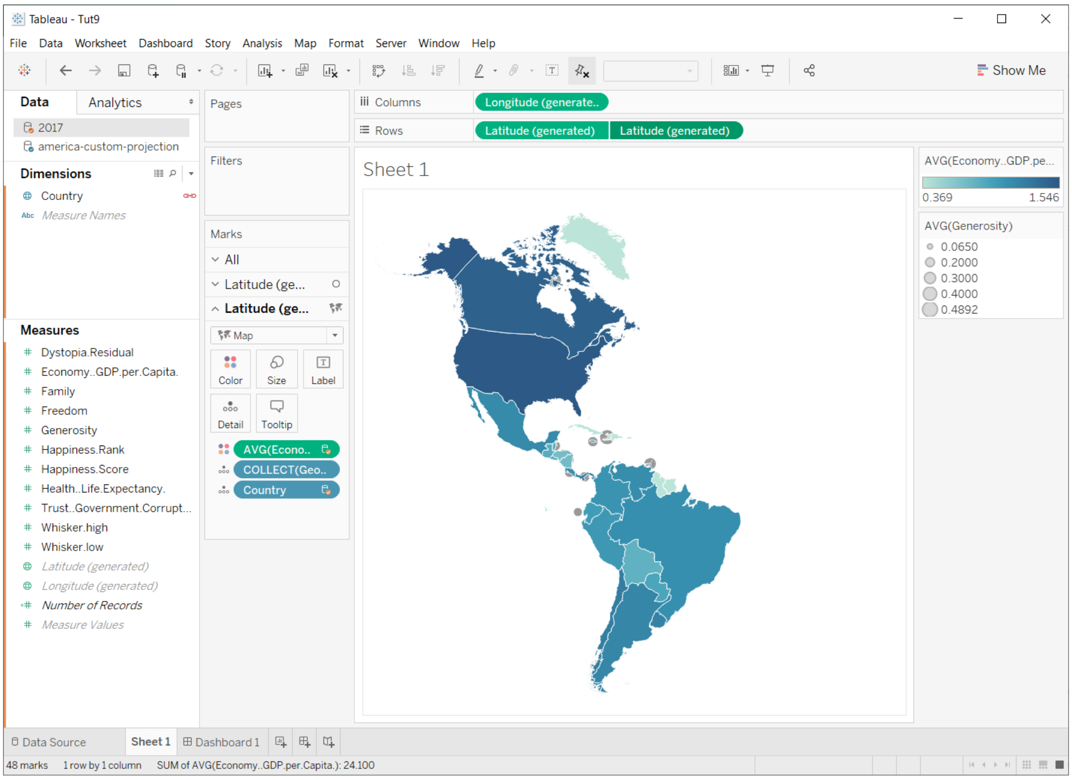
4. You can choose any Measures that you want. But, let's select Generosity for now. Drag the **Generosity** measures to **Size Mark** and **set to AVG**.
5. Then change the mark type **from Map to Circle**.
   As you can see, now we have a proportional symbol map and choropleth map.



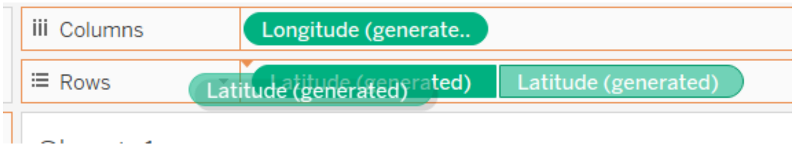
6. The last step will be to activate the dual axis. Click on **Latitude** (generated) then select **Dual Axis**.



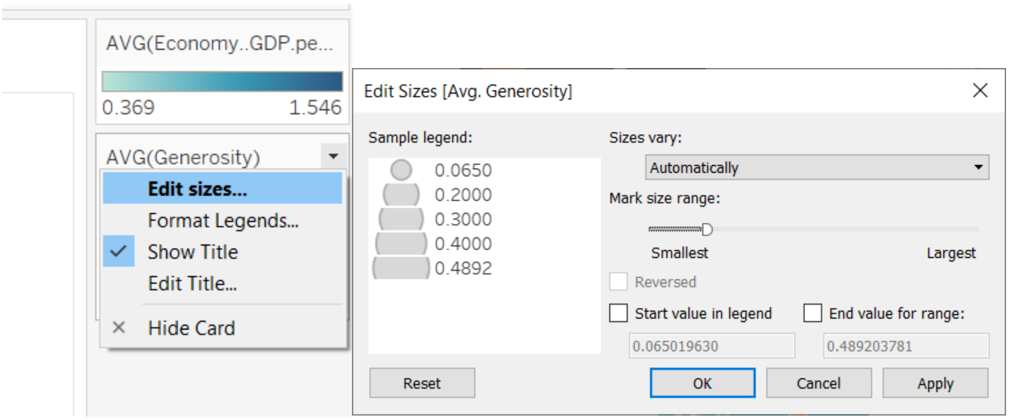
If your maps look like this, it is because the choropleth map layer is located on top of the proportional symbol map.



We can easily fix it by swapping the connected Latitude (generated) Measures in Rows.



7. As a final touch, you can adjust the size of the circles on the legend.



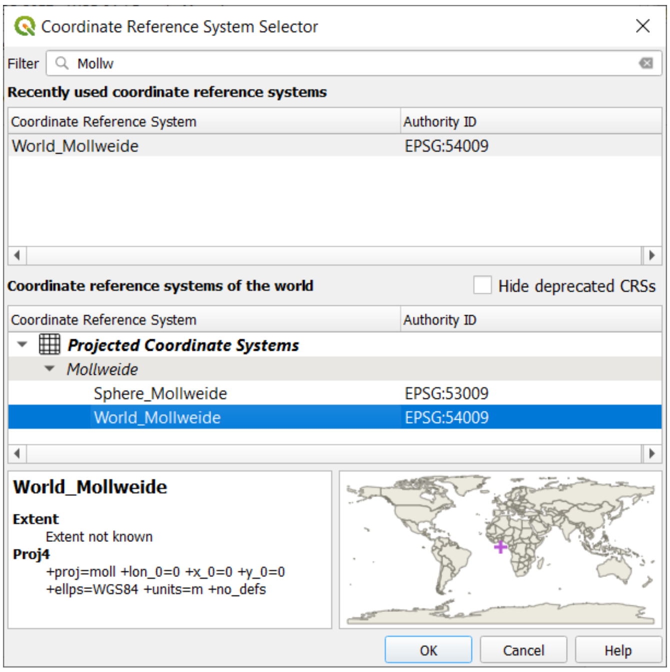
8. Congratulation, you just created a composite map with custom projection in Tableau.

Time for Practice

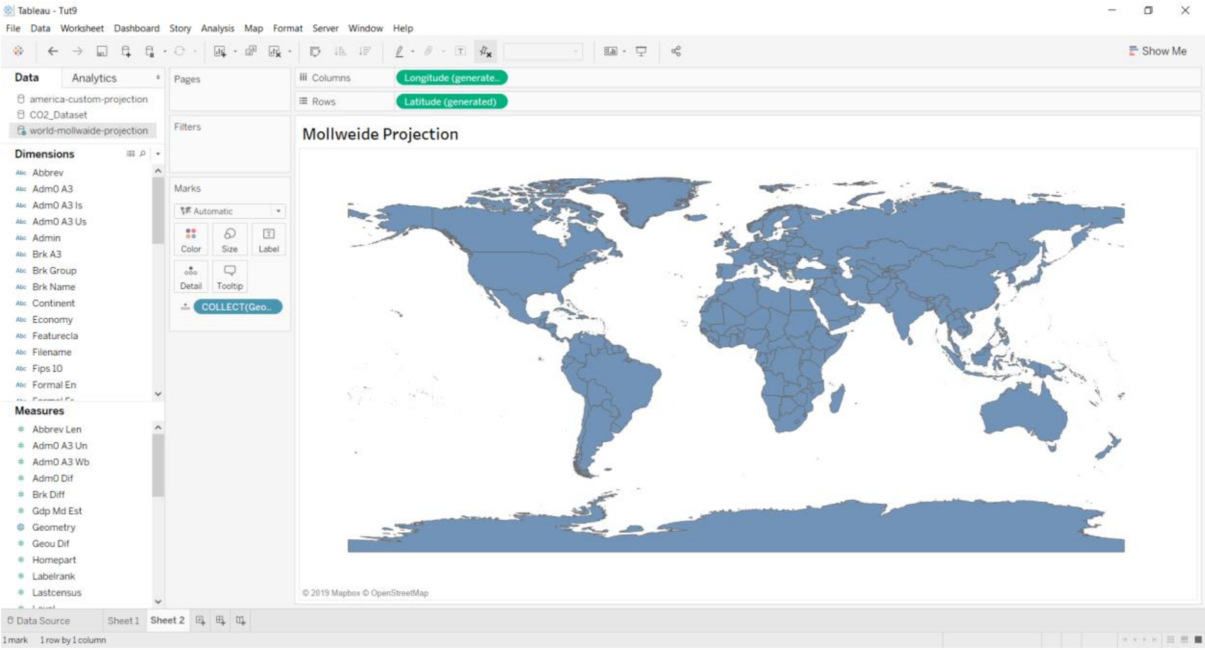
Now it is your time to rehearse what you just learned. Try to apply the same technique for creating world maps with Mollweide and Equal Earth projections.

1. Mollweide

Mollweide projection is integrated in QGIS. Therefore, you do not need to define a custom projection.

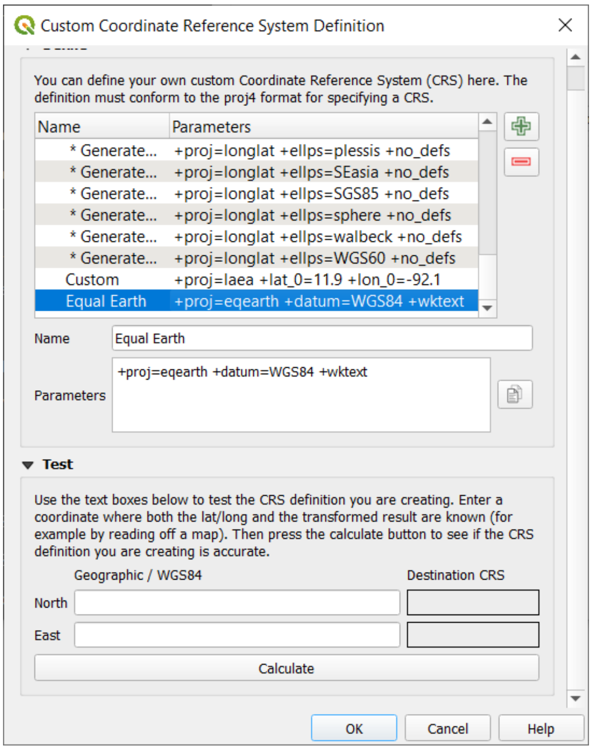


The final result should look like this.

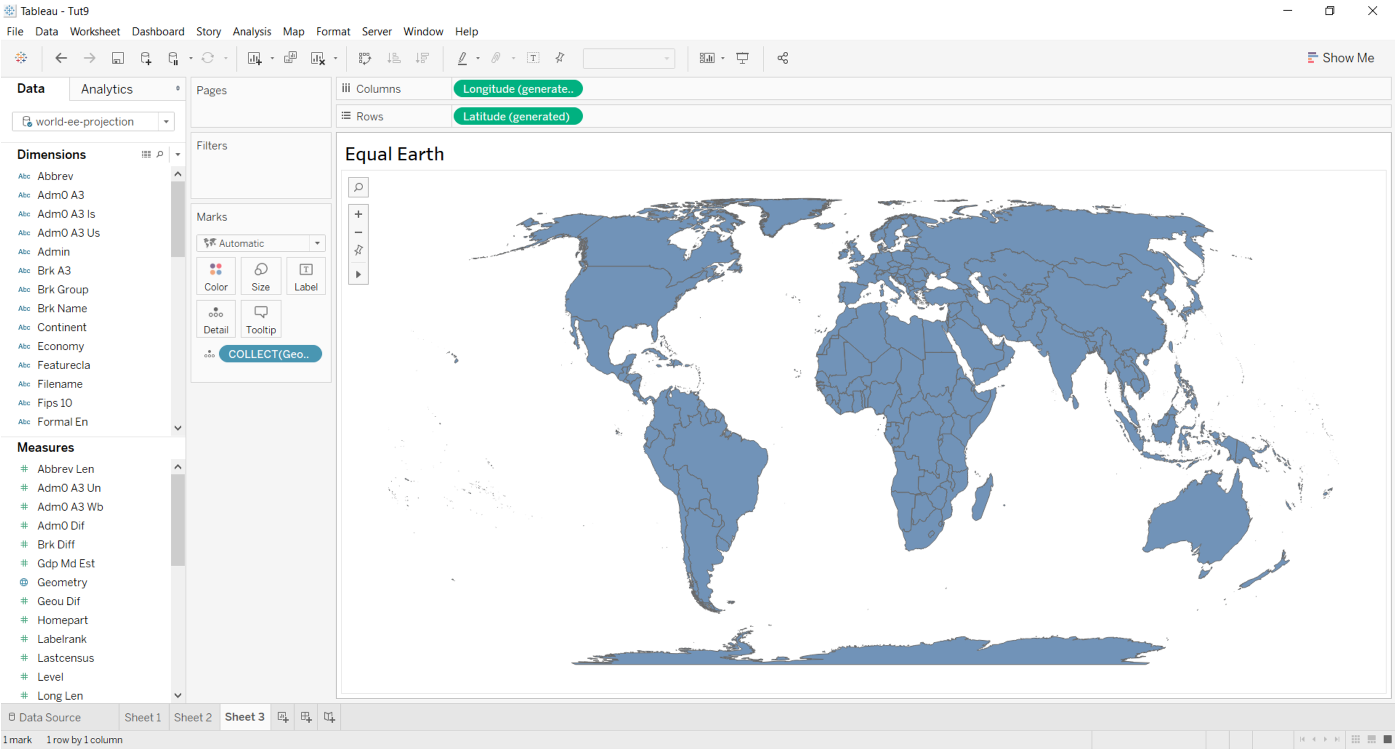


2. Equal Earth Projection

For Equal Earth projection, you need to define a custom projection. Use this string as the parameter: **+proj=eqearth +datum=WGS84 +wktext**



The result on Tableau will look like this.



Good luck and happy Mapping!