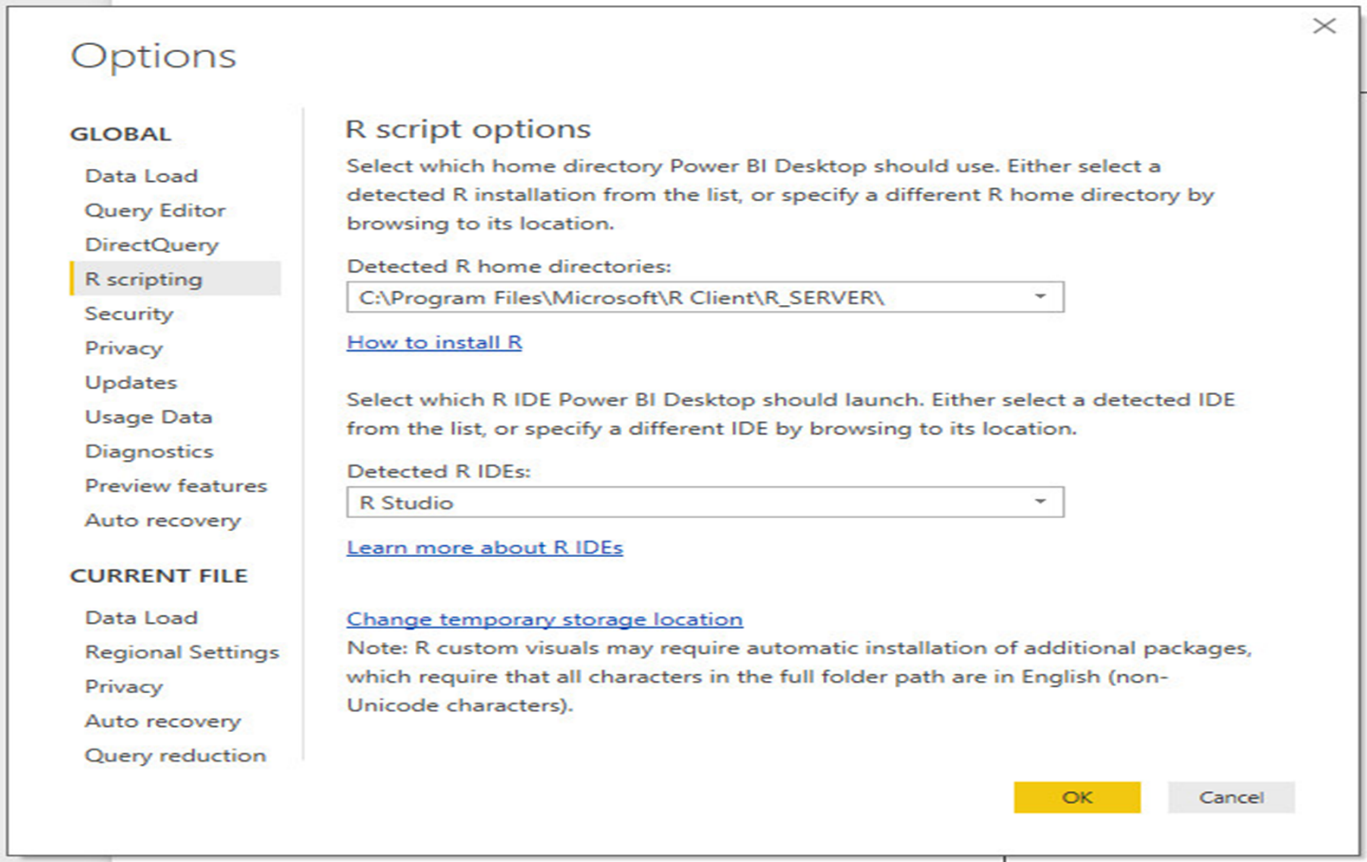
**Before you start:**

**Install R - download for free from the following site** [**https://mran.microsoft.com/open**](https://mran.microsoft.com/open)

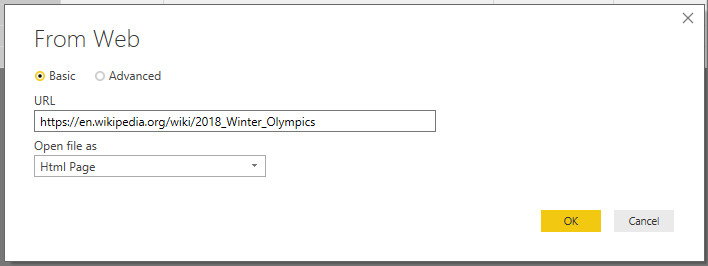
**In Power BI Desktop enable R by going to File > Options > R scripting and filling in the two fields.**

****

**Get Olympics data**

The first thing we need to do is get the data. We will use the Web data connector to pull down data from Wikipedia.

Open up Power BI and get data using the Web connector.



Select the Medal table[edit] data set. This will give us some 2018 Winter Olympics data that we can use.

Here is the link to the data- <https://en.wikipedia.org/wiki/2018_Winter_Olympics>

**Transform the data**

We need to do little bit of cleanup on the data to get it in shape for the purposes of this tutorial.

Apply the following transformations to this data.

1. Edit the data to separate out the country, rename columns to Country and NOC
2. Edit the data to change Olympic Athletes from Russia to just Russia
3. Unpivot Gold, Silver and Bronze and rename the fields (Medal Type and Medal Count)
4. Remove the Total column

Your M code should look as follows:

let

    Source = Web.Page(Web.Contents("https://en.wikipedia.org/wiki/2018\_Winter\_Olympics")),

    Data6 = Source{6}[Data],

    #"Changed Type" = Table.TransformColumnTypes(Data6,{{"Rank", Int64.Type}, {"NOC", type text}, {"Gold", Int64.Type}, {"Silver", Int64.Type}, {"Bronze", Int64.Type}, {"Total", Int64.Type}}),

    #"Split Column by Delimiter" = Table.SplitColumn(#"Changed Type", "NOC", Splitter.SplitTextByDelimiter("#(00A0)", QuoteStyle.Csv), {"NOC.1", "NOC.2"}),

    #"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"NOC.1", type text}, {"NOC.2", type text}}),

    #"Renamed Columns" = Table.RenameColumns(#"Changed Type1",{{"NOC.1", "Country"}, {"NOC.2", "Code"}}),

    #"Replaced Value" = Table.ReplaceValue(#"Renamed Columns","Olympic Athletes from Russia","Russia",Replacer.ReplaceText,{"Country"}),

    #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Replaced Value", {"Rank", "Country", "Code", "Total"}, "Attribute", "Value"),

    #"Renamed Columns1" = Table.RenameColumns(#"Unpivoted Columns",{{"Attribute", "Medal Type"}, {"Value", "Medal Count"}}),

    #"Removed Columns" = Table.RemoveColumns(#"Renamed Columns1",{"Total"})

in

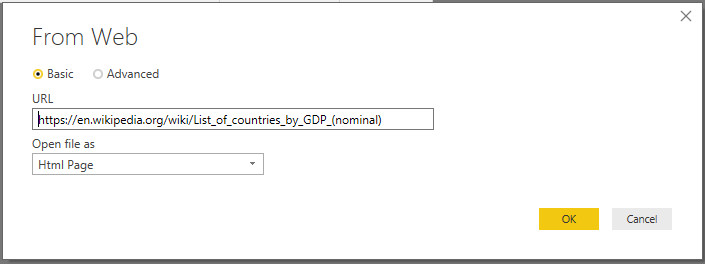
    #"Removed Columns"

Your data table should look like the image below.



**Get GDP data**

Simlar to above we will pull down some GDP by Country data from Wikipedia using the Web data connector.



Here is the link to the data : https://en.wikipedia.org/wiki/List\_of\_countries\_by\_GDP\_(nominal)

**Transform the data**

Lets clean up the data so that its ready to use in this tutorial.

At a high level we will apply the following steps

1. Filter out where Rank is blank or -
2. Remove the /NA from GDP column by using split by delimiter delete the extra column
3. Rename column to GDP
4. Rename Table1 to GDP by Country

Your M code should look like the following snippet.

let

    Source = Web.Page(Web.Contents("https://en.wikipedia.org/wiki/List\_of\_countries\_by\_GDP\_(nominal)")),

    Data1 = Source{1}[Data],

    #"Changed Type" = Table.TransformColumnTypes(Data1,{{"Rank", type text}, {"Country", type text}, {"GDP #(lf)(US$MM)", type text}}),

    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each ([Rank] <> "" and [Rank] <> "—")),

    #"Split Column by Delimiter" = Table.SplitColumn(#"Filtered Rows", "GDP #(lf)(US$MM)", Splitter.SplitTextByEachDelimiter({"/"}, QuoteStyle.Csv, true), {"GDP #(lf)(US$MM).1", "GDP #(lf)(US$MM).2"}),

    #"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"GDP #(lf)(US$MM).1", type number}, {"GDP #(lf)(US$MM).2", type text}}),

    #"Removed Columns" = Table.RemoveColumns(#"Changed Type1",{"GDP #(lf)(US$MM).2"}),

    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"GDP #(lf)(US$MM).1", "GDP"}})

in

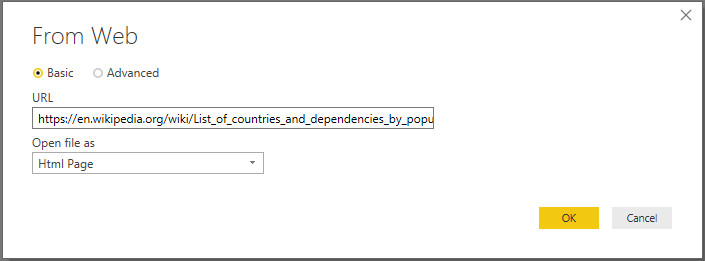
    #"Renamed Columns"

Once done your data should look like this.



**Get Population Data**

Same as before use the Web connnector to pull down data from Wikipedia. In case you are wondering Wikipedia is a great source of data formated in tables that is easy for the connector to use.



Here is the link to the data: https://en.wikipedia.org/wiki/List\_of\_countries\_and\_dependencies\_by\_population

**Tranform the data**

Apply the transformation following steps to the data.

let

    Source = Web.Page(Web.Contents("https://en.wikipedia.org/wiki/List\_of\_countries\_and\_dependencies\_by\_population")),

    Data0 = Source{0}[Data],

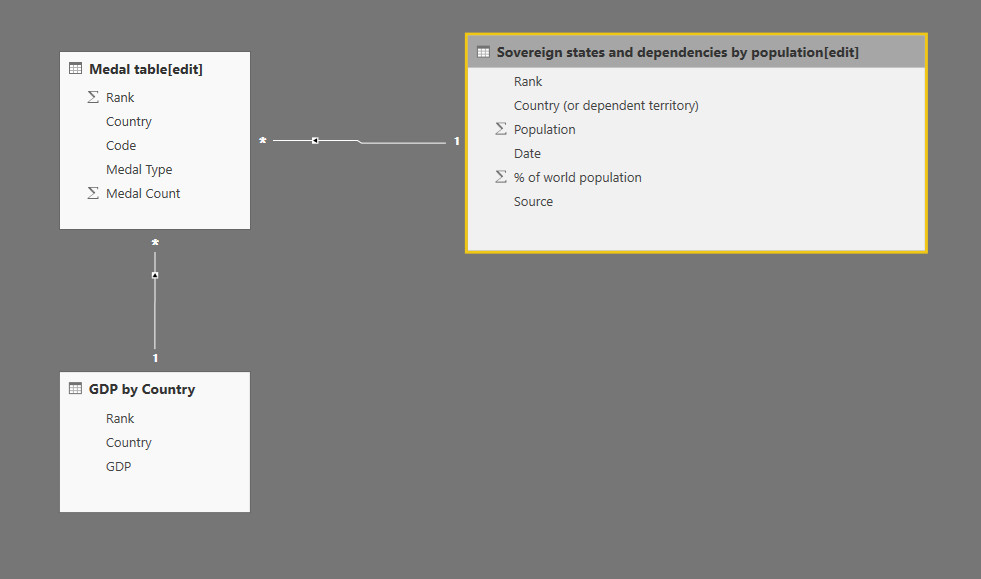
    #"Changed Type" = Table.TransformColumnTypes(Data0,{{"Rank", type text}, {"Country #(lf)(or dependent territory)", type text}, {"Population", type number}, {"Date", type date}, {"% of world #(lf)population", Percentage.Type}, {"Source", type text}})

in

    #"Changed Type"

**Relationships**

You should now have 3 data sets in your Power BI report and they should relate to each other by country.

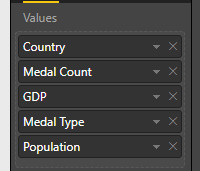


**Visualize the data**

Now lets take a closer look at the data and see if we can find anything interesting. We will use the R visual and ggplot to create some interesting graphs.

**Medal Count by GDP**

First lets take a look at Medal Count by GDP. Select the values Country, Medal Count, GDP, Medal Type and Population to pass into your R visual.



Lets compare the numebr of medals obtained (so far) by GDP using the following R code.

library("ggplot2")

library("scales")

ggplot(dataset, aes(x=GDP, y=`Medal Count`))  +

 geom\_point( aes(colour=`Medal Type`, size=3)) +

 scale\_colour\_manual(values = c("orange", "gold", "gray")) +

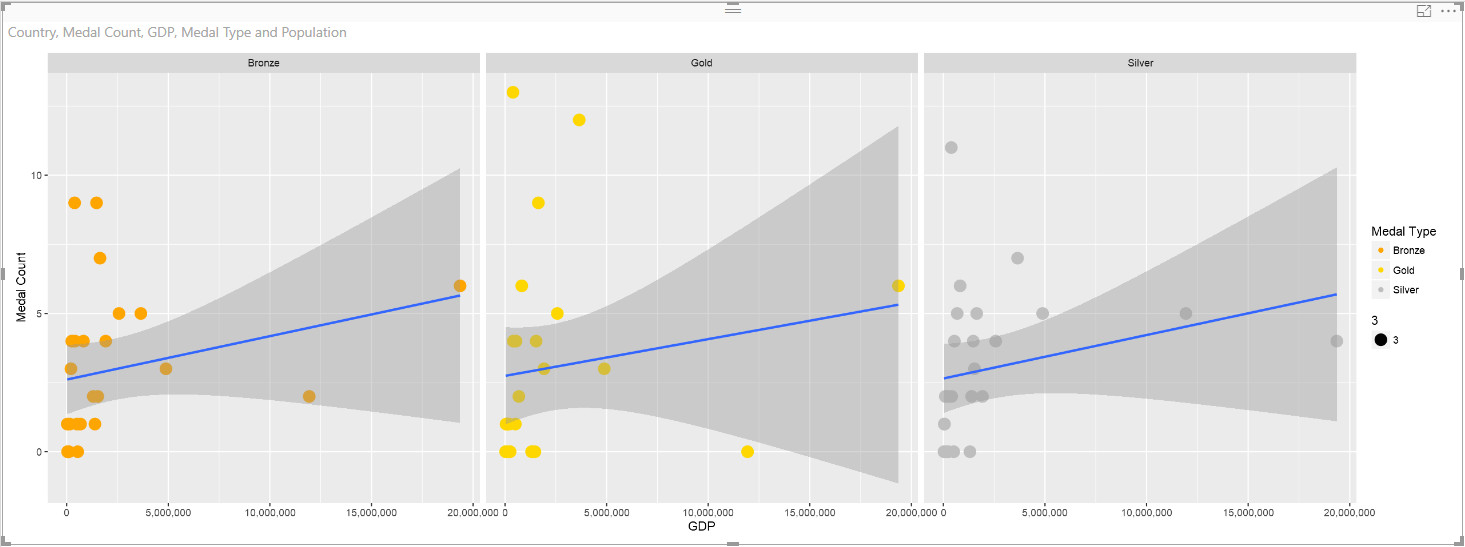
 facet\_wrap(~ `Medal Type` ) +

 scale\_x\_continuous(labels = comma) +

 geom\_smooth(method='lm',formula=y~x)

Notice how I used the back tick ` to plot the Medal Count field on the y axis. This is due to the fact that the Medal Count field has a space in it.

This will produce the following visual.



Feel free to play with the code and see how each line works. You can comment out different lines using the # character but keep in mind you will also need to remove the + sign on the previous line if you are breaking appart the ggplot code.

**Medal Count by Population**

Next lets duplicate this page and modify the R code. Enter the following code into the copied R visual.

library("ggplot2")

library("scales")

ggplot(dataset, aes(x=Population, y=`Medal Count`))  +

 geom\_point( aes(colour=`Medal Type`, size=3)) +

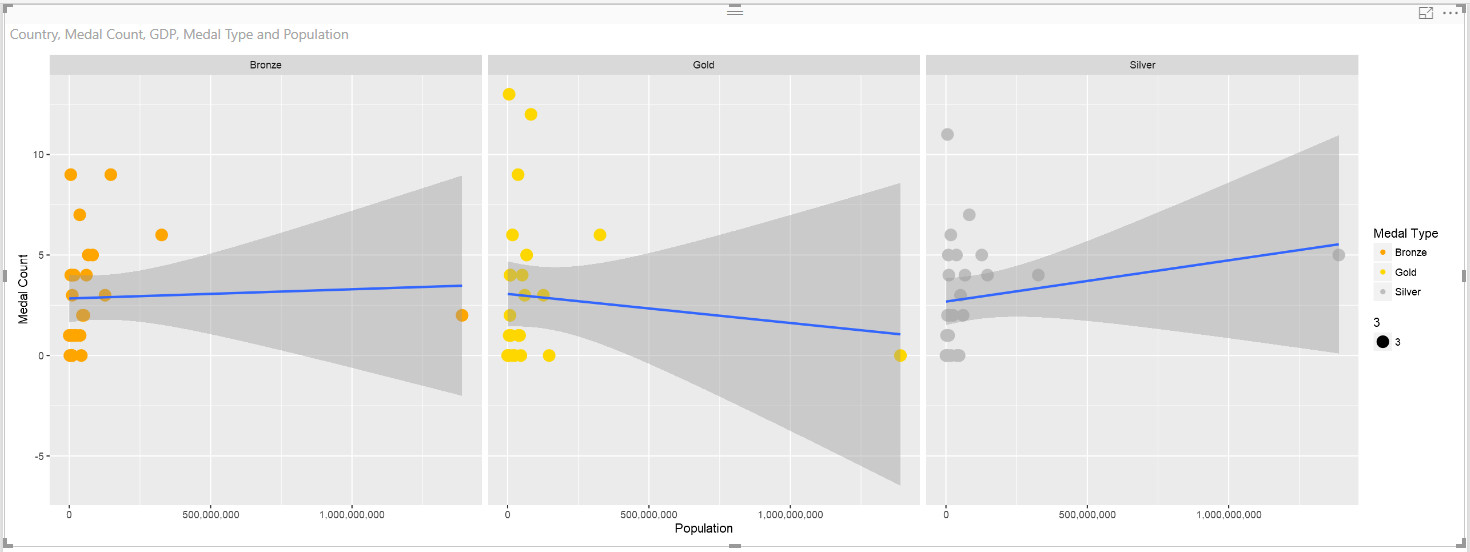
 scale\_colour\_manual(values = c("orange", "gold", "gray")) +

 facet\_wrap(~ `Medal Type` )+

 scale\_x\_continuous(labels = comma) +

 geom\_smooth(method='lm',formula=y~x)

This will give you the following visual.



Add additional visuals to the Power BI report such as slicers and charts so that you can play with the data and see how the plots change based on different filter criteria. For example this is my Medal Count by GDP report.

**Visualize the data**

First we will alter the R code that we developed. Create a duplicate the Medal Count by GDP page and alter the R visual code so that faceting is remove. Your R code should like this.

library("ggplot2")

library("scales")

ggplot(dataset, aes(x=GDP, y=`Medal Count`))  +

 geom\_point( aes(colour=`Medal Type`, size=3)) +

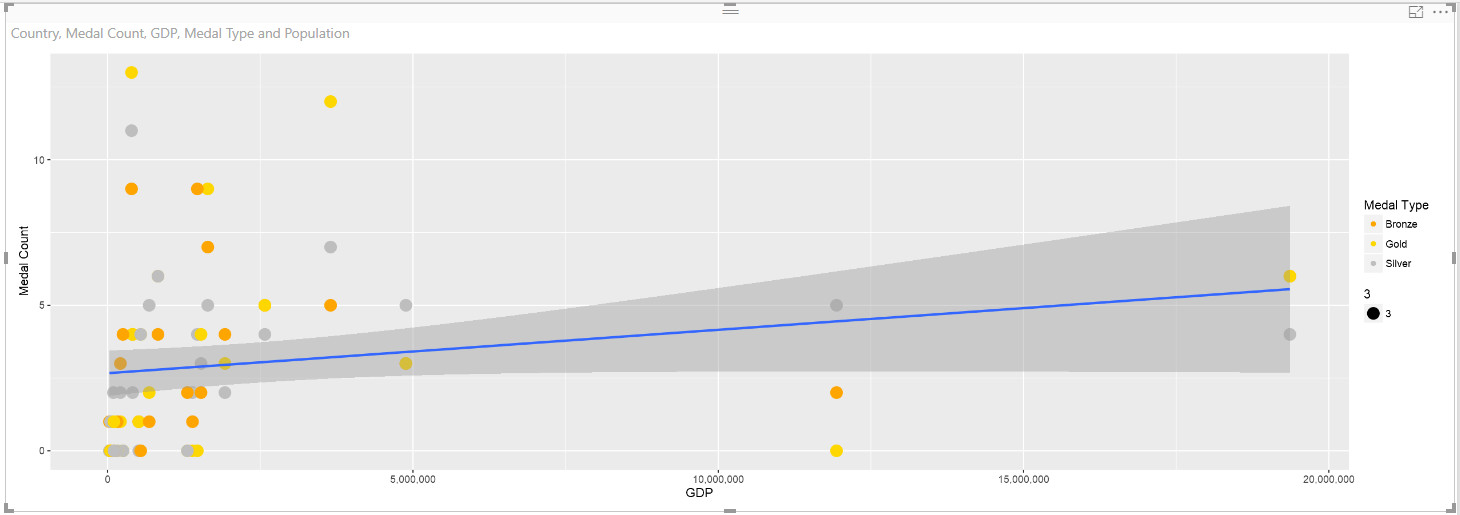
 scale\_colour\_manual(values = c("orange", "gold", "gray")) +

*#facet\_wrap(~ `Medal Type` )+*

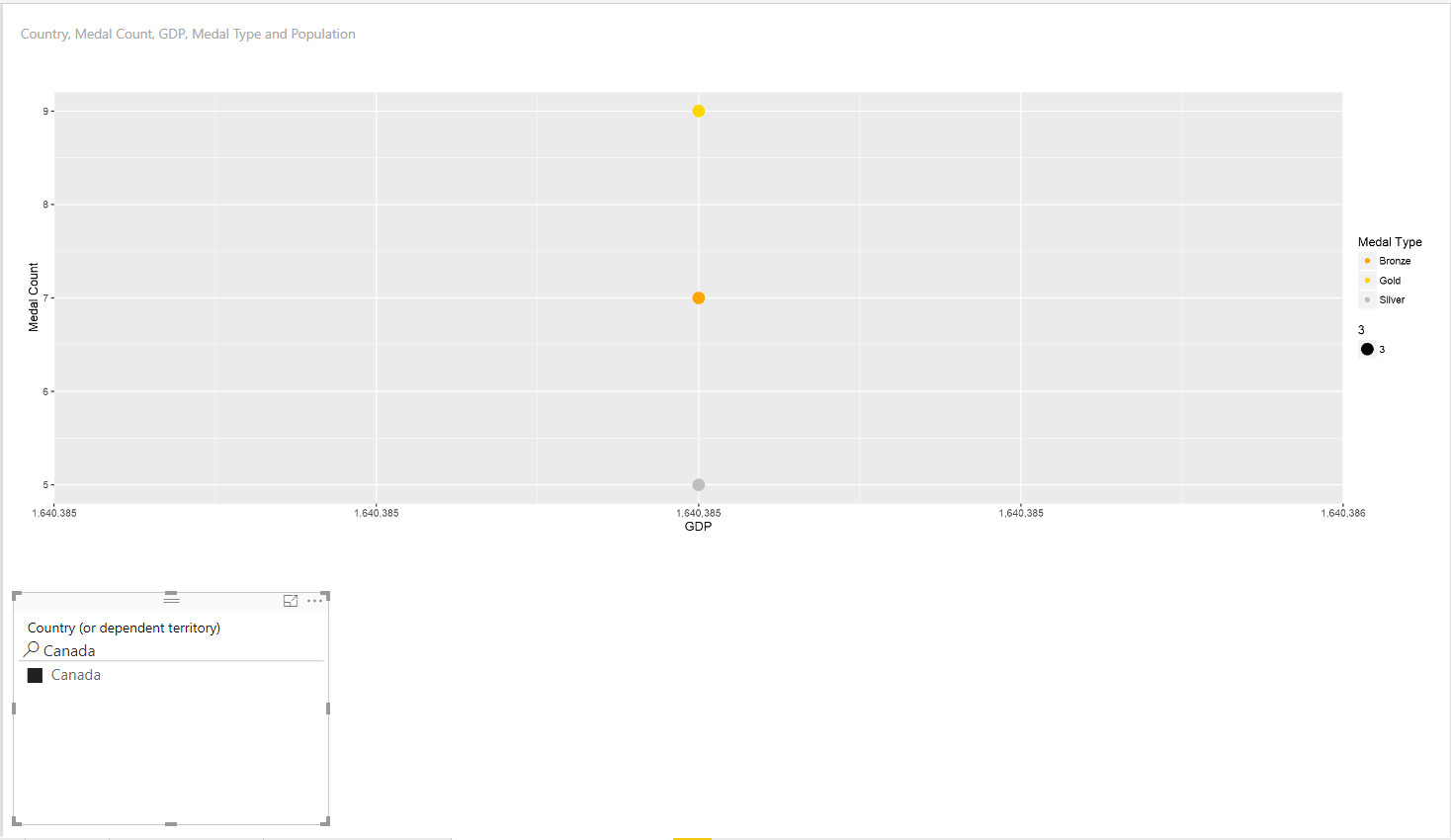
 scale\_x\_continuous(labels = comma)+

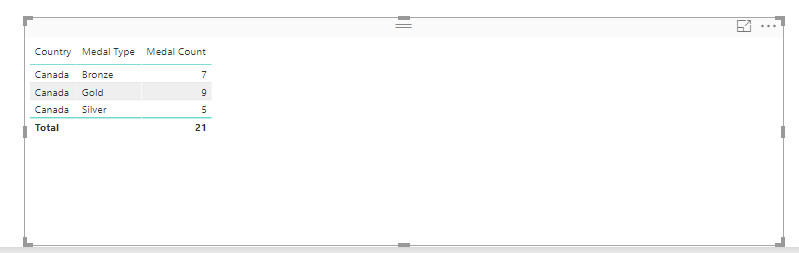
 geom\_smooth(method='lm',formula=y~x)

I commented out the facet\_wrap line using the # character. This produces the following plot.



Drop a slicer onto the canvas of the report and select Country from the Sovereign states and dependencies by population[edit] table. Select Canada from the list of countries on the slicer.

Your report should look something like this.

As you may notice there are 3 plot points for Canada. That is because Canada won 7 bronze, 5 silver and 9 gold so there are 3 unique records in the dataset passed into the R visual. If you drop a simple table on the canvas of the report and select Country, Medal Type and Medal Count you can quickly see these 3 unique records.

Lets tweak the R visual so that the size of the points correlate to the number of medals.

Modify the R code as follows.

library("ggplot2")

library("scales")

ggplot(dataset, aes(x=GDP, y=`Medal Count`))  +

 geom\_point( aes(colour=`Medal Type`, size=`Medal Count`)) +

 scale\_colour\_manual(values = c("orange", "gold", "gray")) +

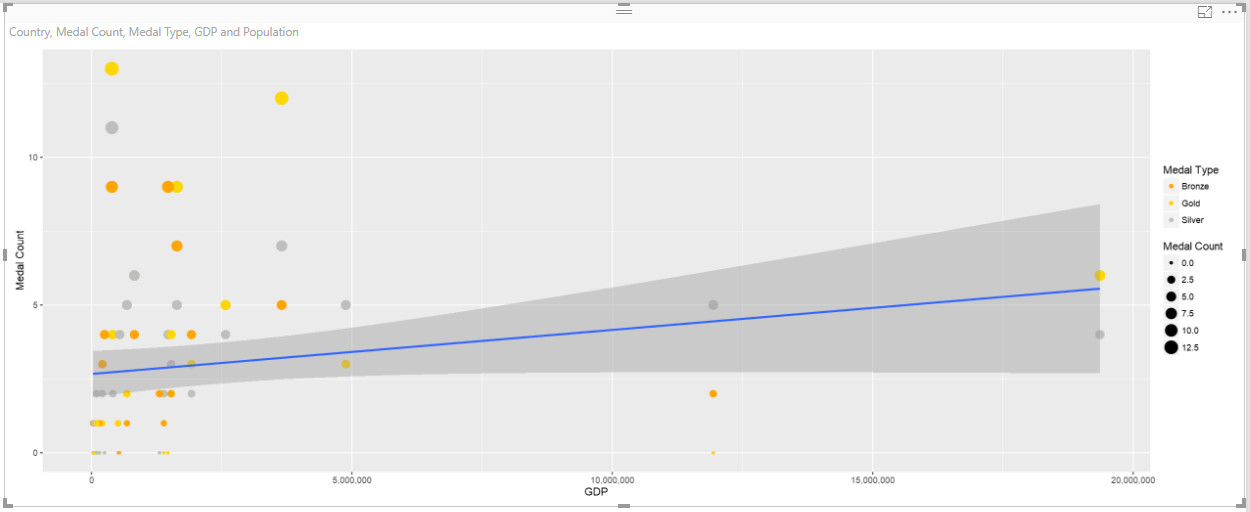
*#facet\_wrap(~ `Medal Type` )+*

 scale\_x\_continuous(labels = comma)+

 geom\_smooth(method='lm',formula=y~x)

We are now setting the size of the points to the value Medal Count.

This will produce the following plot.



**Add new vectors to your data frame**

Next lets add some additional fields to the data set and explore them for value.

Modify the R code as follows.

dataset$Medals.per.Pop <- dataset$`Medal Count` / (dataset$Population/1000000)

dataset$Medals.per.GDP <- dataset$`Medal Count` / (dataset$GDP/10000)

plot(dataset)

#library("ggplot2")

#library("scales")

#ggplot(dataset, aes(x=GDP, y=`Medal Count`))  +

# geom\_point( aes(colour=`Medal Type`, size=`Medal Count`)) +

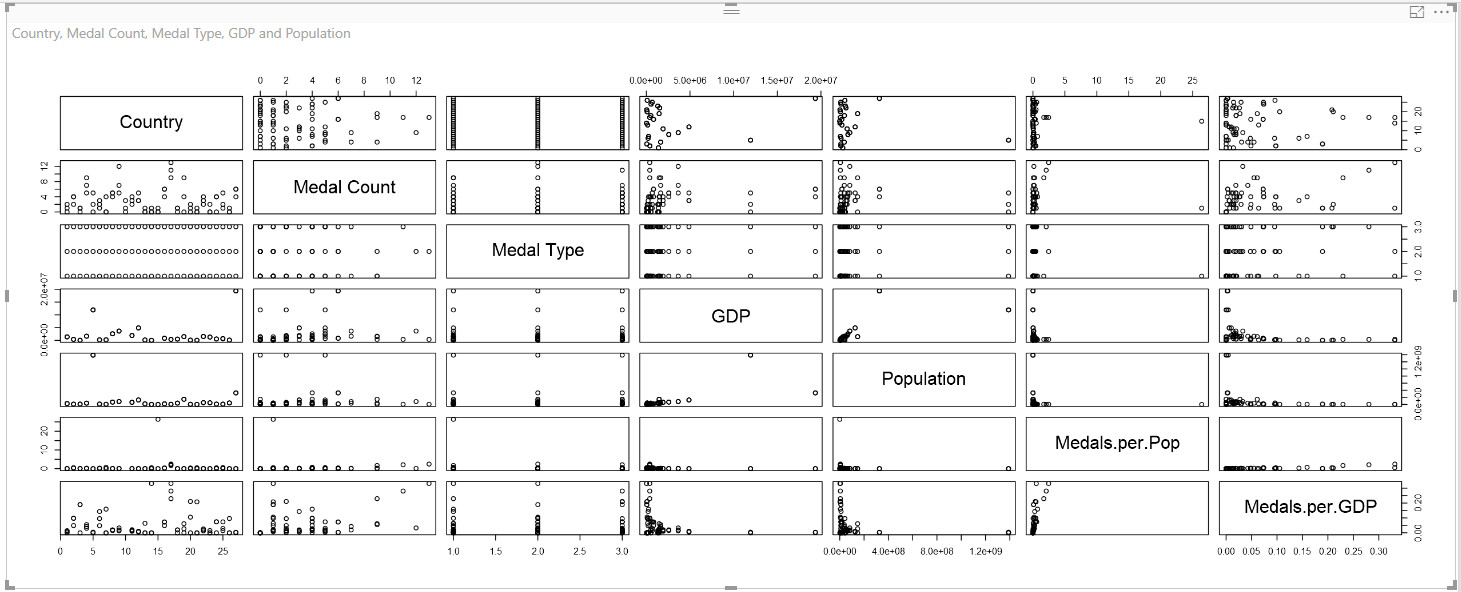
# scale\_colour\_manual(values = c("orange", "gold", "gray")) +

#facet\_wrap(~ `Medal Type` )+

# scale\_x\_continuous(labels = comma)+

*# geom\_smooth(method='lm',formula=y~x)*

This will produce the following plot.



As you can see we now have two new fields in our data set Medals.per.Pop and Medals.per.GDP. I commented out the use of GGPLOT and just used the base Plot due to the simplicity of using it to quickly visualize new data and compare it to existing data points.

**Histograms**

Let's take a closer look at the distribution of values for our two new features.

Make a duplicate of the page created in the previous article and modify the code for the R visual as follows.

#Add new features to the data set which show medal count per million people and medal count per million US GDP

dataset$Medals.per.Pop <- dataset$`Medal Count` / (dataset$Population/1000000)

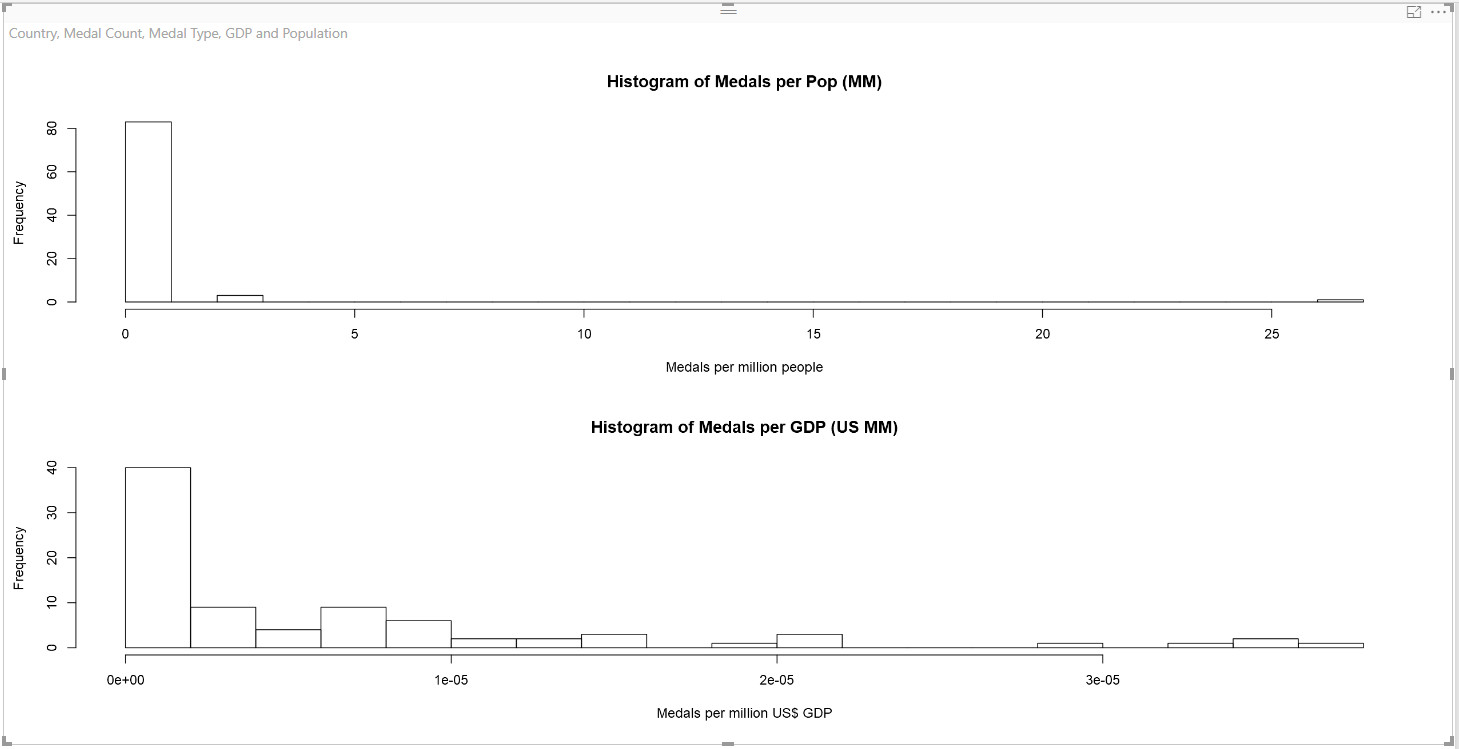
dataset$Medals.per.GDP <- dataset$`Medal Count` / (dataset$GDP)

#Create a two panel plot and show a histogram of the two new values

par(mfrow = c(2,1))

hist(dataset$Medals.per.Pop, xlab="Medals per million people", main="Histogram of Medals per Pop (MM)", breaks =20)

hist(dataset$Medals.per.GDP, xlab="Medals per million US$ GDP", main="Histogram of Medals per GDP (US MM)", breaks=20)

This will produce a plot similar to the following image.

**Scatter Plots**

Next let's add some scatter plots to the visual. Alter your R code so that it looks as follows.

#Add new features to the data set which show medal count per million people and medal count per million US GDP

dataset$Medals.per.Pop <- dataset$`Medal Count` / (dataset$Population/1000000)

dataset$Medals.per.GDP <- dataset$`Medal Count` / (dataset$GDP)

*#Create a two by two panel and plot two histograms and two scatterplots for the new values*

par(mfrow = c(2,2))

#Histograms

hist(dataset$Medals.per.Pop, xlab="Medals per million people", main="Histogram of Medals per Pop (MM)", breaks =20)

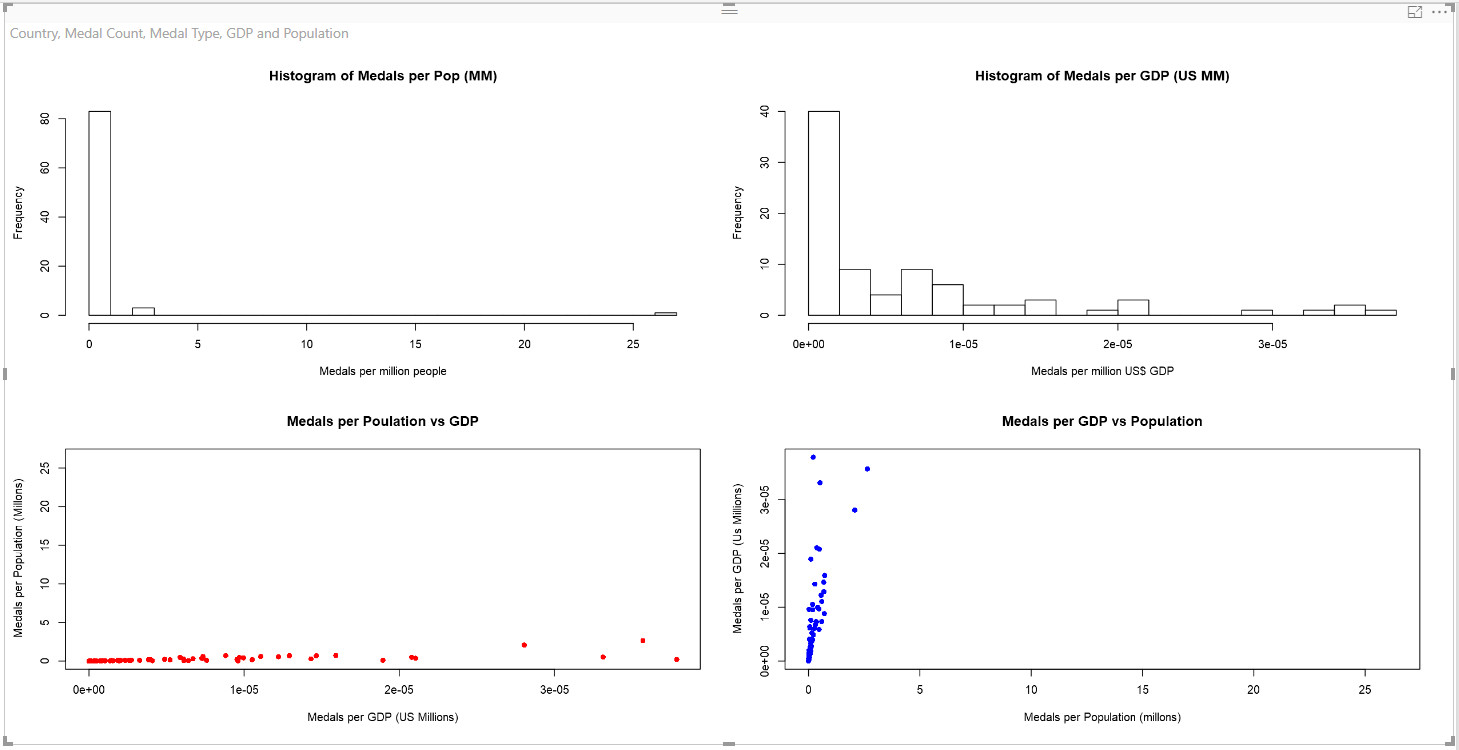
hist(dataset$Medals.per.GDP, xlab="Medals per million US$ GDP", main="Histogram of Medals per GDP (US MM)", breaks=20)

*#Scatterplots*

plot(dataset$Medals.per.GDP, dataset$Medals.per.Pop, pch=16, col="red", xlab = "Medals per GDP (US Millions)", ylab="Medals per Population (Millons)", main="Medals per Poulation vs GDP")

plot(dataset$Medals.per.Pop, dataset$Medals.per.GDP, pch=16, col="blue", xlab ="Medals per Population (millons)",  ylab= "Medals per GDP (Us Millions)", main="Medals per GDP vs Population")

This will produce a plot similar the image below.



As you can see in the image above the data is right skewed. Lets see if we can normalize the distribution of values by putting them onto a logarithmic scale

**Logarithmic scale**

In order to get the two new features onto a logarithmic scale lets alter the R code as follows.

*#Add new features to the data set which show medal count per million people and medal count per million US GDP*

*#Put values on logarithmic scale.*

dataset$Medals.per.Pop <- log(dataset$`Medal Count` / (dataset$Population/1000000))

dataset$Medals.per.GDP <- log(dataset$`Medal Count` / (dataset$GDP))

*#Create a two panel plot for our histograms and scatterplots*

par(mfrow = c(2,2))

#Histograms

hist(dataset$Medals.per.Pop, xlab="Medals per million people", main="Histogram of Medals per Pop (MM)", breaks =20)

hist(dataset$Medals.per.GDP, xlab="Medals per million US$ GDP", main="Histogram of Medals per GDP (US MM)", breaks=20)

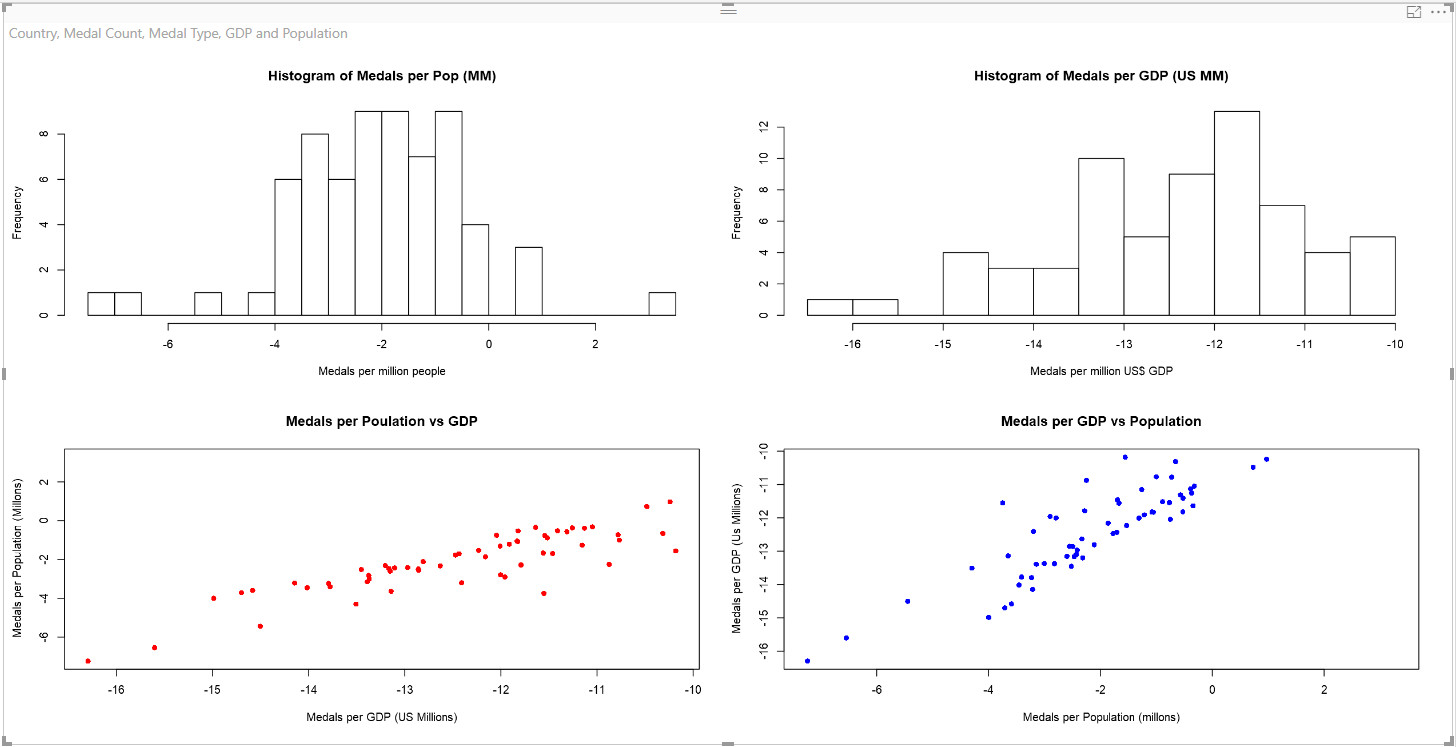
#Scatterplots

plot(dataset$Medals.per.GDP, dataset$Medals.per.Pop, pch=16, col="red", xlab = "Medals per GDP (US Millions)", ylab="Medals per Population (Millons)", main="Medals per Poulation vs GDP")

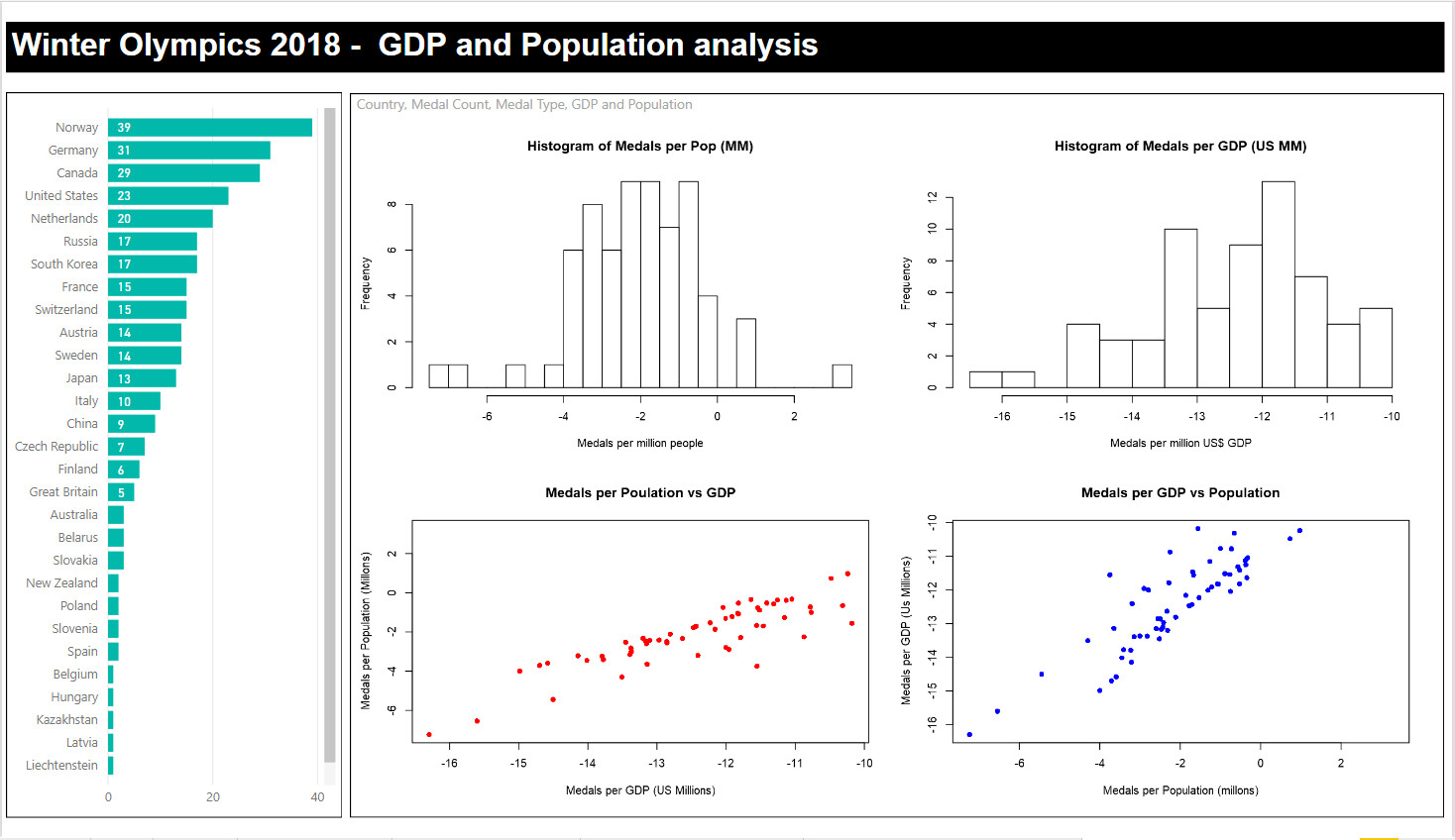
plot(dataset$Medals.per.Pop, dataset$Medals.per.GDP, pch=16, col="blue", xlab ="Medals per Population (millons)",  ylab= "Medals per GDP (Us Millions)", main="Medals per GDP vs Population")

As you can see from the code above to put the values on the log scale we simply called the log function when assigning the values to data frame factors dataset$Medals.per.Pop and dataset$Medals.per.GDP.

This produces the following visual.



As you can see its very easy to use R and Power BI to explore and manipulate your data. Fee free to add slicers and other visuals to the canvas of the report to make it look more visually appealing and interactive. Here is what my report looks like.



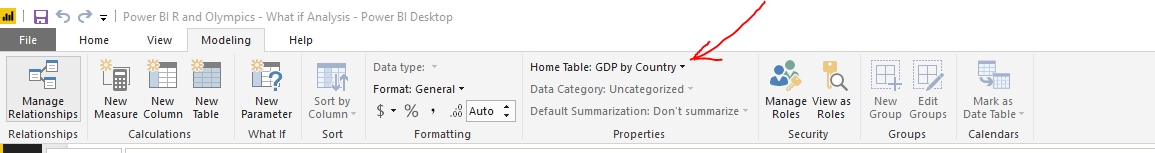
**Move the measures**

In the previous exercise, we created two new measures; one called **WhatIfGDP**on the **GDP by Country** table and one called **WhatIfPop**on the **Sovereign states and dependencies by population[edit]**table.

The first thing we are going to do is move these two measures onto the **Medal Table[edit],**we will do this so that it is easier to use with the R visual. Thankfully Power BI has a handy option called Home Table: that will allow us to easily put these measure in a different table.

**WhatIfGDP**

Select the **WhatIfGDP**measure and change the home table to **Medal Table[edit].**



Once the measure has been moved we will need to modify the code as follows.

WhatIfGDP =

VAR

  Country = SELECTEDVALUE('Number of athletes by National Olympic Committee[edit]'[Country],"All")

RETURN

IF

(

    Country = "All",

    SUMX('Medal table[edit]', RELATED('GDP by Country'[GDP]) \* [What If GDP Value]),

    SUMX('Medal table[edit]', RELATED('GDP by Country'[GDP]) \* IF(Country = 'Medal table[edit]'[Country], [What If GDP Value],1))

)

This code uses the **GDP**value from the **GDP by Country** table by bringing it into the **Medal Table[edit]** table using the DAX function RELATED. Next we will move the **WhatIfPop**using the same Home table option and then modify the DAX.

**WhatIfPop**

After the **WhatIfPop**measure is moved to the **Medal Table[edit]** by using the Home Table: option modify the R code as follows.

WhatIfPop = VAR

  Country = SELECTEDVALUE('Number of athletes by National Olympic Committee[edit]'[Country],"All")

RETURN

IF(Country = "All",

    SUMX('Medal table[edit]', RELATED('Sovereign states and dependencies by population[edit]'[Population]) \* [What if Population Value]),

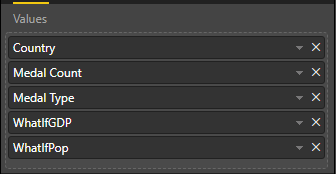
    SUMX('Medal table[edit]', RELATED('Sovereign states and dependencies by population[edit]'[Population])\* IF(Country = 'Medal table[edit]'[Country],[What if Population Value], 1))

)

Now that we have moved the measures and modified the DAX lets use GGPLOT to visualize the What If data.

**GGPLOT**

Make a duplicate of the Power BI page created in the previous article and remove the existing R visual. Drop a new R visual onto the canvas of the report and select the fields **Country**, **Medal Count**, **Medal Type**, **WhatIfGDP**and **WhatIfPop**from the table **Medal table[edit].**



Add the following code to the R visual.

library("ggplot2")

library("scales")

dataset$Medals.per.GDP = log(dataset$`Medal Count` / dataset$WhatIfGDP)

dataset$Medals.per.Pop = log(dataset$`Medal Count` / (dataset$WhatIfPop/1000000))

ggplot(dataset, aes(x=Medals.per.GDP, y=Medals.per.Pop))  +

 geom\_point( aes(colour=`Medal Type`, size=3), show.legend=FALSE) +

 scale\_colour\_manual(values = c("orange", "gold", "gray")) +

 facet\_wrap(~ `Medal Type` )+

 scale\_x\_continuous(labels = comma)+

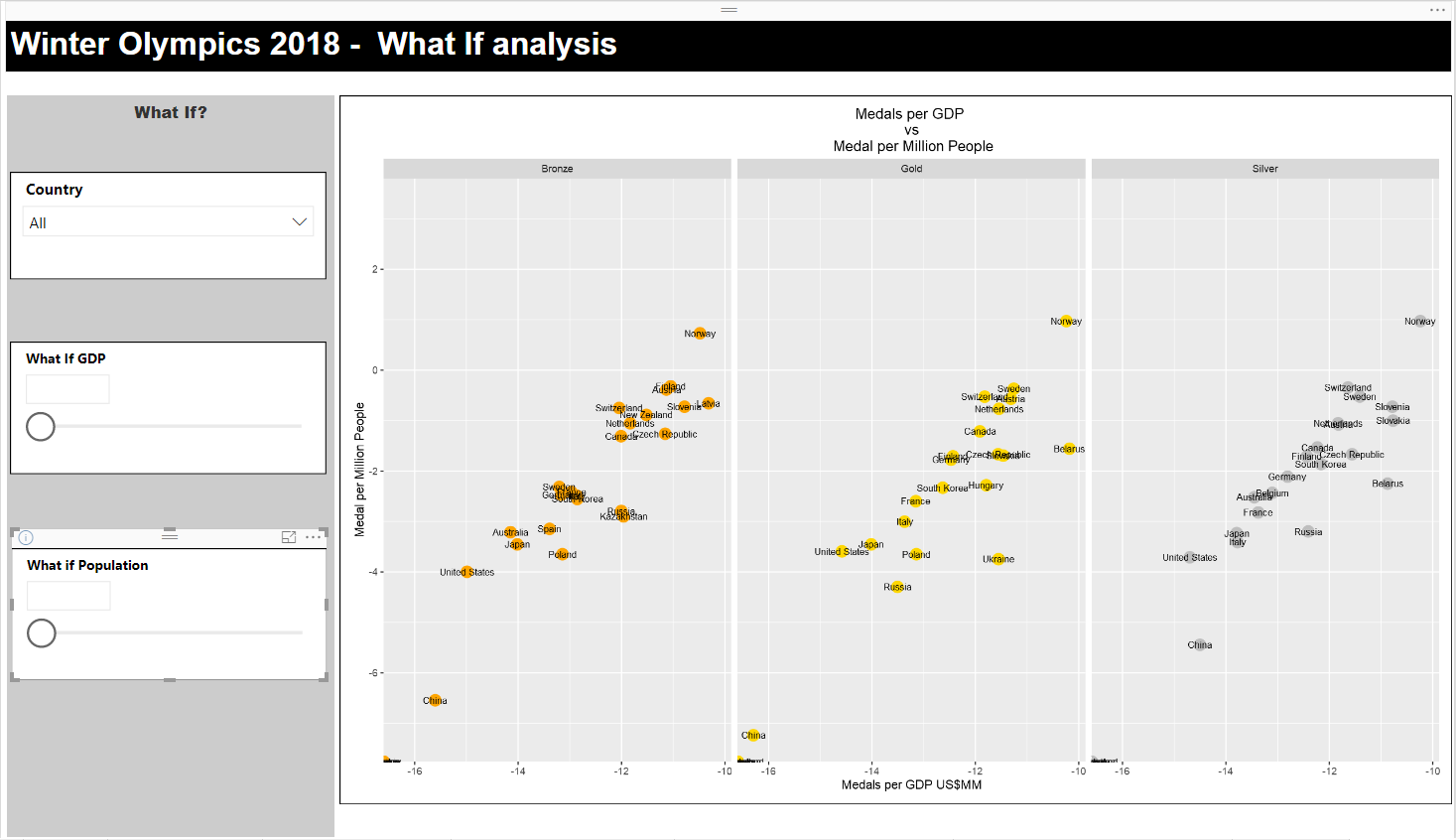
 geom\_text(aes(label=Country), size = 3)+

 ggtitle("Medals per GDP \n vs \n Medal per Million People") +

 xlab("Medals per GDP US$MM") + ylab("Medal per Million People")

If you have been following this series of articles most of the above code should look familiar. I added two new features to the data set based on the What If measures and put them on a logarithmic scale. I am then using GGPLOT to visualize this data.

Your report should look similar to the image below.



The cool thing with this report is that you can pick a country and adjust the GDP and Population for the selected country to see how their standings change in the R visual. For example select Canada and set the Population to 10% and you will see how Canada moves ahead.

**Predictive Modelling:**

In this article I will show you how you can incorporate R predictive models into your Power BI report to instantly visualize different what if scenarios with predictions in your report.

**Manipulate the data**

In order to train predictive model we will need to clean up our data slightly. If you have been exploring the data set in the report you might have noticed that we are missing GDP data from Great Britain and Liechtenstein.

In Power Query we will add new columns to the **GDP by Country**and **Sovereign states and dependencies by population[edit]** tables. In each table we will add a new column called **Country Olympic Names**and use this to join.

Modify the M code associated with the **GDP by Country** table so that it looks as follows.

let

    Source = Web.Page(Web.Contents("https://en.wikipedia.org/wiki/List\_of\_countries\_by\_GDP\_(nominal)")),

    Data1 = Source{1}[Data],

    #"Changed Type" = Table.TransformColumnTypes(Data1,{{"Rank", type text}, {"Country", type text}, {"GDP #(lf)(US$MM)", type text}}),

    #"Filtered Rows" = Table.SelectRows(#"Changed Type", each ([Rank] <> "" and [Rank] <> "—")),

    #"Split Column by Delimiter" = Table.SplitColumn(#"Filtered Rows", "GDP #(lf)(US$MM)", Splitter.SplitTextByEachDelimiter({"/"}, QuoteStyle.Csv, true), {"GDP #(lf)(US$MM).1", "GDP #(lf)(US$MM).2"}),

    #"Changed Type1" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"GDP #(lf)(US$MM).1", type number}, {"GDP #(lf)(US$MM).2", type text}}),

    #"Removed Columns" = Table.RemoveColumns(#"Changed Type1",{"GDP #(lf)(US$MM).2"}),

    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"GDP #(lf)(US$MM).1", "GDP"}}),

    #"Added Conditional Column" = Table.AddColumn(#"Renamed Columns", "Custom", each if [Country] = "United Kingdom" then "Great Britain" else [Country], type text),

    #"Renamed Columns1" = Table.RenameColumns(#"Added Conditional Column",{{"Custom", "Country Olympic Names"}}),

    #"Appended Query" = Table.Combine({#"Renamed Columns1", Liechtenstein})

in

    #"Appended Query"

Similarly modify the M code associated with the **Sovereign states and dependencies by population[edit]** table as follows.

let

    Source = Web.Page(Web.Contents("https://en.wikipedia.org/wiki/List\_of\_countries\_and\_dependencies\_by\_population")),

    Data0 = Source{0}[Data],

    #"Changed Type" = Table.TransformColumnTypes(Data0,{{"Rank", type text}, {"Country #(lf)(or dependent territory)", type text}, {"Population", type number}, {"Date", type date}, {"% of world #(lf)population", Percentage.Type}, {"Source", type text}}),

    #"Added Conditional Column" = Table.AddColumn(#"Changed Type", "Custom", each if [#"Country #(lf)(or dependent territory)"] = "United Kingdom" then "Great Britain" else [#"Country #(lf)(or dependent territory)"], type text),

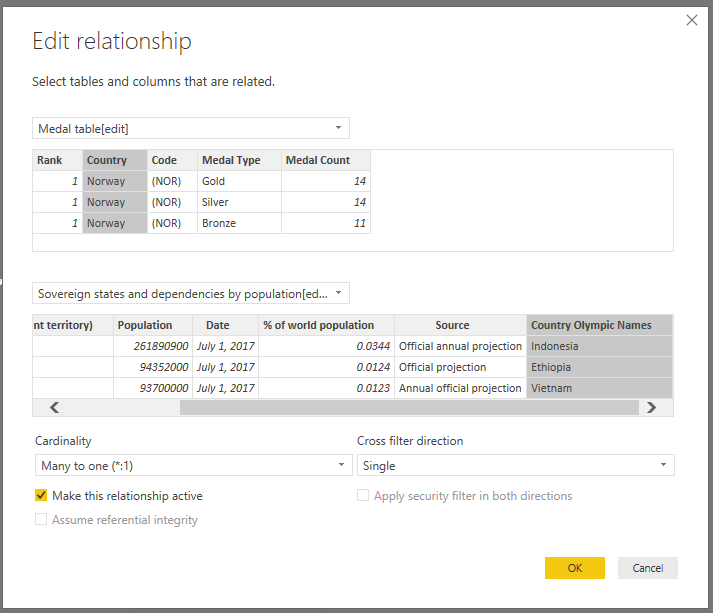
    #"Renamed Columns" = Table.RenameColumns(#"Added Conditional Column",{{"Custom", "Country Olympic Names"}})

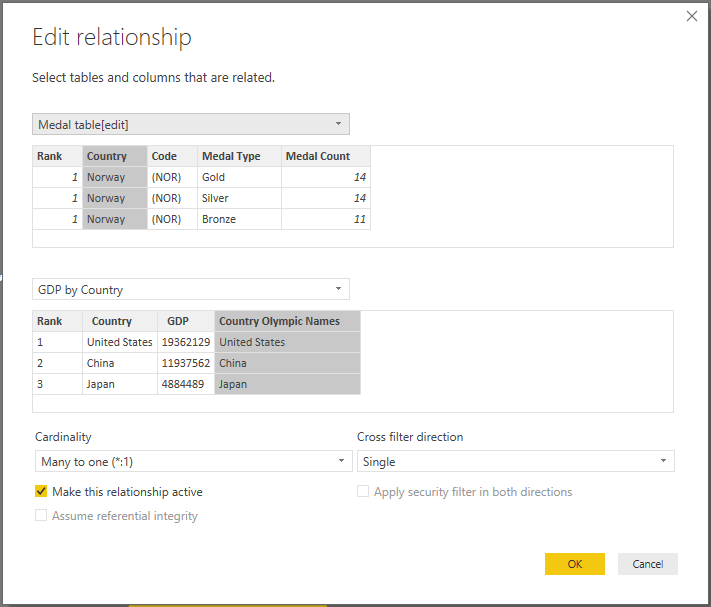
in

    #"Renamed Columns"

**Change Relationships**

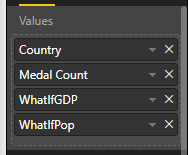
Change the existing relationships so that the new **Country Olympic Names** are used to join the tables **Medal table[edit], GDP by Country**and **Sovereign states and dependencies by population[edit].**





**R visual**

Copy the page from the previous report and remove the existing R visual. Drop a new R visual onto the canvas of the report and select the fields **Country, Medal Count, WhatifGDP, WhatIfPop.**



Add the following lines of code into this visual.

library("ggplot2")

library("scales")

require("reshape")

dataset$Medals.per.GDP = log(dataset$`Medal Count` / dataset$WhatIfGDP)

dataset$Medals.per.Pop = log(dataset$`Medal Count` / (dataset$WhatIfPop/1000000))

model\_lrm <- lm(`Medal Count` ~ Medals.per.GDP + Medals.per.Pop, data = dataset)

dataset$Predicted.Medal.Count <- round(predict(model\_lrm, dataset),0)

data.reshaped<-  melt(dataset[,c("Country","Medal Count", "Predicted.Medal.Count")])

ggplot(data.reshaped, aes(Country, value)) +

  geom\_bar(aes(fill = variable), position = "dodge", stat="identity")

If you have been following this series of article most of the code should look familiar as such I will not break down every line of code but just focus on the changes.

require("reshape")

I've added this new library in order to use the melt function. I need to change the shape of the data in order to plot a grouped bar chart with GGPLOT.

model\_lrm <- lm(`Medal Count` ~ Medals.per.GDP + Medals.per.Pop, data = dataset)

This line of code trains a linear regression model which attempts to predict the number of medals by using the Medals.per.GDP and Medals.per.Pop features in the dataset.

dataset$Predicted.Medal.Count <- round(predict(model\_lrm, dataset),0)

data.reshaped<-  melt(dataset[,c("Country","Medal Count", "Predicted.Medal.Count")])

The first line of code adds a new prediction column to our dataset by using the predict function with our model and data. The second line of code unpivots the data using the melt function so that I have one row for the actual number of medals and one row for the predicted number of medals.

ggplot(data.reshaped, aes(Country, value)) +

  geom\_bar(aes(fill = variable), position = "dodge", stat="identity")

This last line of code plots the data using a grouped bar chart.

Once you have completed the code see if you can rearrange the page so that you have a report that looks similar to the following.

See how the plot changes as you modify the different What If parameters.