Integrating Power BI with PowerApps and Flow

Lab Time: 60 minutes

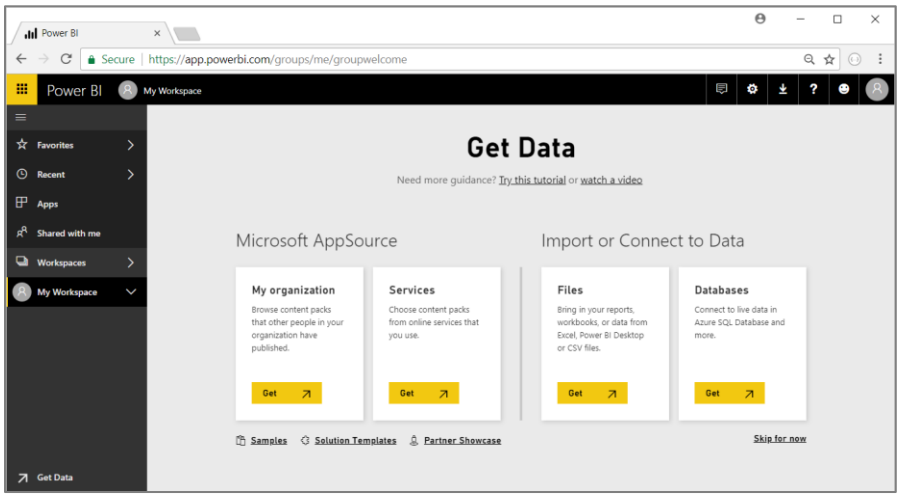
Lab Folder: C:\Student\Modules\09_PowerBI\Lab

Lab Overview: In this lab, you will learn how to begin working with reports and dashboards in Power BI. Then you will learn how to embed a Power BI dashboard tile in an app you create with PowerApps. In the final exercise, you will create a streaming dataset in Power BI and then populate this streaming dataset using a flow that listens on Twitter for incoming tweets with a specific hashtag. This will allow you to create a real-time dashboard which monitors Twitter for specific tweets of interest.

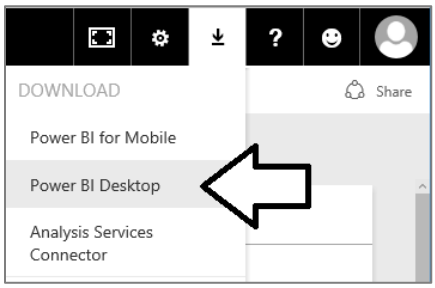
Exercise 1: Adding Power BI Content to a New App Workspace

In this exercise, you will first download and install Power BI Desktop if you have not already done so. Note that if Power BI desktop is already installed on your student workstation, you can skip ahead in this exercise to step 2.

1. Log into the Power BI Service.
   a) Using the browser, navigate to the Power BI service at https://app.powerbi.com.
   b) Sign in using your Office 365 trial account.
   c) If you have not yet added any Power BI content, you should see the Get Data page as shown in the following screenshot which indicates your personal workspace currently contains no reports or dashboards.



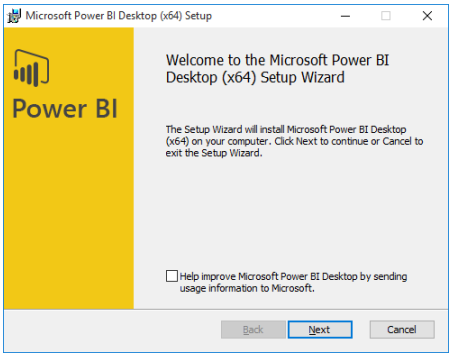
2. Install Power BI Desktop
   a) On the top right of the Power BI service window, drop down the **Downloads** menu and click the **Power BI Desktop** menu command to begin the download of the installation file.



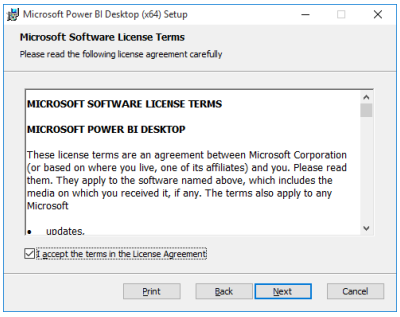
   b) Wait for the MSI file to download.

c) Once the file has downloaded, click the **Run** button to begin the installation of Power BI Desktop.

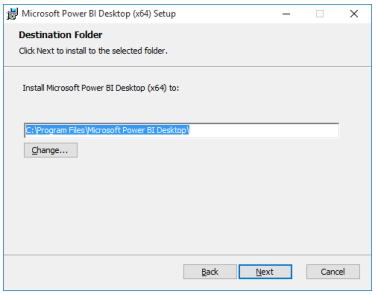
d) When you see the Welcome screen, click **Next** to continue with the installation.



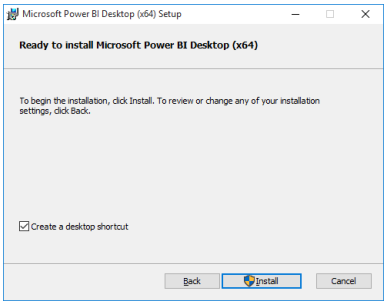
e) Click the checkbox to accept the license agreement and click **Next**.



f) Accept the default location for the installation and click **Next**.



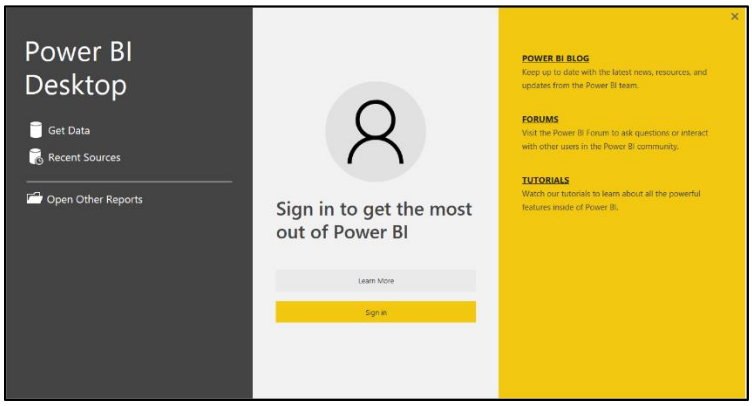
g) On the next screen, click **Install**.



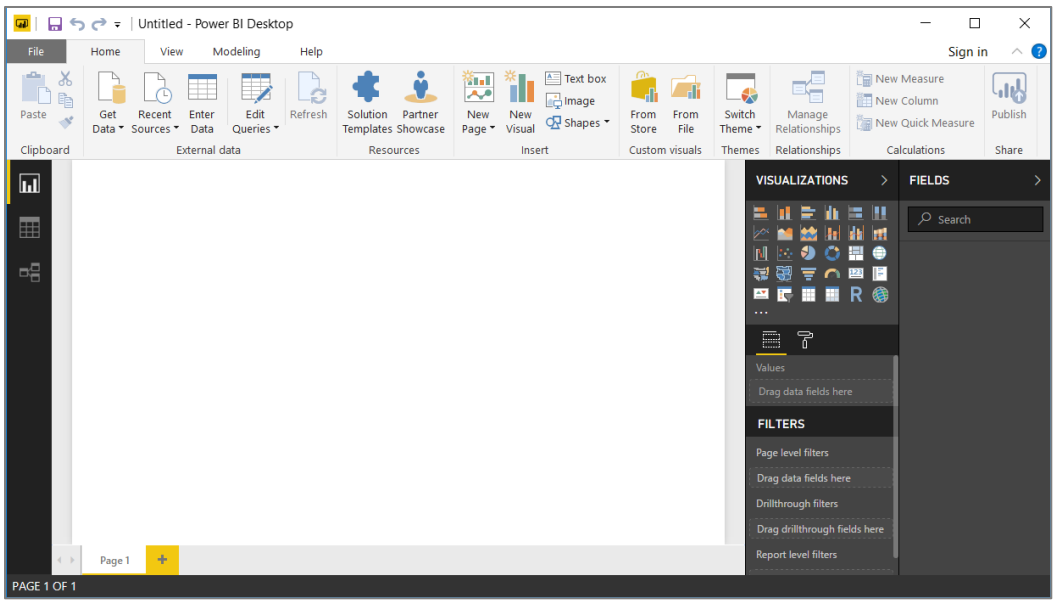
h) When you see **Completed the Microsoft Power BI Desktop Setup Wizard** screen, click **Finish** to launch Power BI Desktop.

i) When Power BI Desktop launches for the first time, it displays a Welcome screen as shown in the following screenshot.

j) Click the (**X**) button in the upper right corner to close this window.

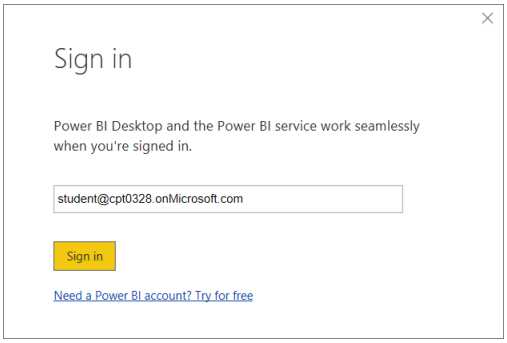


k) At this point, you should have Power BI Desktop running with a new, unsaved project as shown in the following screenshot.



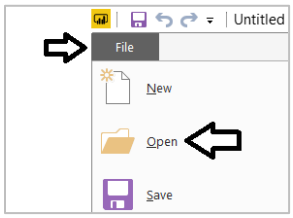
You can start this exercise here if Power BI Desktop was already installed.

3. Sign In with Power BI Desktop using your Office 365 trial account
   a) Click the **Sign In** link in the upper right corner of the Power BI Desktop window
   b) Sign in using your Office 365 trial account.



   c) When prompted, enter your password to sign in.
   d) Verify that you have successfully logged in.

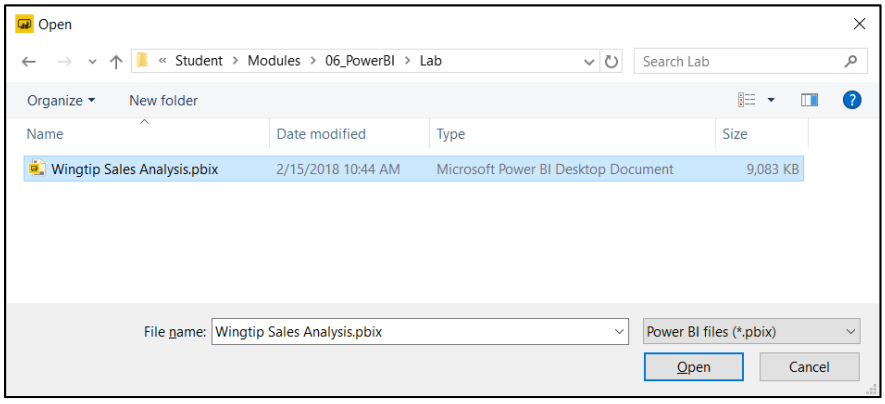
4. Open the Power BI Desktop project file named **Wingtip Sales Analysis.pbix**.
   a) Select he **File > Open** command from within Power BI Desktop.



   b) Locate the PBIX file located at the following path.

```
C:\Student\Modules\09_PowerBI\Lab\Wingtip Sales Analysis.pbix
```

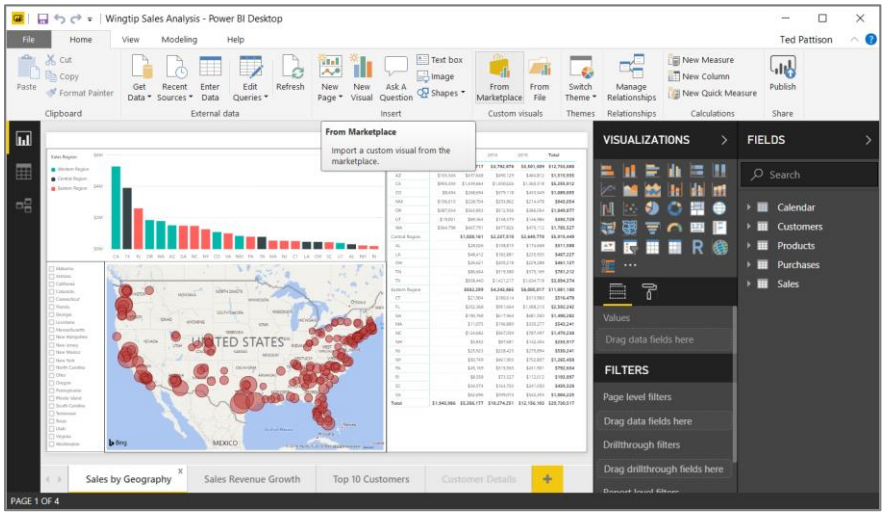
   c) Open **Wingtip Sales Analysis.pbix** to load this project into Power BI Desktop.



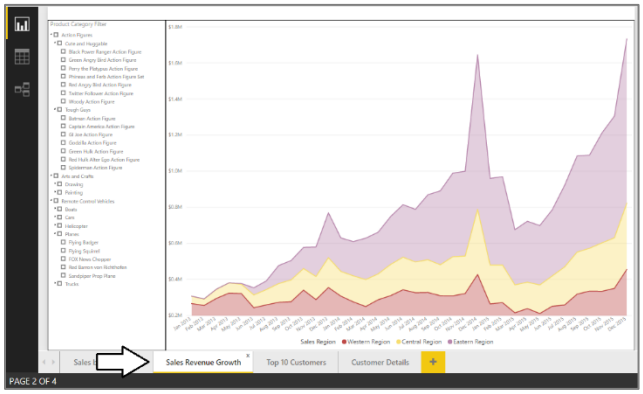
The project should now be open in Power BI desktop.

5. Inspect the contents of the Power BI Desktop project named **Wingtip Sales Analysis.pbix**.

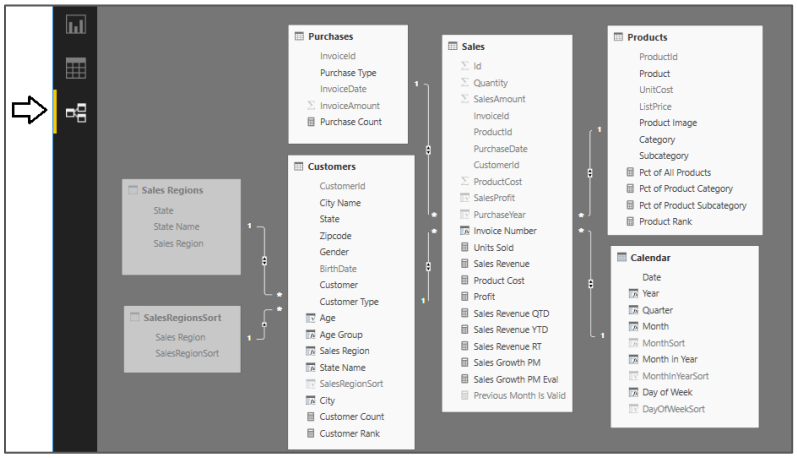
a) Inspect the report that has been created inside this project. You should see it provides four pages.



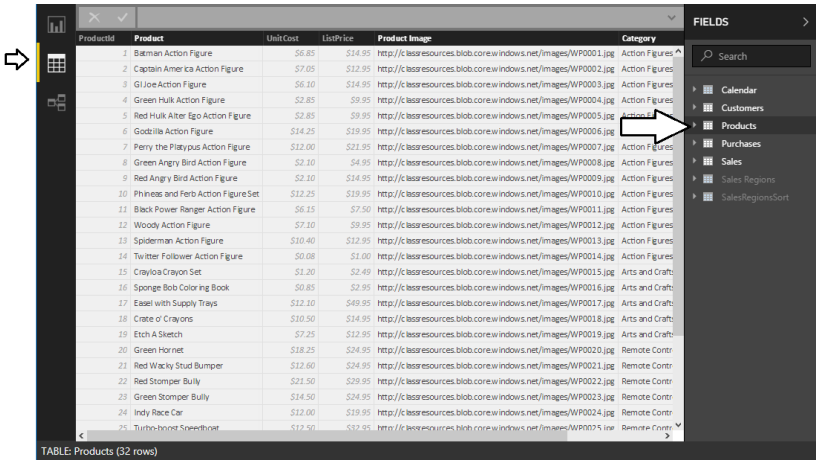
b) Using the navigation tabs at the bottom of the report, move from page to page to inspect each page in the report.



c) Click on the Relationship view button in the left navigation to see the tables included in data model and their relationships.



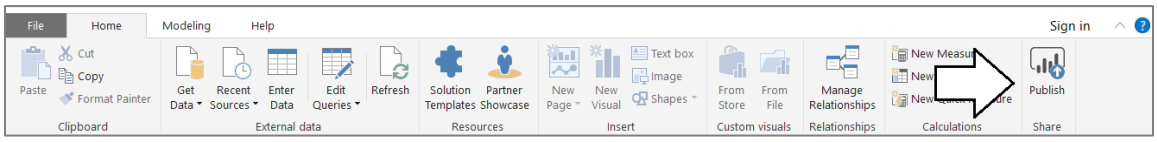
d) Click on the Data view button in the left navigation to see a tabular view of the data inside the project's data model. Note that you can select a table in the FIELDS list on the right to see the data in that table.



You do not need to make any changes to the Power BI Desktop project named **Wingtip Sales Analysis.pbix**. The purpose of this lab is for you to open an existing project that has already been completed and then to publish it to your personal workspace.

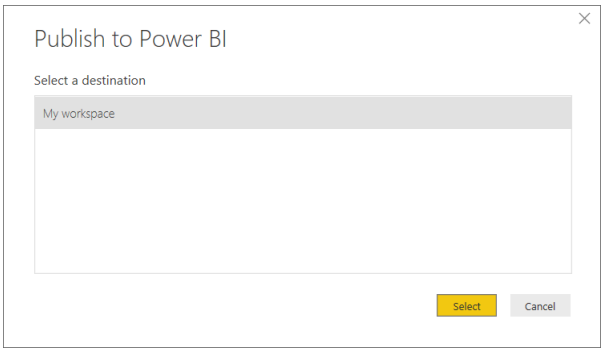
6. Publish the **Wingtip Sales Analysis.pbix** project to the Power BI Service.

a) Navigate to the **Home** tab in the ribbon and click the **Publish** button on the far right-hand side.

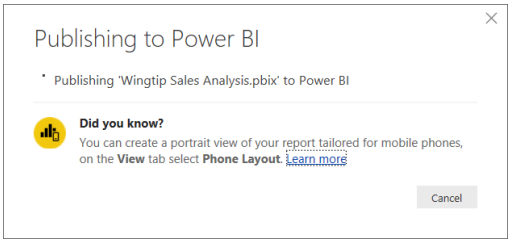


b) When prompted for your password, sign into the Power BI service.

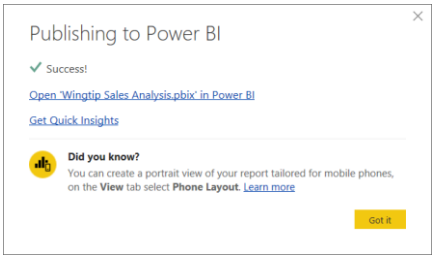
c) When Power BI Desktop prompts you with the **Publish to Power BI** dialog, select **My workspace** and then click **Select**.



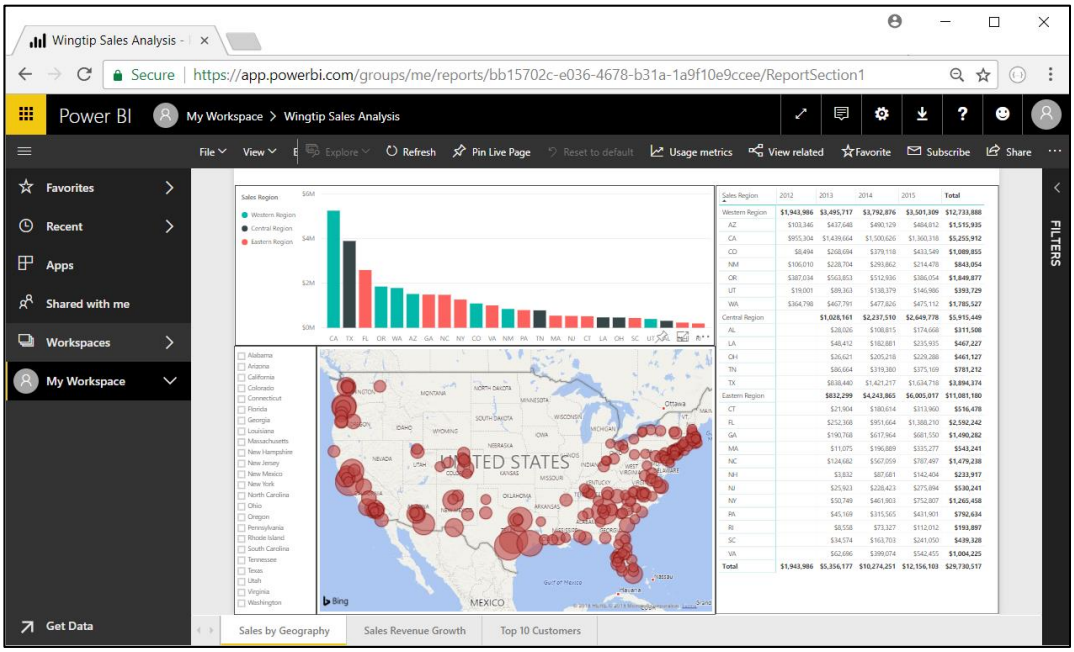
d) Power BI Desktop will display the **Publishing to Power BI** dialog as the publishing process begins.



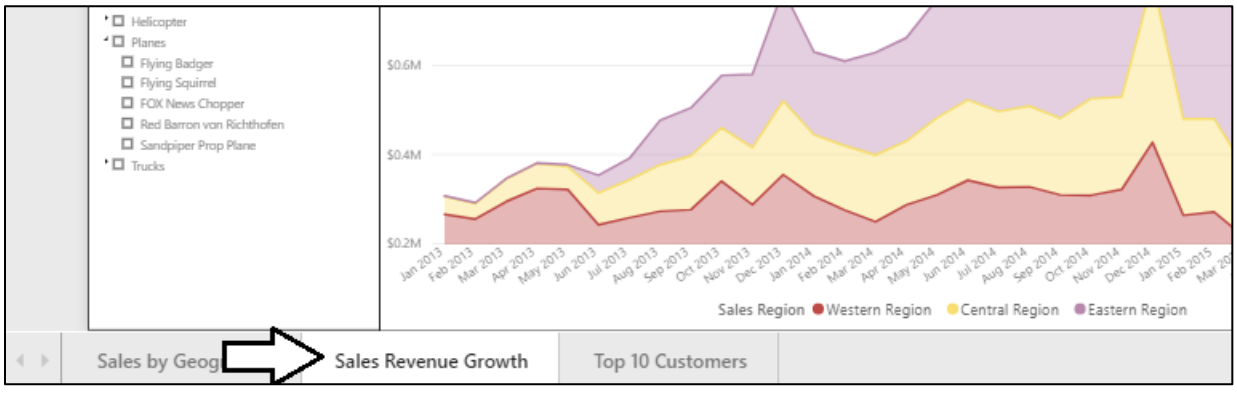
e) Once the publishing process has completed, the **Publishing to Power BI** dialog will display a success message and provide you with a link to **Open Wingtip Sales Analysis.pbix in Power BI**. Click on that link to navigate to the Power BI service using the browser.



f) You should now be able to see the **Sales by Geography** page of the report you just published.

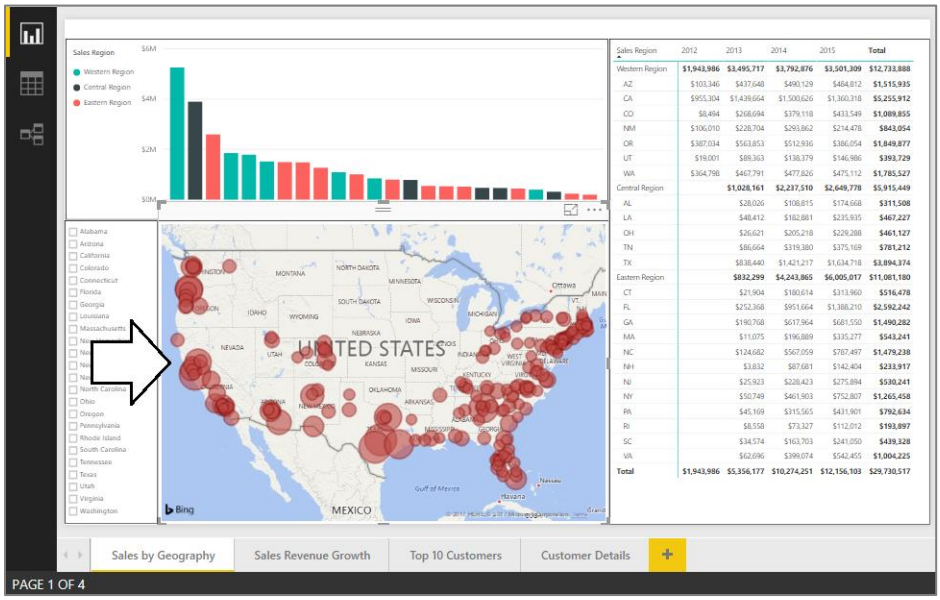


g) Click on the **Sales Revenue Growth** link at the bottom of the screen to see the second page of the report.

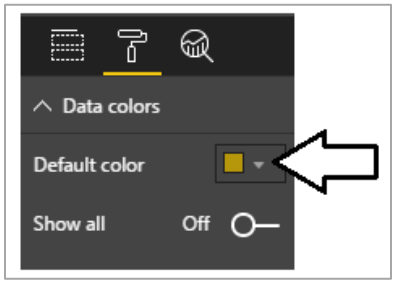


You have now successfully published a PBIX project using Power BI Desktop. But what happens when you want to make a change to a report after it has been published? It's very easy because you can make changes to your Power BI Desktop project and republish it on top of a previous version of the same project that has already been published.

7. Change the type of the visual that displays sales revenue by month and purchase type.
   a) Navigate back to Power BI Desktop and the project named **Wingtip Sales Analysis.pbix**.
   b) Return to the **Sales by Geography** page.
   c) Select the **Map** visual.



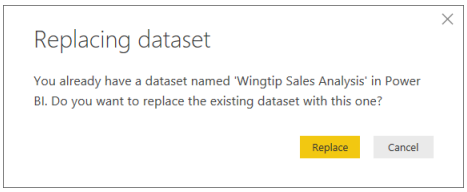
   d) Update the **Default color** property in the **Data colors** section in the **Format pane** to change the color of the bubbles from red to a different color such as yellow or purple.



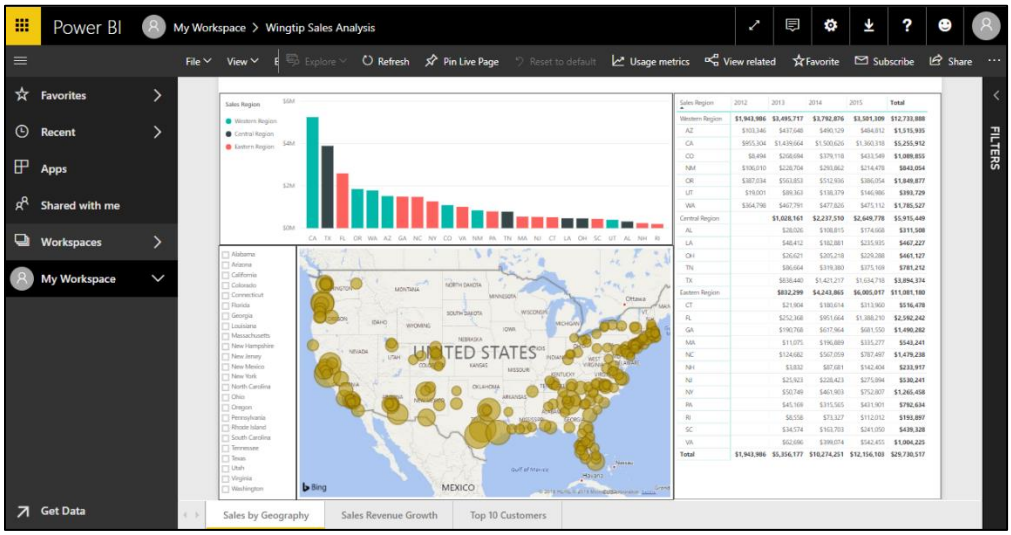
   e) Verify that the bubbles in the Map visual are now a different color than red.

   f) Save your changes to **Wingtip Sales Analysis.pbix**.

8. Republish the project to the Power BI service.
   a) Click the **Publish** button on the far right-hand side of the **Home** tab in the ribbon.
   b) When Power BI Desktop prompts you with the **Publish to Power BI** dialog, select **My workspace** and then click **Select**.
   c) When prompted with the **Replacing dataset** dialog, click Replace to begin the publishing process.

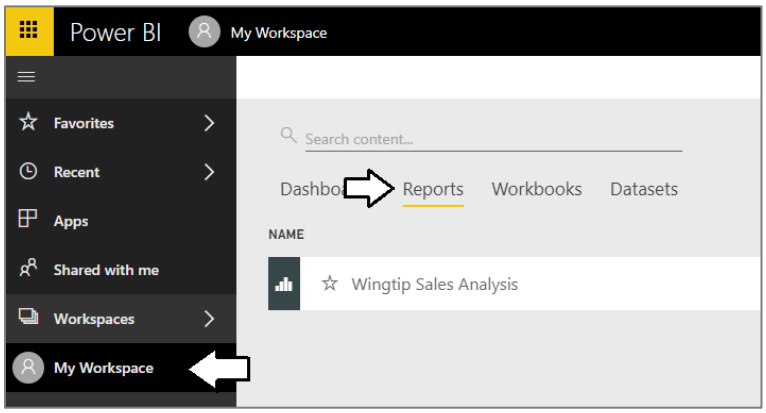


   d) Once the publishing process has completed, inspect the published report in the Power BI service using the browser. Verify that the bubble color within the Map visual has been updated.

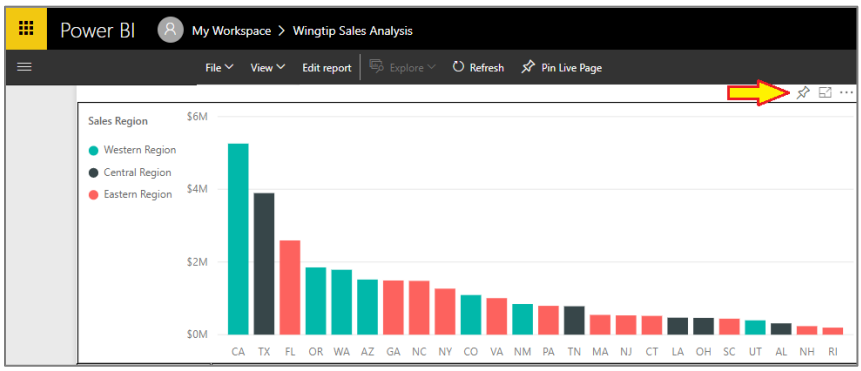


over the next few steps, you will create a new dashboard and then add tiles by pinning visuals from the report you just published.

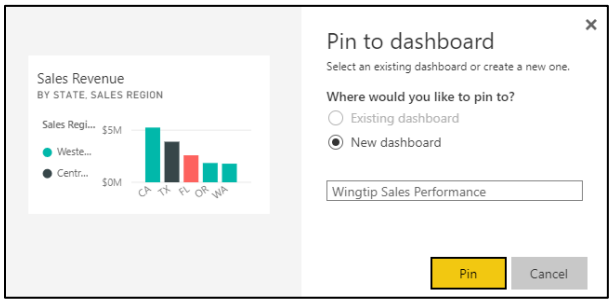
9. Create the Wingtip Sales Performance dashboard.
   a) Navigate to the **Wingtip Sales Analysis** report.



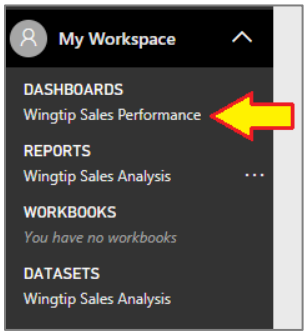
   b) Make sure the first page named **Sales by Geography** is the active page in the report.
   c) Click the thumbtack icon on the column chart to pin the visual to a dashboard.



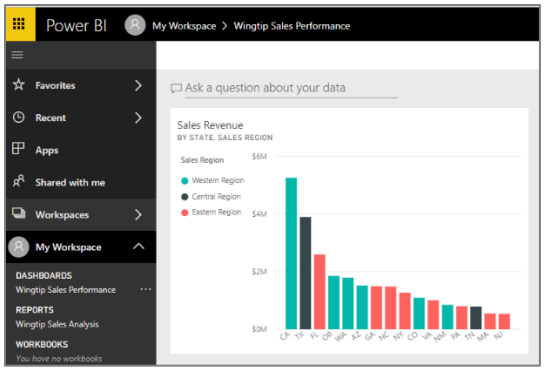
d) The **Pin to dashboard dialog** prompts you enter name for a new dashboard name. Enter a value of **Wingtip Sales Performance** as the new dashboard name and then click the **Pin** button to create the new dashboard and pin the visual to it.



e) At this point, you should be able to see the new **Wingtip Sales Performance** dashboard in the **Dashboards** section

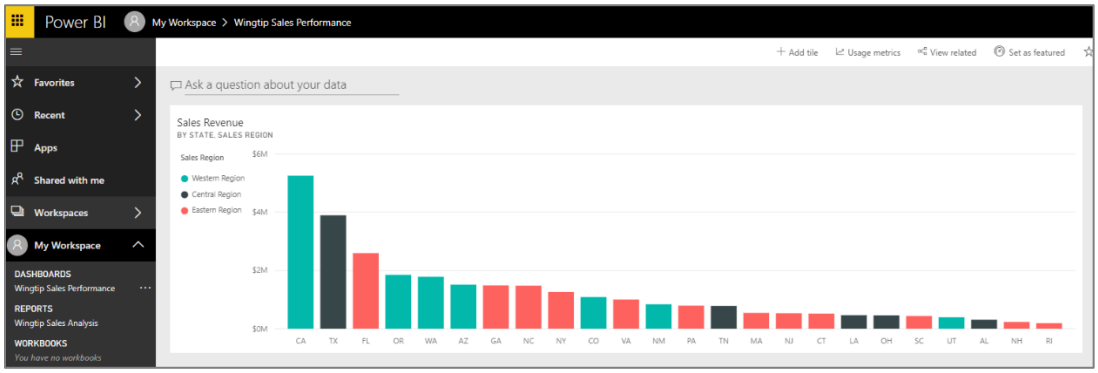


f) Click on the link for the **Wingtip Sales Performance** dashboard in the **Dashboards** section to examine the new dashboard.



g) The dashboard tile you have created is a little too narrow.

h) Use the mouse to resize the new dashboard tile to be twice its original width.

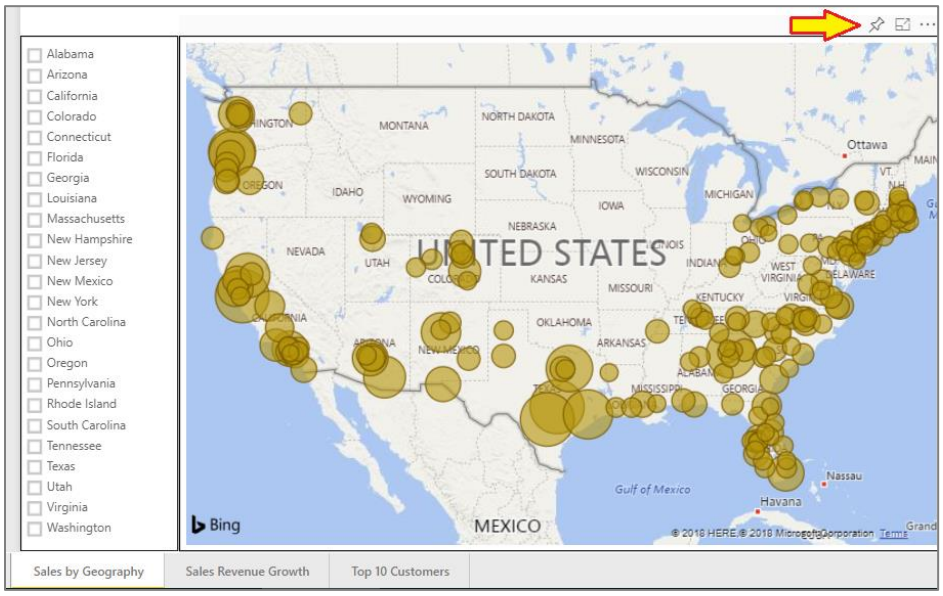


10. Pin a second visual from the **Sales by Geography** page to the **Wingtip Sales Performance** dashboard.

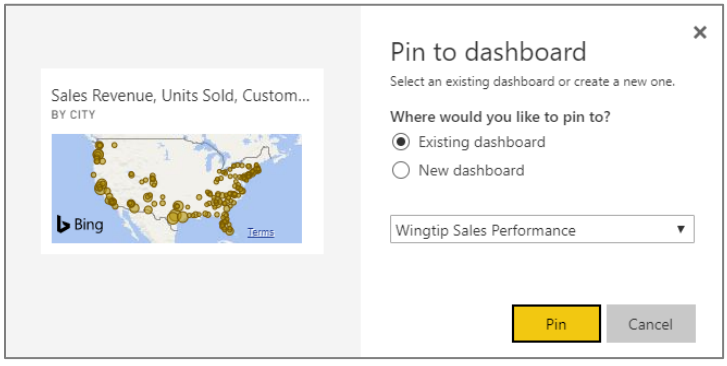
a) Navigate to the **Wingtip Sales Analysis** report using the left navigation.

b) Navigate to the **Sales by Geography** page.

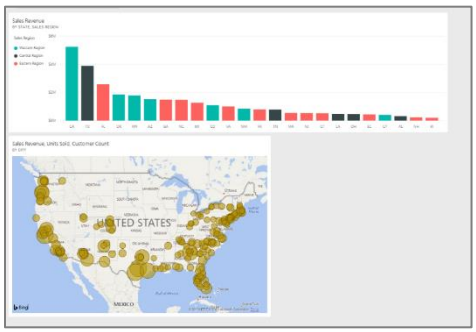
c) Select the map visual and click on the thumbtack icon button to display the **Pin to dashboard** dialog.



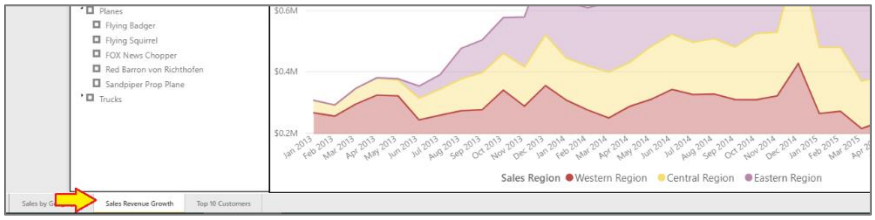
d) In the **Pin to dashboard** dialog, click the **Pin** button to pin the visual to the **Wingtip Sales Performance** dashboard.



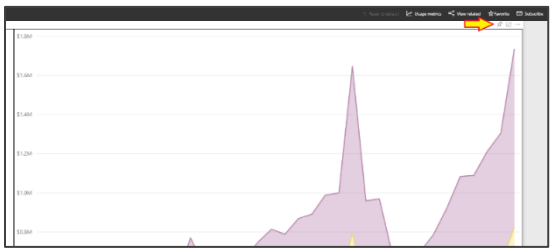
e) Navigate to the **Wingtip Sales Performance** dashboard and confirm a new tile has been added with the map visual.



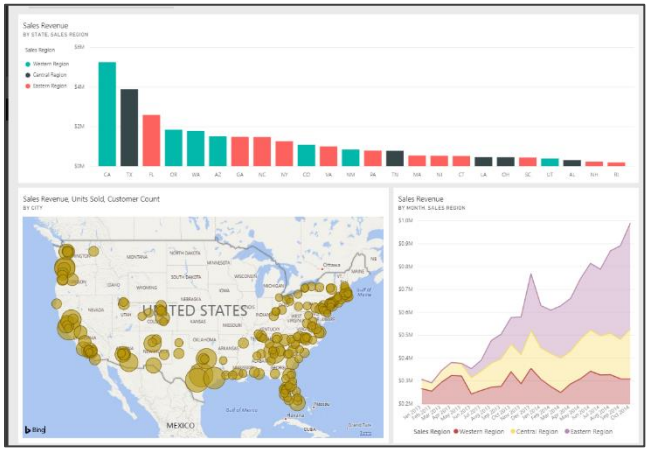
11. Pin a third visual from the **Sales by Geography** page to the **Wingtip Sales Performance** dashboard.
    a) Navigate to the **Wingtip Sales Analysis** report using the left navigation.
    b) Navigate to the **Sales Revenue Growth** page.



    c) Click the thumbtack on the stacked area chart visual

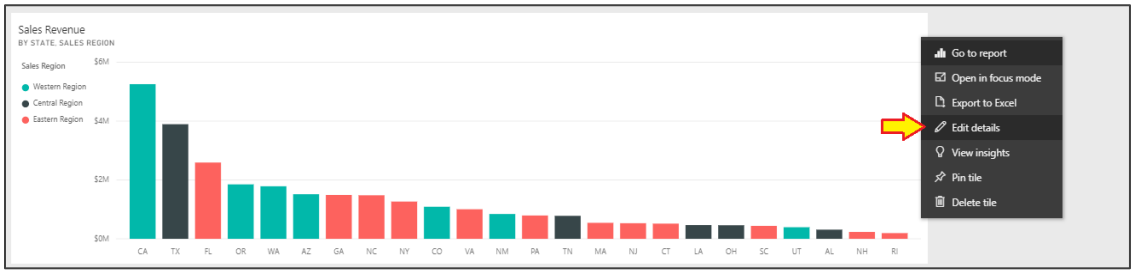


    d) Pin the stacked area chart visual to the Wingtip Sales Performance dashboard.
    e) Rearrange the dashboard tiles to match the layout in the following screenshot.

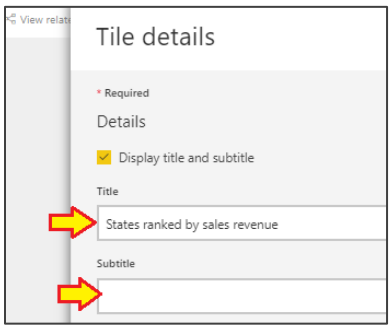


12. Update the **Title** property for all three dashboard tiles.

a) Use the dropdown menu in the upper right corner of the tile with the column chart and select the **Edit details** command.

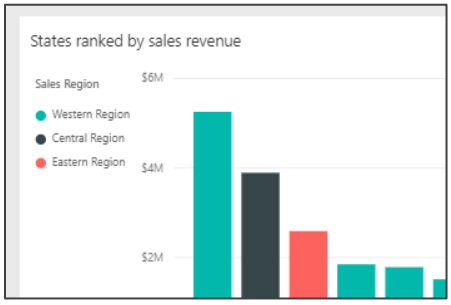


b) Update the **Title** property to **States ranked by sales revenue** and delete the contents of the **Subtitle**.

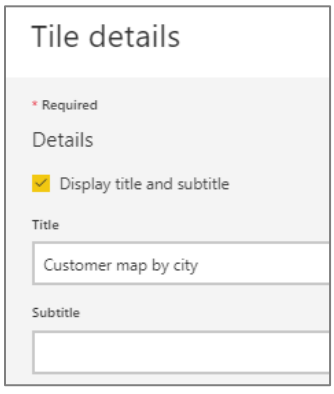


c) Click the **Apply** button at the bottom of the **Tile details** pane to save your changes.

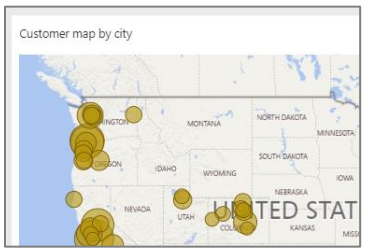
d) Verify that you have successfully changed the title's **Title** property.



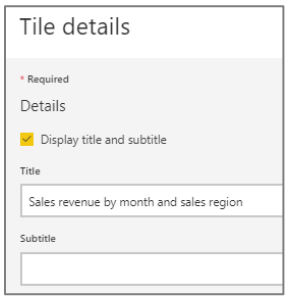
e) Update the **Title** of the map visual to **Customer map by city** and delete the content of the **Subtitle**.



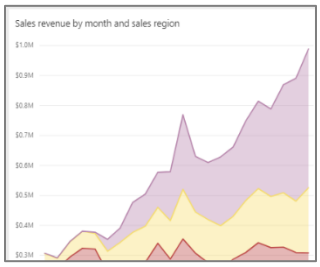
f) Save your changes to the map visual.



g) Update the **Title** of the stacked area chart tile to **Sales revenue by month and sales region** and delete the **Subtitle**.



h) Save your changes to the stacked area chart tile.

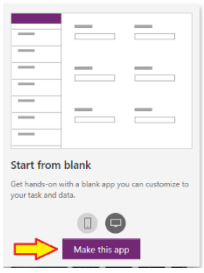


There is an import reason why you updated the **Title** property for each tile. When you embed these tiles in PowerApps, you will reference these tiles using their **Title**.

Exercise 2: Embed Power BI Dashboard Tiles in PowerApps

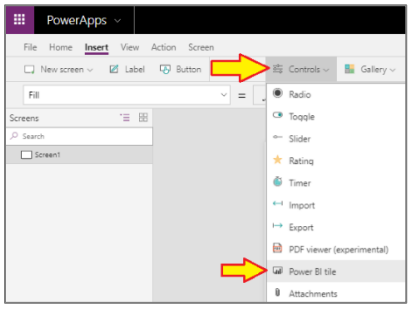
In this exercise, you will create a new app in PowerApps and then you will embed the dashboard tiles you created in the previous exercise using the Power BI tile control.

1. Create a new app in PowerApps using the **Start from blank** template.
   a) In a separate browser window, navigate to PowerApps Studio.
   b) Create a new app using **Tablet** layout and the **Start from blank** template.

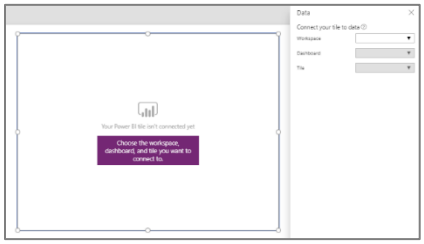


   c) When the app has been created, navigate to the **Insert** tab.

d) Drop down the **Controls** menu and select the **Power BI** tile control.



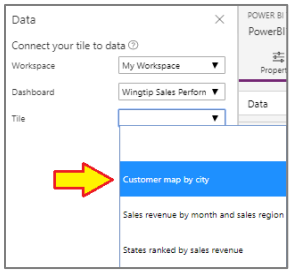
e) Once you have created the tile control, the **Data** pane provides three dropdown menus to configure the data source.



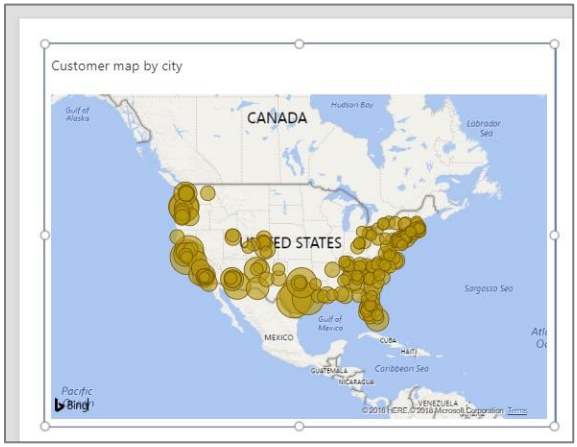
f) Set the **Workspace** setting to **My Workspace**.

g) Set the **Dashboard** setting the **Wingtip Sales Performance**.

h) Set the **Tile** setting to **Customer map by city**.



i) You should now see the dashboard tile with the map in PowerApps Studio.

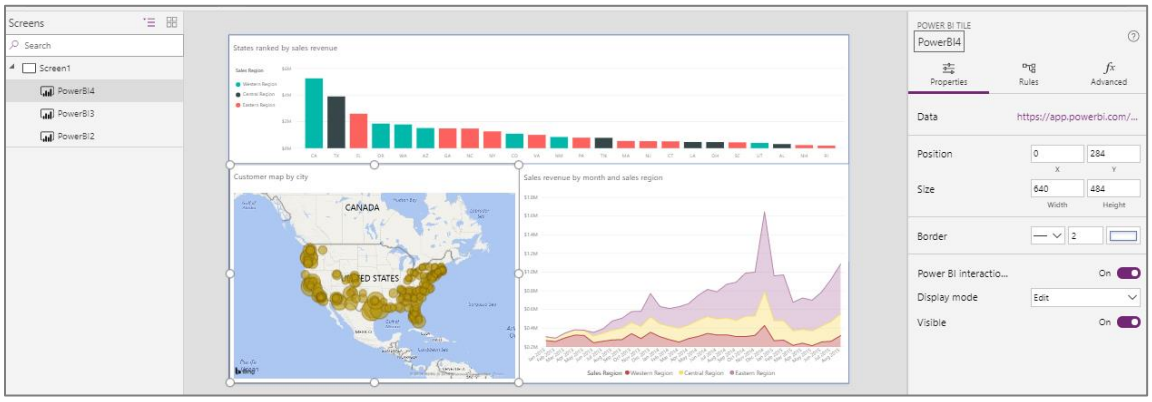


2. Add the two other dashboard tiles.

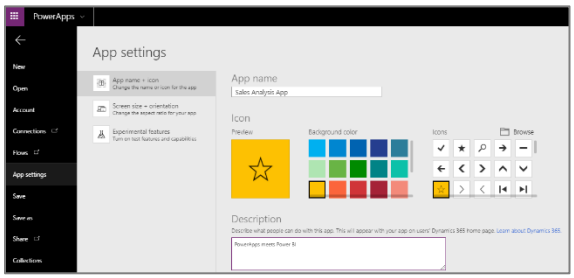
a) Using the same set of steps, add another Power BI tile controls for the **States ranked by sales revenue** tile.

b) Using the same set of steps, add another Power BI tile controls for the **Sales revenue by month and sales regions** tile.

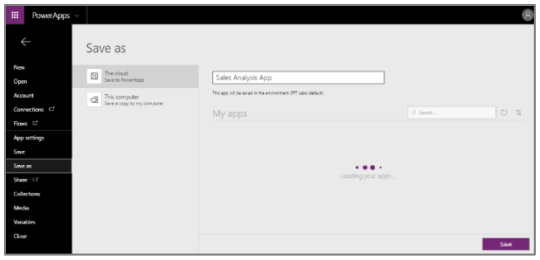
c) Arrange the three tile controls to match the following screenshot.



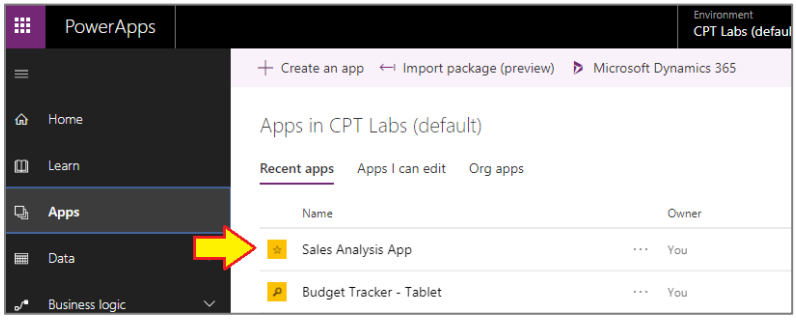
d) Navigate to the **App Setting** tab and give the app a name of **Sales Analysis App**.



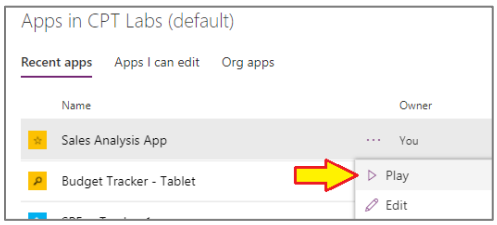
e) Save the app to the cloud



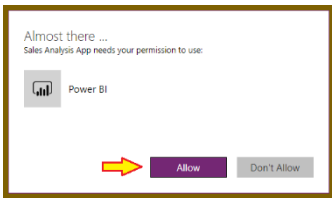
f) Navigate to the Apps tab so you can see the new **Sales Analysis App**.



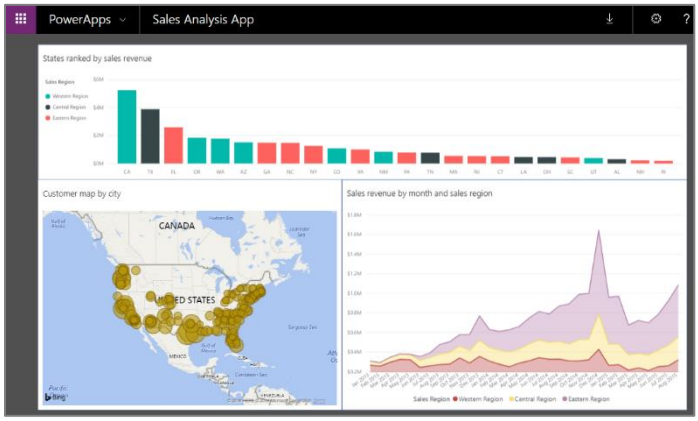
g) Start the **Sales Analysis App** by clicking the **Play** button.



h) When you are prompted to use Power BI, click **Allow**.



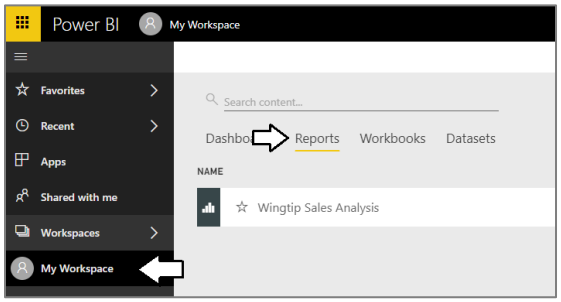
i) Now you can see what your app looks like with embedded Power BI dashboard tiles.



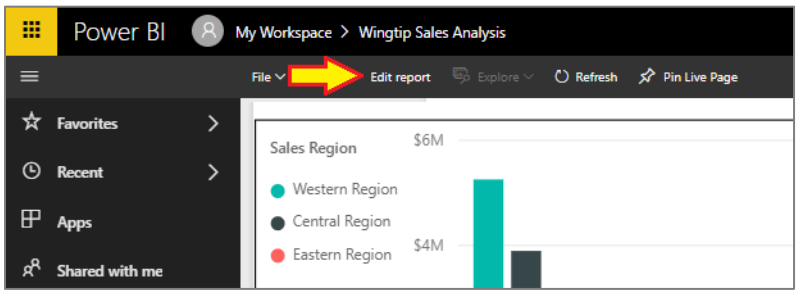
Exercise 3: Extend a Power BI Report using PowerApps

In this exercise, you will create a new page in the **Wingtip Sales Analysis** report and then you will add a PowerApps custom visual to embed an app you will create using PowerApps.

1. Create a new page in the **Wingtip Sales Analysis** report.
   a) Return to the Power BI Service in the browser
   b) Navigate to the **Wingtip Sales Analysis** report.



   c) Click **Edit report** to move the report into edit mode.

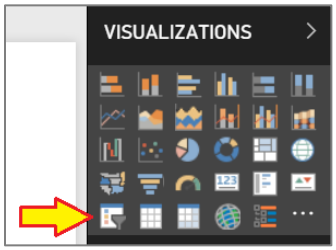


d) Click the **+** button to the right of the page tabs to add a new page.

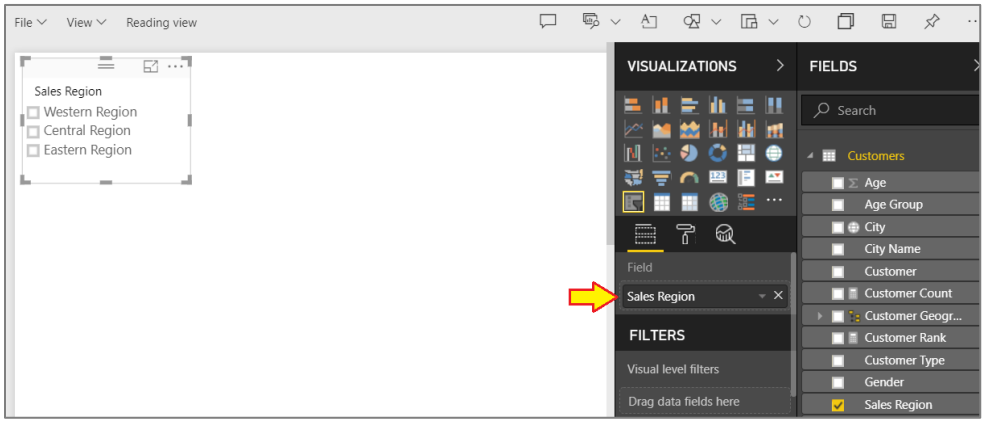
e) Rename the page PowerApps lab.

2. Add a new slicer visual to the page.

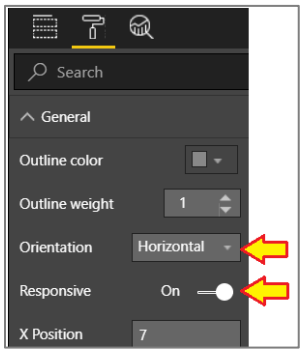
a) In the **Visualization** pane, click the slicer tile with the slicer visual.



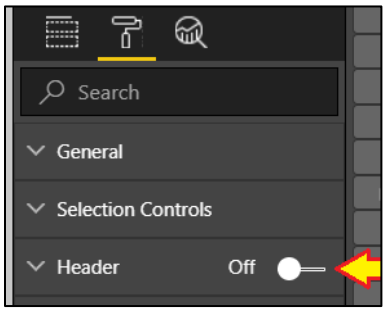
b) Configure the slicer using the **Sales Region** column from the **Customers** table.



c) In the **General** section of the **Properties** pane, set the **Orientation** and **Responsive** properties to the settings shown in the following screenshot.

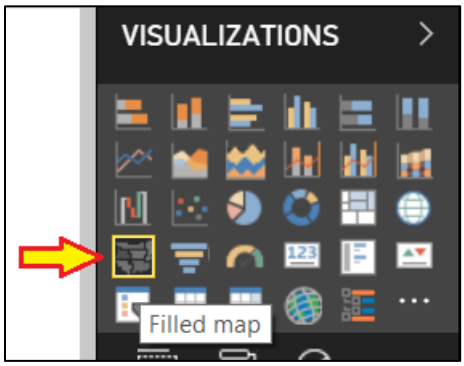


d) Set the **Header** property to **Off**.



e) Reposition the slicer visual as shown in the following screenshot.

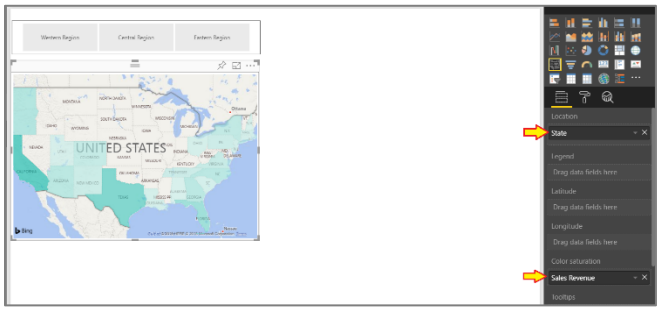
3. Add a **Filled map** visual to the page.
   a) Click on the white background of the report page to make sure the slicer visual is not selected.
   b) Click on the **Filled map** tile in the **Visualizations** pane to create a new filled map visual.



c) Using the **Fields** list, drag and drop the **State** column from the **Customers** table into the **Location** well.

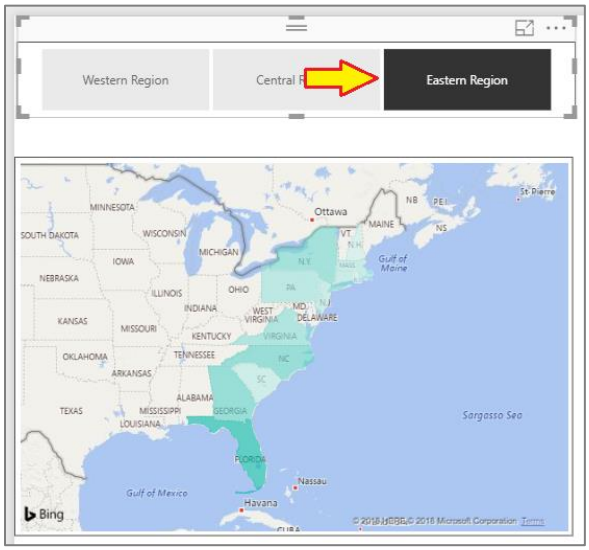
d) Using the **Fields** list, drag and drop the **Sales Revenue** column from the **Sales** table into the **Color saturation** well.

e) Reposition the new filled map visual as shown in the following screenshot.



f) Experiment clicking buttons in the slicer which will apply filtering on the Sales Region column.

g) You should see that the filled map updates whenever you select a different sales region.

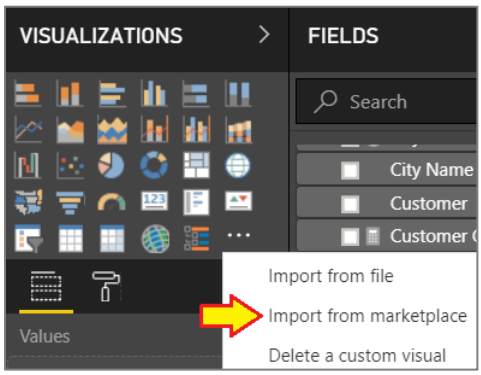


h) When you are done, click Eastern Region once more to unselect it and remove any filtering on **Sales Region**.

4. Add the PowerApps Custom Visual to your report.

a) Drop down the ellipse (**…**) menu at the bottom right corner of the **Visualizations** pane.

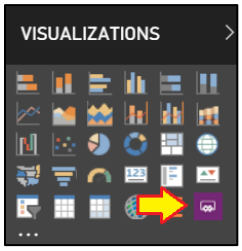
b) Select the **Import from marketplace** command.



c) When you see the Power BI Visuals dialog from the Marketplace, enter “PowerApps” into the search box.

d) When you search for “PowerApps”, locate the **PowerApps (Preview)** custom visual and click the **Add** button.

e) Once the PowerApps custom visual has been added to your report, you should see a tile for it in the **Visualizations** list.



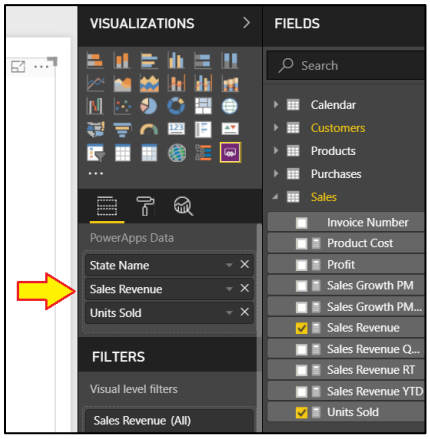
f) Click on the PowerApps custom visual tile in the **Visualizations** list to add a new PowerApps visual to the report.

g) Note when the PowerApps visual is selected, you can see a well named **PowerApps Data** in the **Fields** pane.

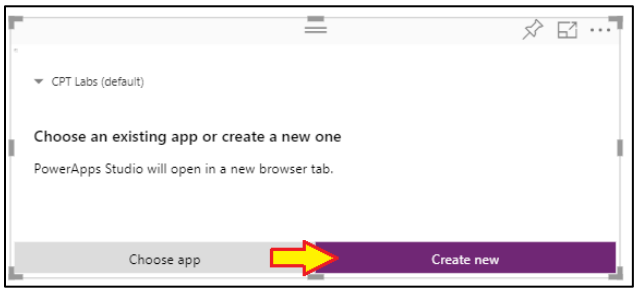
h) Using the **Fields** list, drag and drop the **State Name** column from the **Customers** table into the **PowerApps Data** well.

i) Using the **Fields** list, drag and drop the **Sales Revenue** measure from the **Sales** table into the **PowerApps Data** well.

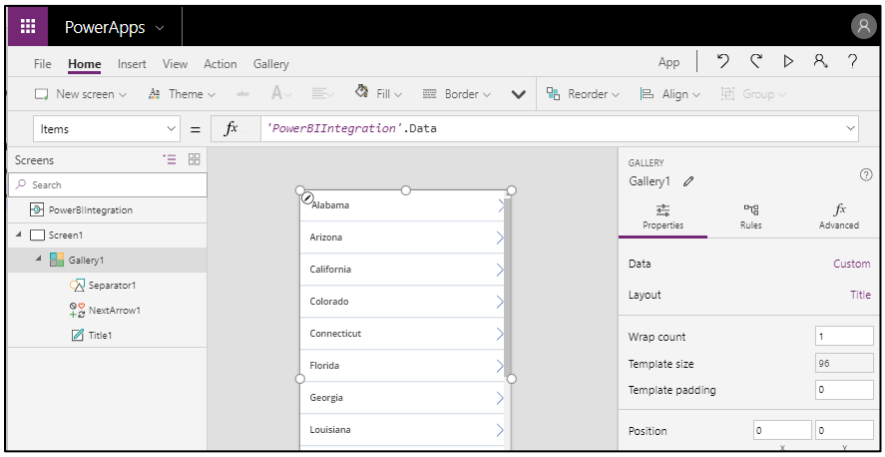
j) Using the **Fields** list, drag and drop the **Units Sold** measure from the **Sales** table into the **PowerApps Data** well.



5. Create a new app for the PowerApps custom visual.
   a) Once you have added one or more fields in the **PowerApps Data** well, the visual should display the **Create new** button.
   b) Click the **Create new** button to create a new app in PowerApps.

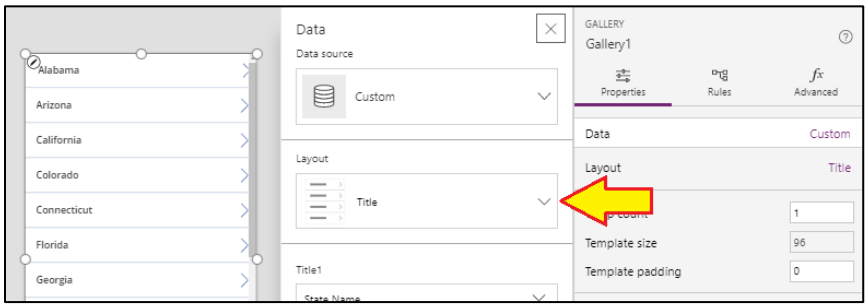


   c) You should be redirected into PowerApps Studio and you should see a new app that matches the following screenshot.



   d) Make sure Gallery1 is selected in the left navigation.
   e) In the **Properties** pane, click on the **Data** property setting which is currently set to **Custom**.

   f) When the **Data** pane appears, you should see that the **Layout** is currently set to **Title**.



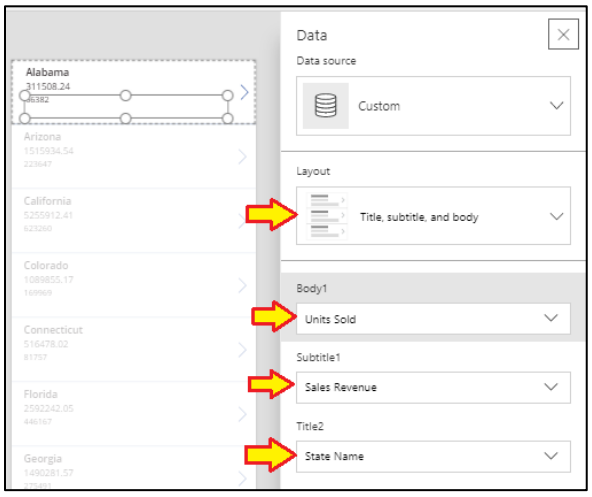
g) Set the **Layout** property to **Title, subtitle and body**.

h) Set **Body1** to **Units Sold**.

i) Set **Subtitle1** to **Sales Revenue**.

j) Set **Title2** to **State Name**.

k) Your **Data** pane should now match the following screenshot.

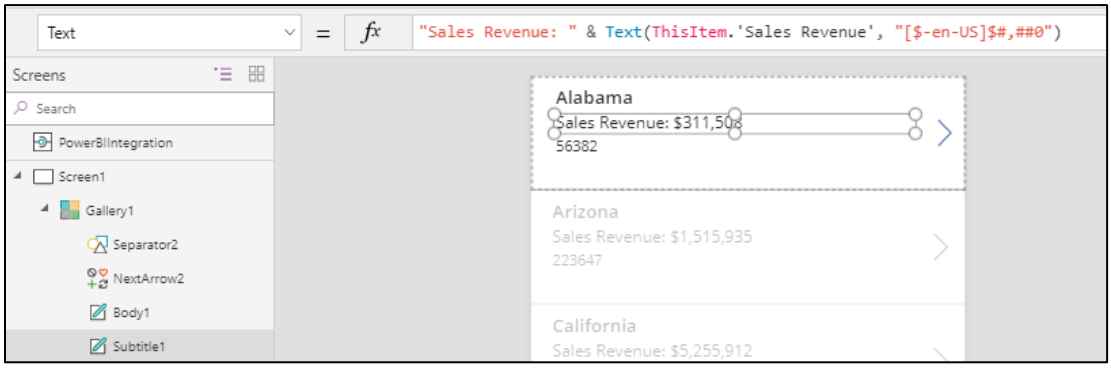


l) Select the control named **Subtitle1** in the left navigation.

m) Set the **Text** property of **Subtitle1** using the following formula.

```
"Sales Revenue: " & Text(ThisItem.'Sales Revenue', "$#,##0")
```

n) The formula bar for the **Text** property should match the following screenshot.



o) Select the control named **Body1** in the left navigation.

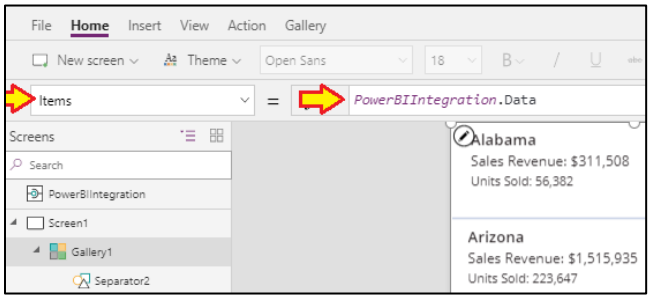
p) Set the **Text** property of **Body1** using the following formula.

```
"Units Sold: " & Text(ThisItem.'Units Sold',"[$-en-US]#,##0")
```

q) The formula bar for the **Text** property should match the following screenshot.

r) Select the control named **Gallery1** in the left navigation.

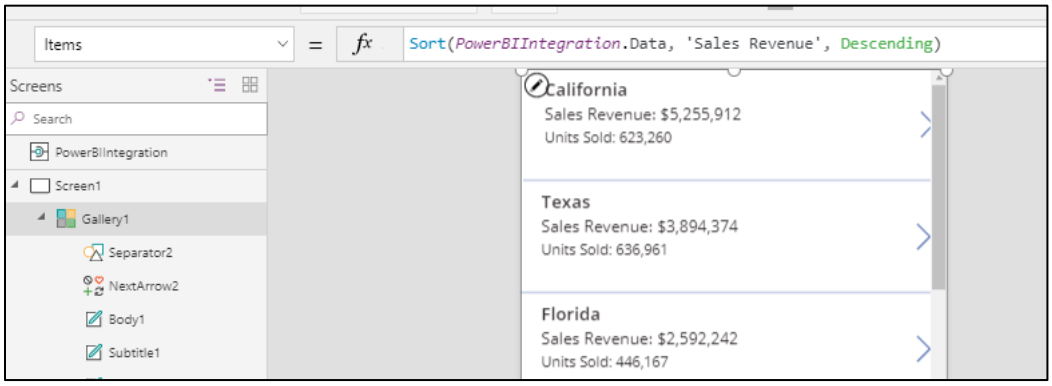
s) Inspect the **Items** property in the formula bar. You can see it is currently set to **PowerBIIntegration.Data**.



t) Update the **Items** property in the formula bar to match the following formula.

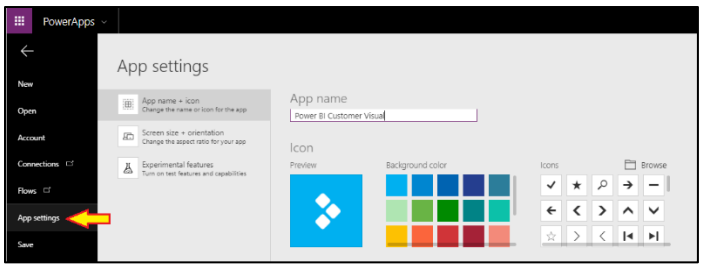
```
Sort(PowerBIIntegration.Data, 'Sales Revenue', Descending)
```

u) The formula bar for the I**tems** property should match the following screenshot and you should see that the states displayed inside the gallery are now being sorted with the states with the greatest revenue at the top.

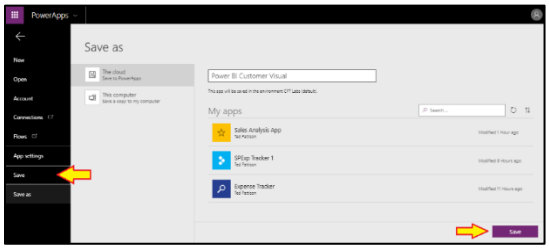


6. Save the app to the cloud.

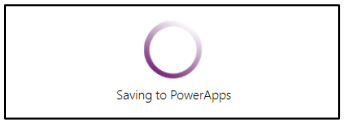
a) Navigate to the **App setting** tabs and give the new app a name of **Power BI Custom Visual**.



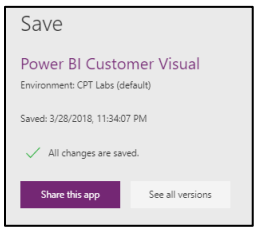
b) Navigate to the **Save** tab and save the app to the cloud by clicking the **Save** button in the lower, right corner.



c) Wait until the app has been saved.



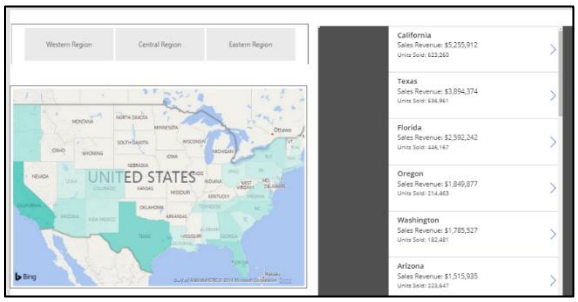
d) Wait until you see the All changes are saved confirmation.



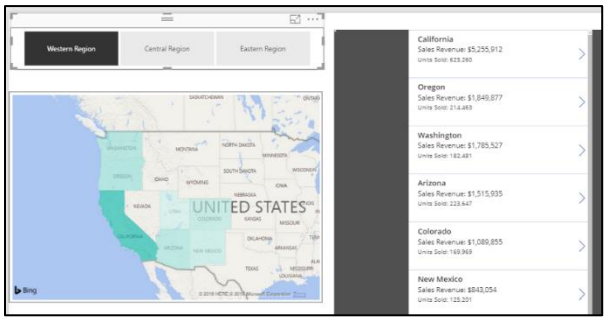
7. Return to Power BI and try out the new PowerApps custom visual.

a) Return to the Wingtip Sales Analysis report in the Power BI Service.

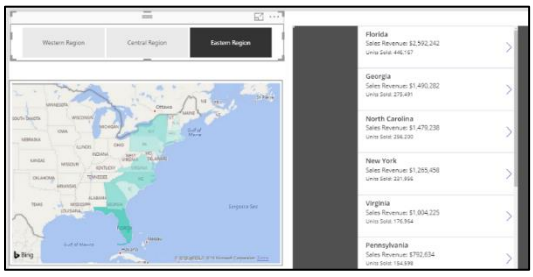
b) The PowerApps custom visual should now display alongside the other two visuals.



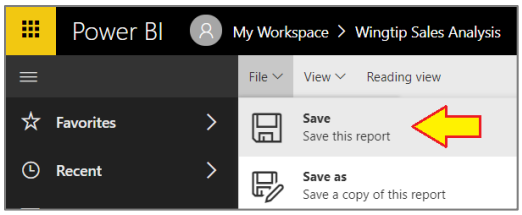
c) Click the slicer buttons for different sales regions and observe how the data is updated in the PowerApps custom visual.



d) When you click Eastern Region, the PowerApps custom visual updates to display states from the eastern region.



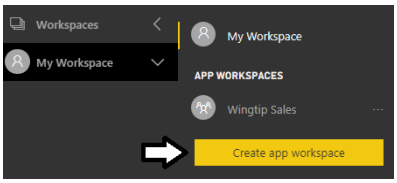
8. Save your work.
   a) Drop down the **File** menu and select **Save** to save your work.



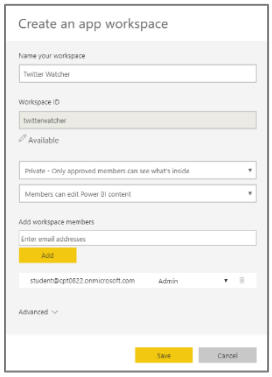
Exercise 4: Use a Flow to Create a Real-time Dashboard in Power BI

In this exercise, you will create a new streaming dataset that will be used in later exercise to build a real-time dashboard.

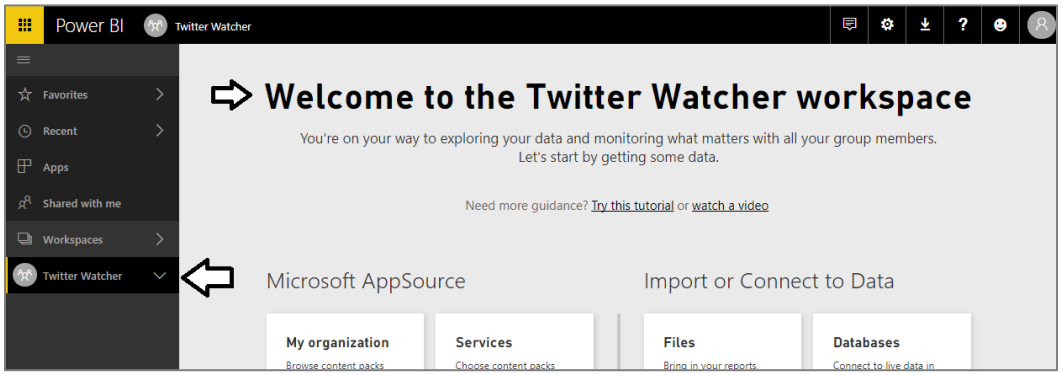
1. Create a new app workspace named **Twitter Watcher**.
   a) Log into the Power BI service in the browser.
   b) Expand the Workspaces flyout menu and click the **Create app workspace** button.



   c) Create a new app workspace named **Twitter Watcher** and make yourself an **Admin**.

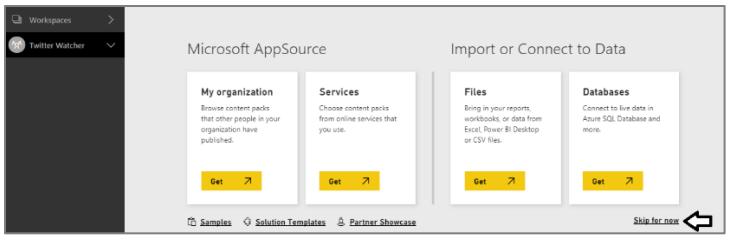


   d) You should now see the welcome page for the new app workspace.

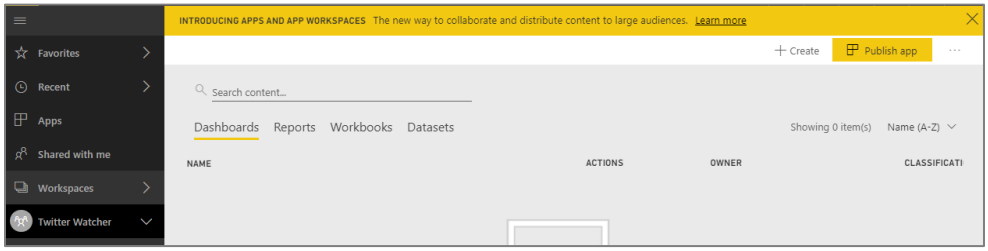


2. Create a Hybrid dataset that supports streaming.

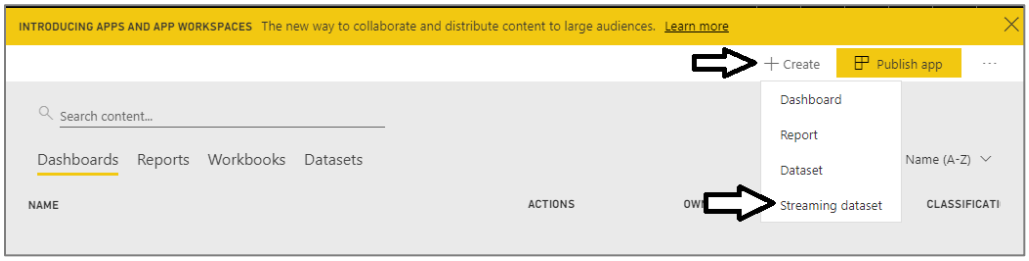
a) On the welcome page for the new app workspace, click the **Skip for now** button on the bottom right.



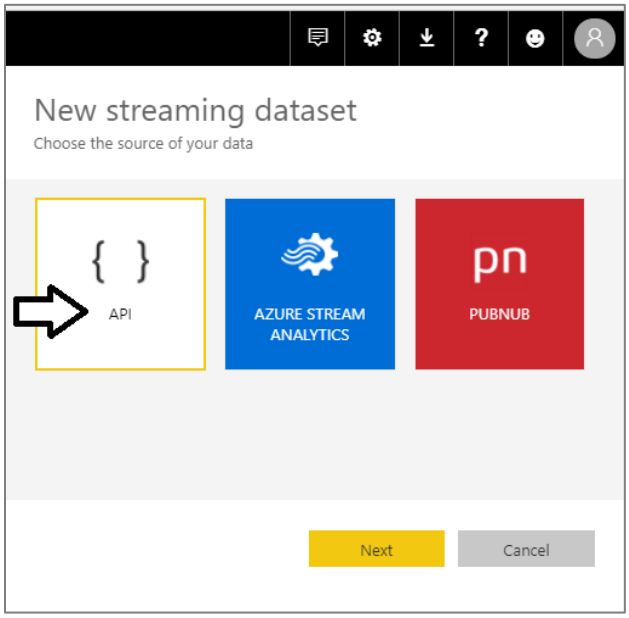
b) You should now see the summary page for the app workspace as shown in the following screenshot.



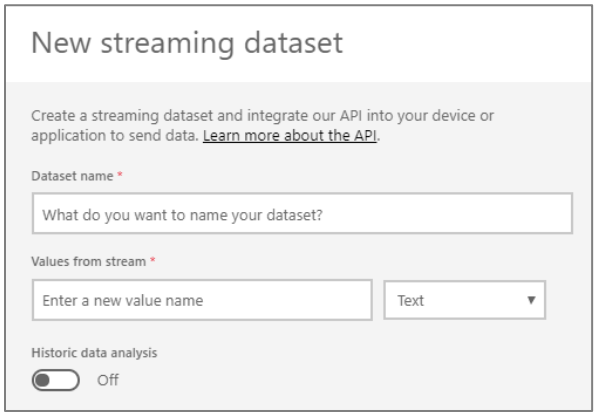
c) Click the **+ Create** dropdown menu and then click **Streaming dataset**.



d) When prompted to **Choose the source of your data**, select **API** and then click **Next**.

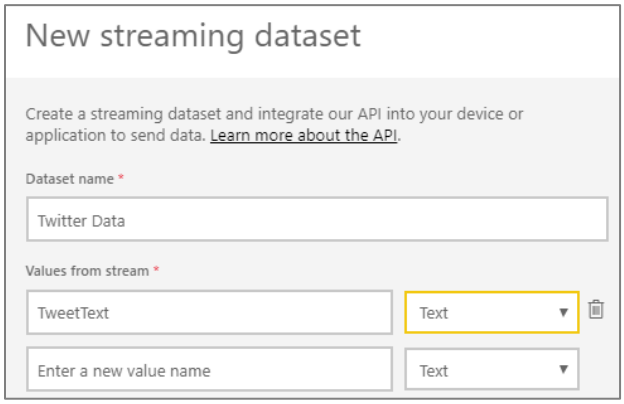


e) Next, you should be prompted with a form to fill in the Dataset name and columns for a table.



f) Enter a Dataset name of **Twitter Data**.

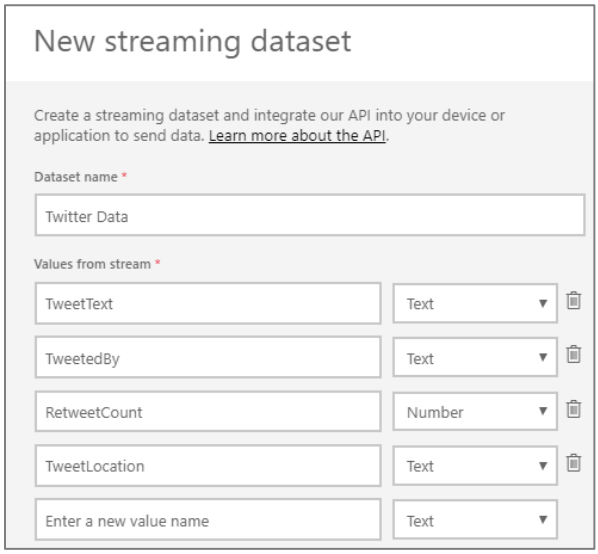
g) In the textbox under **Values from stream**, enter a column name of **TweetText**. and leave the default column type of **Text**.



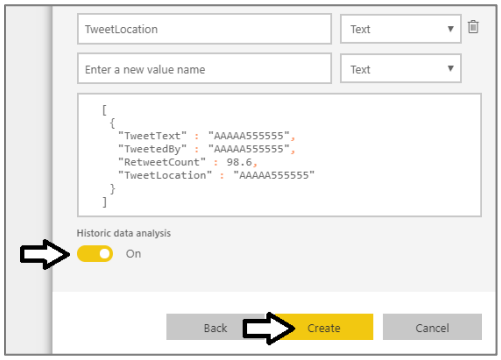
h) Add a second column named **TweetedBy** and leave the default column type of **Text**.

i) Add a third column named **RetweetCount** and change the column type of **Number**.

j) Add a fourth column named **TweetLocation** and leave the default column type of **Text**.

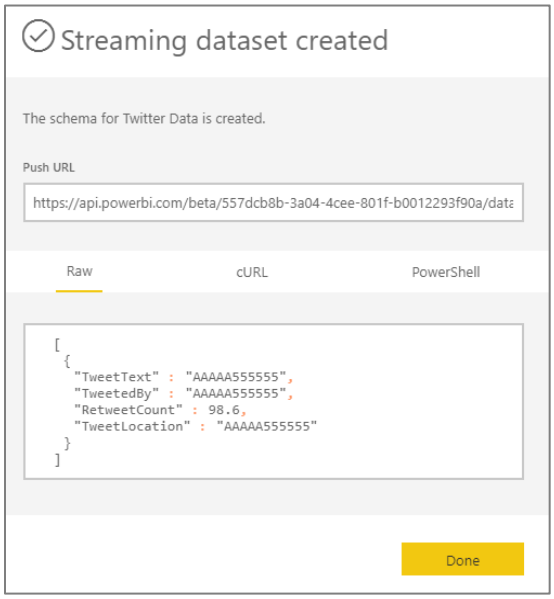


k) Set the **Historic data analysis** option to **On** and then click **Create** to create the new steaming dataset.

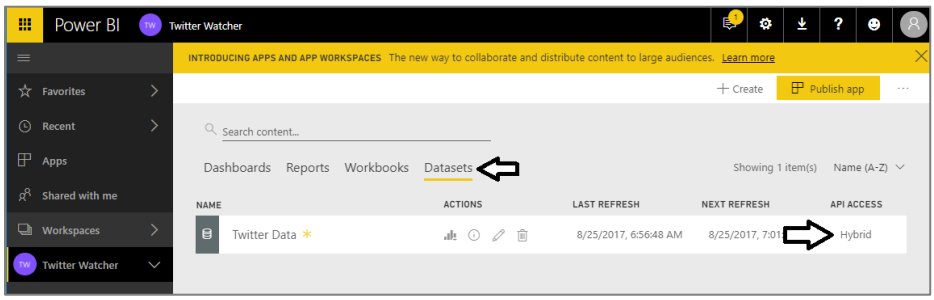


When the **Historic data analysis** option is disabled, Power BI only caches the data for the dataset in memory and the data is not guaranteed to remain in memory for more than 60 minutes. When you enabled the **Historic data analysis** option, Power BI stores the data for the dataset in an Azure SQL database. The data can then be persisted for longer periods such as days and weeks. Another important factor is that when you've enabled the **Historic data analysis** option, you can use the standard Power BI report designer to create reports on top of the datasets. Therefore, it is essential in this lab that you enabled the **Historic data analysis** option.

l) You should see a page indicating that the streaming dataset has been created. Click **Done** to dismiss this page.

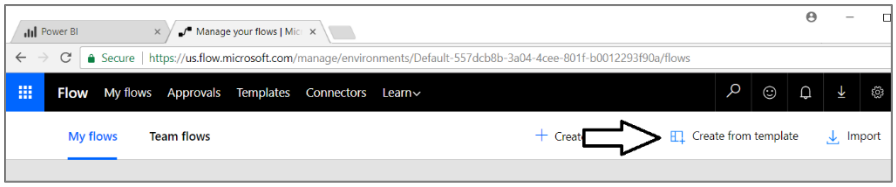


m) Navigate to the summary page for the **Twitter Watcher** app workspace and click the **Datasets** tab. You should be able to see your new streaming dataset. Note that the **API ACCESS** for your dataset is configured as a **Hybrid** dataset.



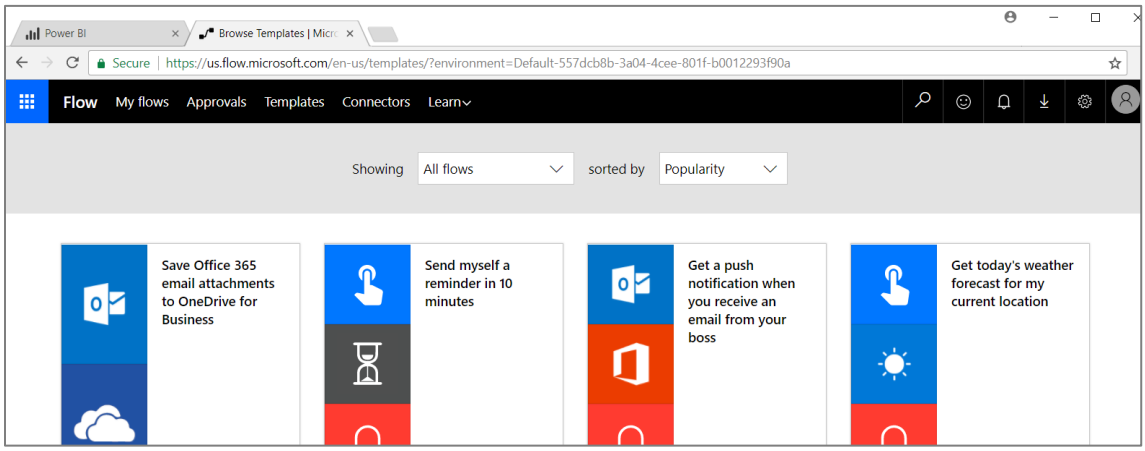
You have created a new streaming dataset named **Twitter Data**. What is not obvious is that Power BI has created a table within this dataset named **RealTimeData** which includes the columns you added in the previous step.

3. Create the Flow that triggers and collects the Twitter information.
   a) To do this navigate to http://flow.microsoft.com, and click the **Sign in** link.
   b) Click the **My flows** link to navigate to the **My flows** page.
   c) On the **My flows** page, click the **Create from template** link.

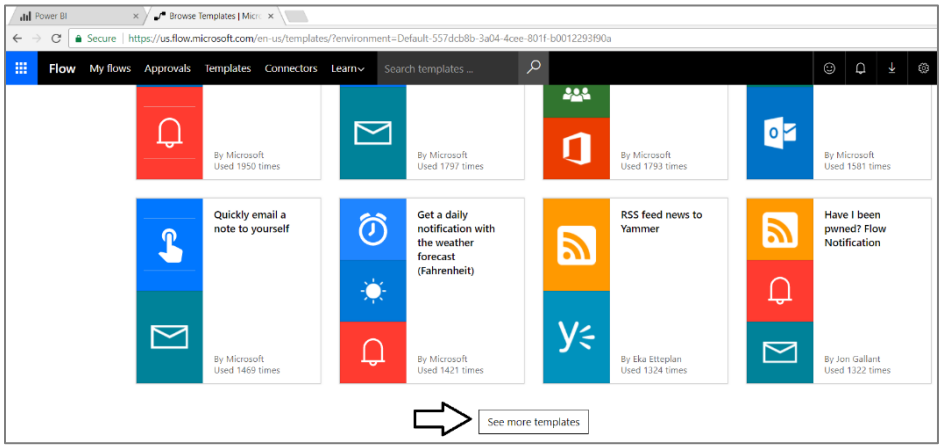


Fortunately, there are a couple of out-of-the-box templates provided by Microsoft Flow that do much of what we want such as the templates named **Save Tweets to Google Sheet**, **Save Tweets to SharePoint List** and **Save Tweets to Excel File**. You will start by using an existing flow template named **Email myself new Tweets about a certain keyword**.

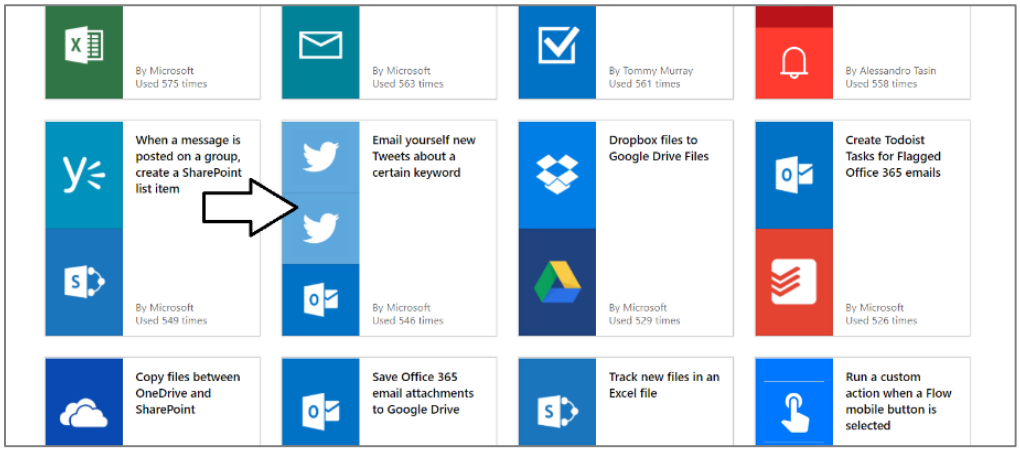
4. Select a flow template.
   a) After clicking the **Create from template** link, you are presented with a page of existing templates.



   b) Scroll to the bottom of the page and click the **See more templates** button.

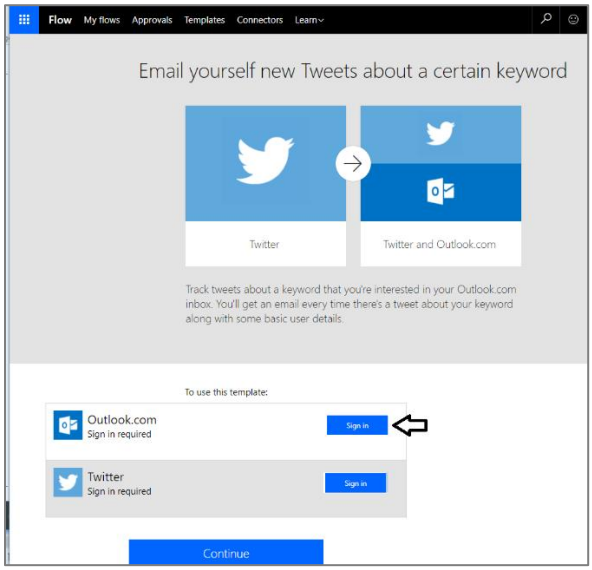


c) Select the template with the name **Email myself new Tweets about a certain keyword**.

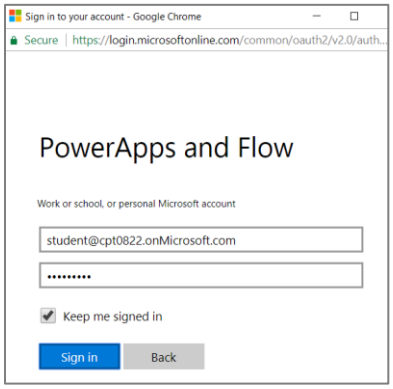


5. Log into both Outlook.com and Twitter.

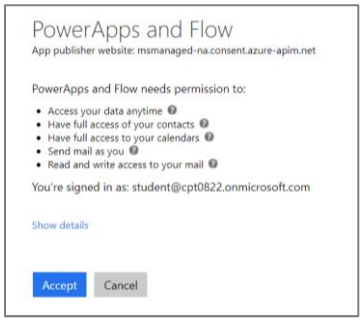
a) After selecting the template, you might be prompted to sign into Outlook and Twitter. If you are prompted to log into, click the **Sign In** button for Outlook. However, it is likely you will already be signed in from a previous lab.



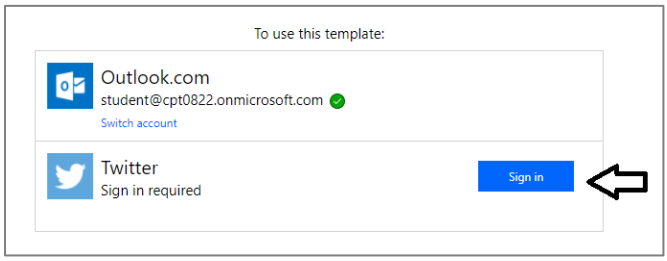
b) Sign in using your primary Office 365 account.



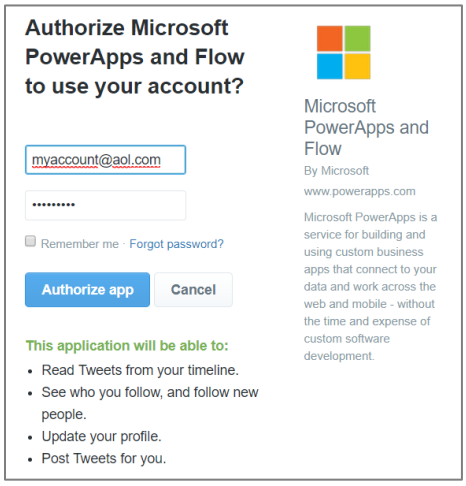
c) Click the **Accept** button to give Microsoft Flow permission to your Outlook account.



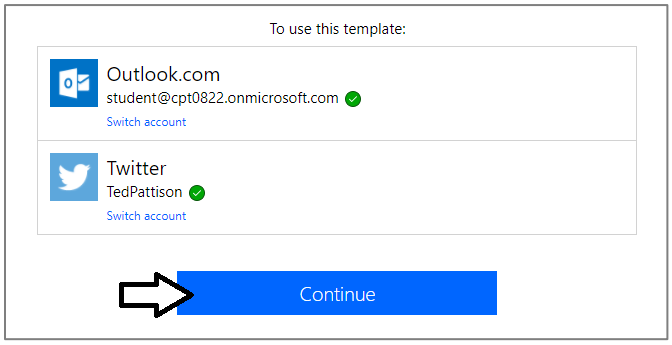
d) Click **Sign in** to sign into Twitter.



e) Sign in using your Twitter account and click the **Authorize app** button.

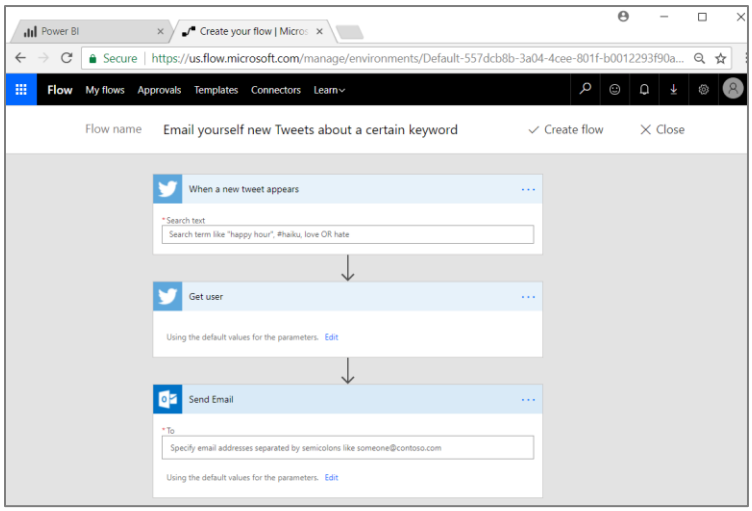


f) Once you have logged into both Outlook.com and Twitter, click **Continue**.

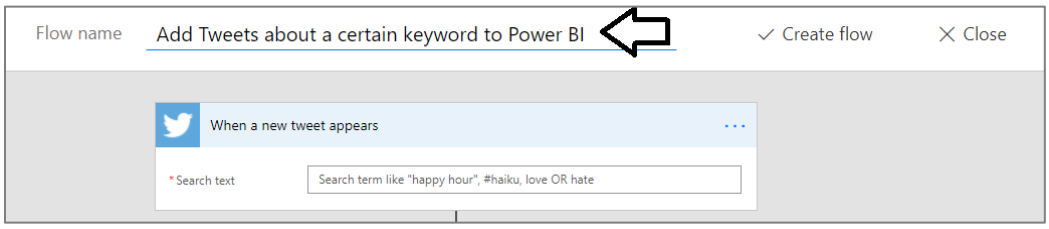


6. Design the new flow using the flow designer.

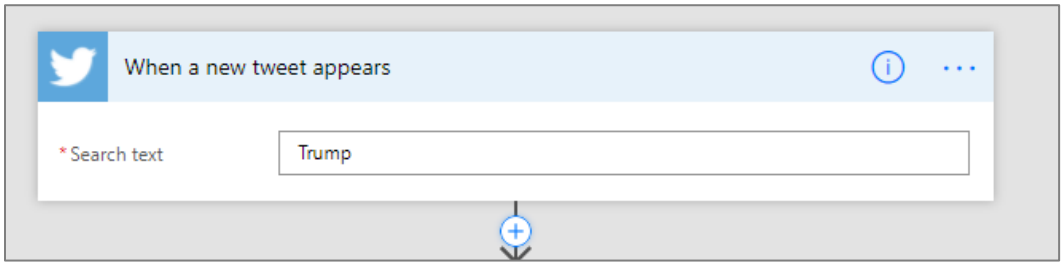
a) At this point, you should see your new flow in the flow designer.



b) Update the Flow name to **Add Tweets about a certain keyword to Power BI**.

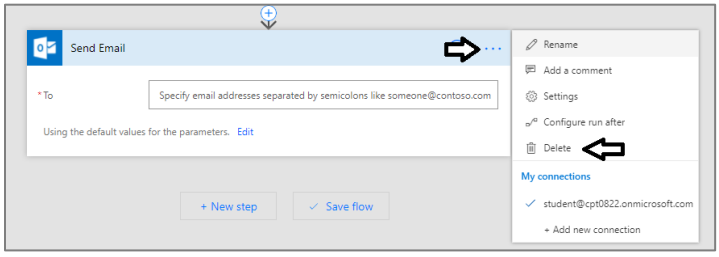


c) In the When a new tweet appears action, add a popular hashtag search term such as #PowerApps or **#PowerBI**..

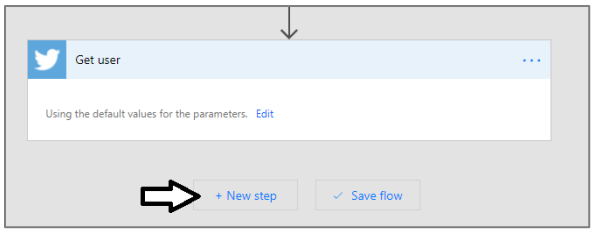


You can add any hashtag or search term you would like. Search terms that find lots of tweets are better. Feel free to use the search term like “Trump” if you want to harvest a large number of tweets in a short amount of time.

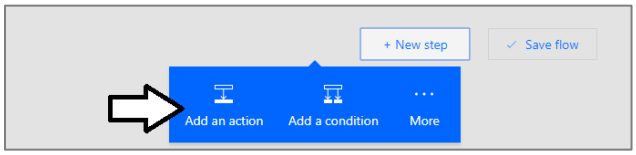
d) Delete the **Send Email** action by using the ellipse (…) menu at the top right to invoke the **Delete** command.



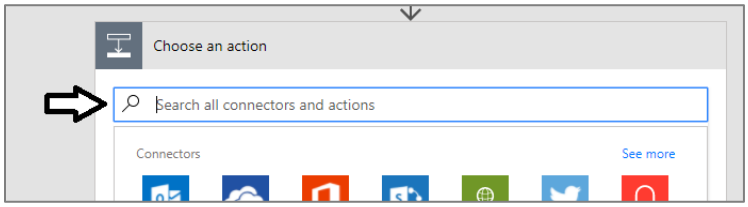
7. Modify the Flow to include the Power BI activity, that inserts the data into the Power BI data set.
   a) Underneath the Get user action, click the **New step** button.



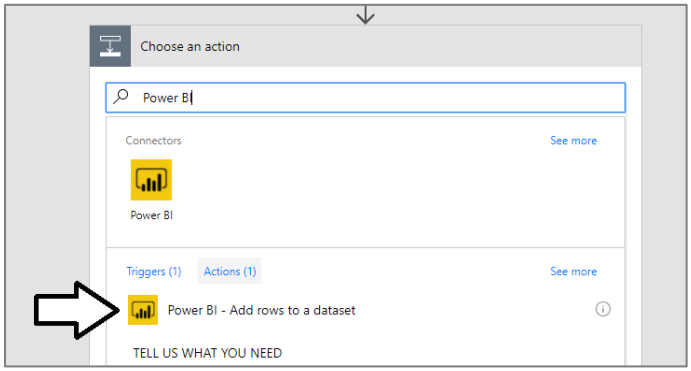
   b) Click **Add an action**.



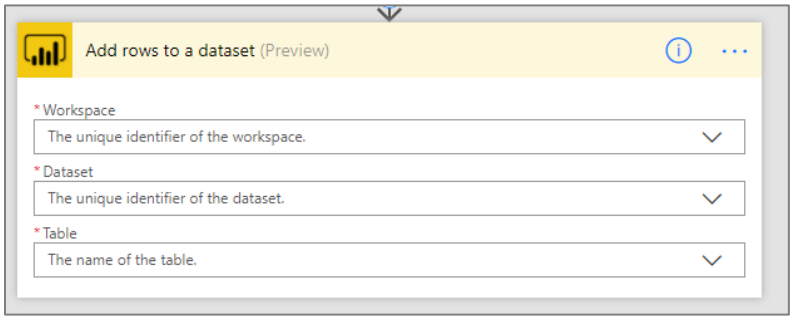
   c) You will be prompted to **Choose an action** and there is a search box to run a search of available actions.



   d) Type **Power BI** into the search box to find Power BI actions. **Select the Power BI - Add rows to a dataset** action.



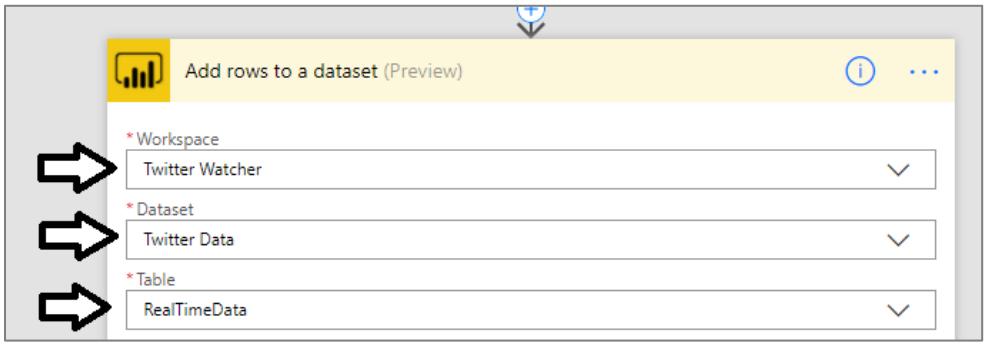
   e) You should now see the **Add rows to a dataset** action.



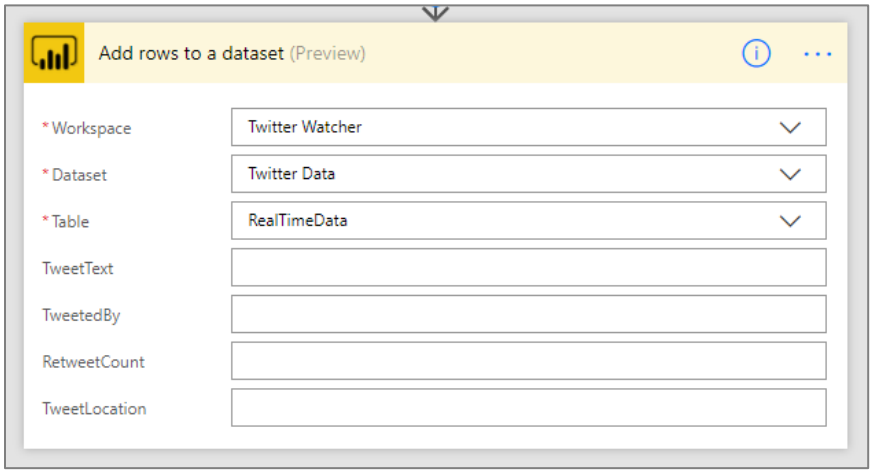
f) Set the **Workspace** to **Twitter Watcher**.

g) Set the **Dataset** to **Twitter Data**.

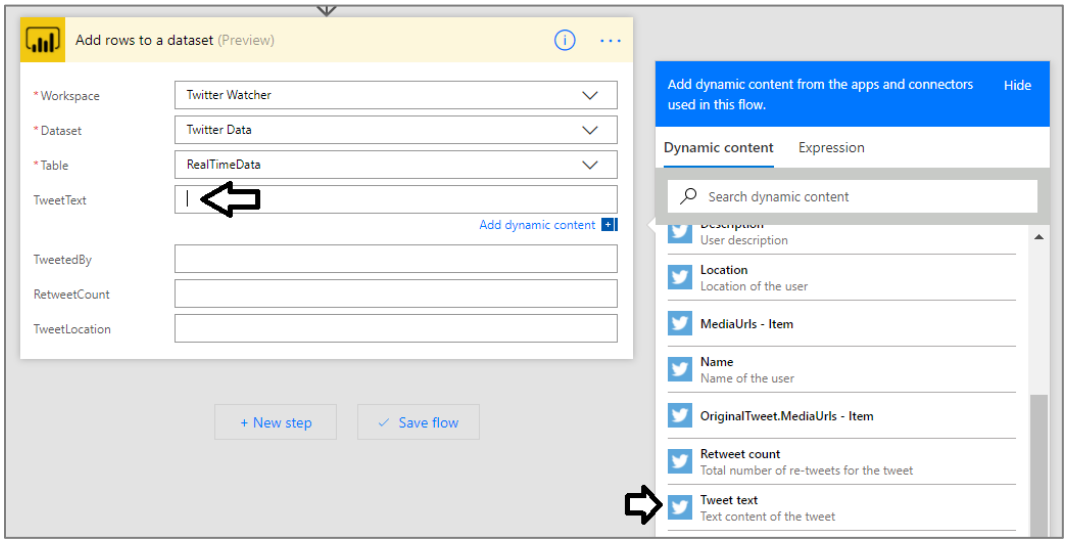
h) Set the **Table** to **RealTimeData**.



i) After you select the **RealTimeData** table, new input controls will appear for each column you added to the streaming dataset.



j) Using the mouse, place your cursor inside the textbox for **TweetText**. When you do this, you can add dynamic content by mapping fields from the data for a tweet. Select Tweet text as shown in the following screenshot.

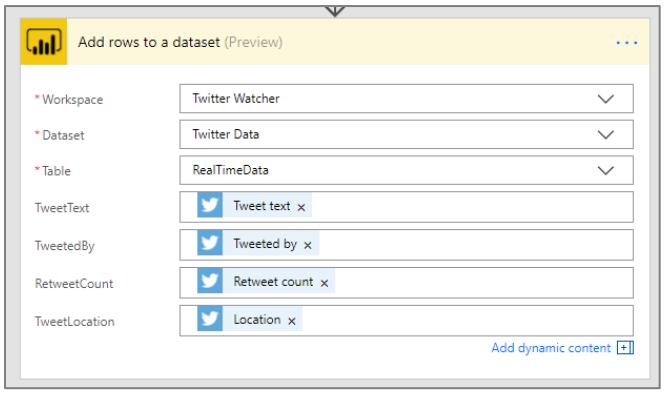


k) Map **TweetedBy** to **Tweeted by**.

l) Map **RetweetCount** to **Retweet count**.

m) Map **TweetLocation** to **Location**.

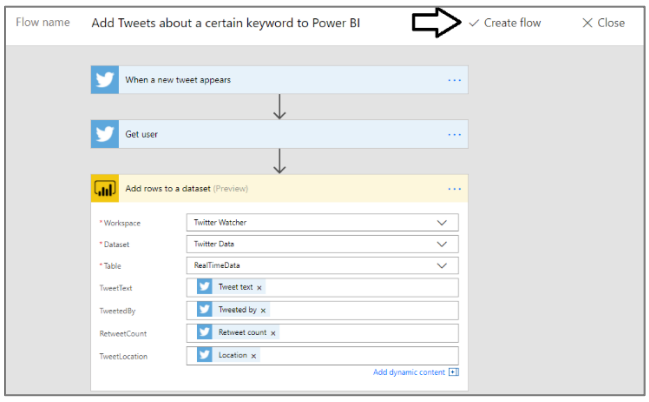
n) Your **Add rows to a dataset** action should now match the following screenshot.



You are now finished designing the flow. It is time to complete your work by creating the new flow.

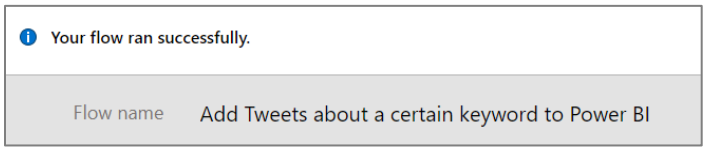
8. Create the flow

a) Click the **Create Flow** button at the top of the page to begin the process of creating the flow.



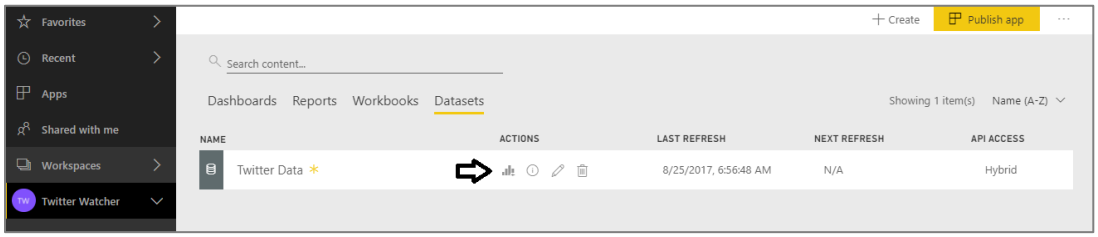
b) After a few seconds, you should see a message indicating that the flow has been created successfully.

c) After a few more seconds, the message should change indicating that the flow has run successfully.



At this point you are done creating the flow. The final step to this lab is to return to Power BI and create a report and dashboard on top of the streaming dataset which is now being populated with the flow you have just created.

9. Back to Power BI, create a report and dashboard.



a) Create something that looks like this.