

Chula Vista - Support Implementation of Open Data Program & Governance Standard

Table of Contents

Problem Statement	2
Goals	2
Overarching Goal	2
Specific Project Goal	2
Best Practices From Other Cities	2
Accomplishments	3
Final Result: Chula Vista DSD Permit Data Dashboard Web Page	3
Link to Web Page	3
Links to Individual Dashboards	3
How to Create Interactive Dashboards	4
Part 1: Identify User Demands	4
Part 2: Data Preparation	6
Part 3: Create Interactive Dashboards	10
Part 4: Publish to Web Page	17
Other Concerns	20
How often should we update the dashboards?	20
Are we relying too heavily on Power BI/ What if we stop using Power BI?	20
ADA Compliance	20
Update to Chula Vista Color Scheme	20
Appendix A - Tidy Data Problem Examples	21

Student Team

Team Name: Team Random Variable

Ester Tsai (Team Lead) | Data Science | tsaier@ucsd.edu

Ted Feng | Applied Math | zef007@ucsd.edu

Kaung Min Khant | Economics | kkhant@ucsd.edu

Problem Statement

“The City has set an objective to maximize the use of data and analytics to improve city services and increase public access to city information. While the City of Chula Vista has made strides in improving its capability for data collection and analytics, ‘silos’ still exist between various systems and there is no single repository of collected data that can be seamlessly accessed by multiple departments and the general public.”

Goals

Overarching Goal

Improve information accessibility and increase transparency for the City of Chula Vista by helping the Data Governance Team implement the Open Data Program.

Specific Project Goal

Design and publish user-friendly dashboards for the building permit data provided by Chula Vista’s Development Services Department (DSD).

Best Practices From Other Cities

The Open Data Portals of San Francisco and New York City have very similar user interfaces. They all have a page introducing the metadata of the datasets, which includes information like when the dataset was created and last updated, a description for each column, and a preview of the dataset. Furthermore, to help the public understand the data, they both provide an online visualization tool that the public can use to create quick graphs themselves.

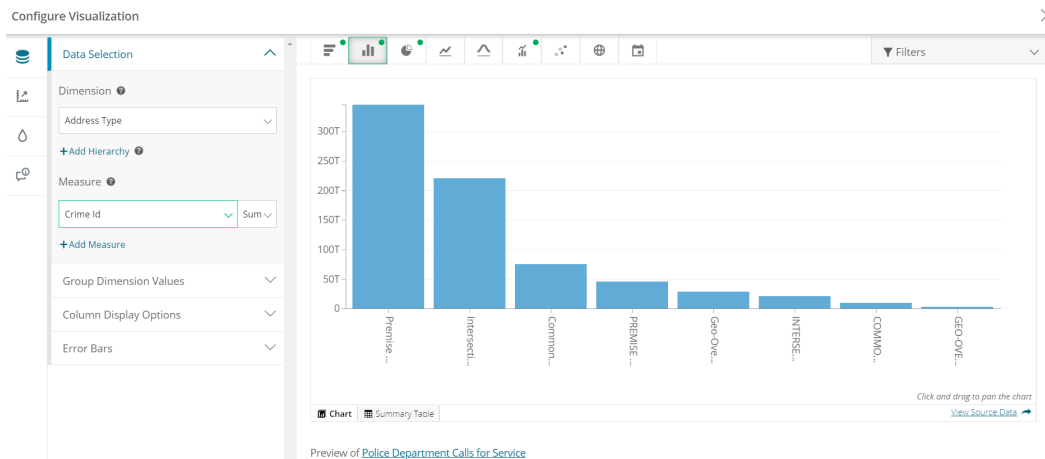


Figure 1 - San Francisco Online Data Visualization Tool

Instead of providing a visualization tool for the public, the Denver website publishes a simple table for “Average Plan Review Times” and an infographic for the department’s performance last quarter. And this inspired us to create an interactive dashboard made up of tables, cards, and bar graphs so that people who have no knowledge in data science or data visualization can gain insights from the data simply by filtering.

Average Plan Review Times

We accept emailed or online submittals for all categories listed below. [View electronic instructions here.](#)

30-Day Average:

The review times listed in the table below apply to the initial submittal only. They do not reflect resubmittals.

Plan Type	Target Initial Review Times	Current Initial Review Times (based on recent averages)
Major residential projects <i>Ex: new homes, additions 400 square feet or larger, landmark projects</i>	4 weeks	4.5 weeks
Intermediate residential projects <i>Ex: type approved submittals, additions of 400 square feet or smaller</i>	2 weeks	2 weeks
Walk-through residential projects <i>Ex: fences, interior renovations, egress windows, patios, decks, car ports, sheds, zoning-only review of solar systems</i>	2 days	5 days
Major commercial projects	4 weeks	5 weeks

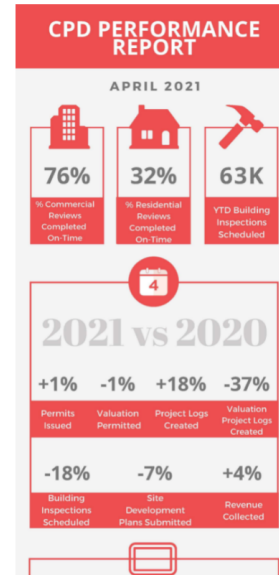


Figure 2 - Denver Permit Data Table and Infographic

Accomplishments

Final Result: Chula Vista DSD Permit Data Dashboard Web Page

Link to Web Page

<https://www.chulavistaca.gov/departments/development-services/permitdata>

This site allows visitors to find information on permit applications, permit issuances, review times, and project completions for the permits issued in the years 2018-2020.

Links to Individual Dashboards

[Estimated Review Time](#)

[New Residential Construction](#)

[Commercial Construction](#)

[Other Permit Types](#)

[Business Analysis](#)

How to Create Interactive Dashboards

For this project, we used Microsoft Power BI to create the interactive dashboards. The dashboards are interactive in the sense that users can filter data using categories such as permit type, permit subtype, zip code, and year, and they can hover over data points for the specific numbers.

This section provides standardized steps for how to create interactive dashboards, beginning with how to identify user demands and ending with publishing to the web.

Part 1: Identify User Demands

Identifying user demands is the first step in creating dashboards that deliver what the general public are most interested in.

Identify User Demands Overview:

1. Identify User Questions
2. Categorize the Questions
3. Identify Needed Data Fields

Step 1: Identify User Questions

Before we start building anything for the dashboards, pause and think—What are people interested in knowing? What information are people looking for? The final dashboards should be able to answer the questions people are most interested in. There are several keywords you can search up to identify user questions, replacing the word “permit” with your subject of interest: “Permit Data Analysis,” “[insert a major city name] Permit Data Report,” “Benefits of Permit Data,” and “What Can I Know with Permit Data.”

Some example questions that we can answer with data are:

What is the expected application review time and application fee (for my permit type)?
Do you want to see who the major players are in the roofing industry in your area?

Step 2: Categorize the Questions

After identifying what the public wants to know about permit data, we categorized the questions so it is easy to create separate dashboards later. For example, we categorized building permit data questions into the following:

- Permit Application Questions
 - Expected plan review time
 - Cost of different types of permit
- Business/ Land Developer Questions
 - Most popular service providers for a specific service type based on region
- Department Performance Questions
 - Number of different types of permits issued

Step 3: Identify Needed Data Fields

After categorizing questions, we identified what specific data fields were needed to answer each question. A data field is a column in a table, which contains data for a single variable, e.g. ‘Permit Type’.

Here is an example of data-answerable questions with their corresponding data fields:

Permit Application Questions	
Question	Data Fields
Expected plan review time	Permit Type, Permit Subtype, Date Applied, Date Issued
Application fee for different permit types	Permit Type, Permit Subtype, Total Fee

Business/ Land Developer Questions	
Question	Data Fields
Most popular service providers based on location	Service Provider, Zip Code
Most popular service providers for each permit subtype	Permit Type, Permit Subtype, Service Provider

Department Performance Questions	
Question	Data Fields
Number of different types of permits issued	Permit Type, Permit Subtype, Date Applied, Date Issued
How much fees have been collected for different types of permits, or for different years/months	Permit Type, Permit Subtype, Date Issued, Total Fee

After identifying the needed data fields, we can move on to getting the dataset and preparing it for graphing.

Part 2: Data Preparation

Not all datasets can be used for graphing. The goal of data preparation is to fix different kinds of problems in a potentially messy dataset and prepare the dataset for easy graphing.

Data Preparation Overview:

1. Get dataset
2. Make sure every data table is in Tidy Data format
3. Remove unnecessary columns and columns with sensitive information
4. Identify and correct any value error, then standardize value names
5. Add useful columns
6. Rename columns

Step 1: Get Dataset

NOTE: The simplest form of a dataset is a singular table consisting of columns and rows. There can also be several datasets, each one having only one table. When a dataset has more than one table, all the tables must share a primary key, which is a column of unique IDs that all tables have, so it is possible to combine, or join, the separate tables.

After identifying the data fields needed, export the data from the source to an Excel file. Power BI also accepts CSV, XML, JSON, and other types of data files, but Excel is the most convenient because we can easily perform data cleaning in Excel.

Step 2: Make Sure Every Data Table is in Tidy Data Format

The data being exported should be in [Tidy Data format](#), defined as follows:

1. Each variable is a column
2. Each observation is a row
3. Each type of observational unit is a table

Common dataset problems include:

- Column headers are not on row 1
- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

For examples of these Tidy Data problems, see Appendix A.

For details on how to correct these Tidy Data problems, please reference [this document](#).

Step 3: Remove Unnecessary Columns & Columns with Sensitive Info

To [delete a column in Excel](#), right click on the column's letter header and select "Delete". Remove columns that are not in your list of data fields needed, but be sure to keep the unique ID column if there is one, so it is possible to identify a row by its unique ID.

A published dataset should not contain any private or sensitive data that trace to a specific person. Remove any column with private or protected data. Every city may have variations in what is defined as protected data, but protected data is generally defined as:

- Any data which if published would violate state or federal laws governing privacy or privileged information
- Any data which if published would violate contractual or proprietary rights
- Any data which if published would jeopardize the security of individuals or property

Step 4: Identify and Correct Any Value Error, Then Standardize Value Names

Each cell in a table (that is not a column name) contains a value. If we incorporate a wrong value into the dashboard, the dashboard would display inaccurate information. To prevent that, we must perform a thorough "sanity check" on each column. A sanity check checks for anything that is obviously wrong.

How to do a sanity check:

1. For all the columns with **text data**, look at how many categories there are, and how many values are there for each category. Note, after removing unnecessary columns, which often include removing "Description" columns, all the remaining columns should be categories such as "Permit Type" or "Zip Code." Check if all the category names make sense, and if there are any categories with extremely few values.
2. For all the columns with **numerical data** and **datetime data**, plot the data distribution and look for anything unexpected, such as unreasonable outliers or a strange distribution shape.

Common errors to catch with sanity checks:

- Unreasonable values and outliers
 - For example, a dataset for permits applications in the years 2015-2020 should not contain permits from 2006. Similarly, the application date for a permit should never be later than its issuance date, since applications always come before issuances.
- Typos
 - For example, the value contains an extra zero or a missing decimal. These errors can often be caught looking at data distributions.

- Inconsistent value names
 - Some of the values mean the same thing but are counted as different values due to small naming differences.
 - Here is an example: The value names in the column RECORD STATUS need to be standardized.

RECORD STATUS	
Applied	569
C of O Issued	3
Closed	27543
Corrections Letter Sent	349
Corrections Required	2
Expired	2762
Expired - Closed	3
Final Inspection Complete	15
In Process for Issuance	20
In Review	163
In-Review	2
Incomplete Submittal	70
Issued	4436
Ready To Issue	58
Ready to Issue	155
Streamlined Submittal	4
Submitted	1
Withdrawn	391

Figure 3 - Inconsistent Value Names in Original Permit Dataset

This can be fixed by replacing all variations of a value name to the same name (e.g. “Ready To Issue” changed to “Ready to Issue”). Before moving on, make sure the value names are standardized across all your datasets, not just this one table.

- Missing values
 - If a column contains too many (> 50%) missing values or null values, you might consider removing the entire column.
- Duplicate values
 - Every observation (e.g. a particular permit application) should have a unique ID, and should only take one row. When there are multiple rows with the same ID, remove the extra rows and only keep one row. It is important to find out what caused the duplicate rows; for example, the duplicate rows in the initial building permit dataset were caused by multiple submissions and review cycles for the same permit. In that case, it is best to keep only the data for each permit application's first review cycle.

Note that errors can still remain after rounds of sanity checks. Taking great care to understand what each variable means and what is expected of each variable can help detect abnormalities. For more details on sanity checks please check out [this article](#).

Step 5: Add Useful Columns

Sometimes we need to do some calculations or text slicing on existing variables to derive the exact information we want. For example:

Existing Variable(s)	Method	New Variable(s)
Full Address 617 Davenport Ln, Chula Vista, CA 91911	Extract substring in Excel	Zip Code 91911
Date Applied 5/1/2020 Date Issued 5/31/2020	[Date Issued] - [Date Applied] Column calculation in Excel	Plan Review Time (Days) 30

NOTE: text slicing can cause errors, so be sure to catch anticipated errors with IF, IFERROR, or ISERROR. For example, what if a Full Address value does not have a zip code? Take a look at this example Excel formula:

```
=IF(ISERROR(FIND("9191", RIGHT(O2, 5))), "", RIGHT(O2, 5))
```

Part of Formula	Function Syntax	Meaning
FIND("9191", RIGHT(O2, 5))	RIGHT (text, [num_chars]) FIND (find_text, within_text, [start_num])	Find the position of the target text “9191” within the rightmost 5 characters of the cell O2. If the target text does not exist, an error occurs.
ISERROR(FIND(...))	ISERROR (value)	If the position of the target text “9191” produces an error, meaning it does not exist, then this ISERROR function returns True.
IF(ISERROR(...), "", RIGHT(O2, 5))	IF (logical_test, [value_if_true], [value_if_false])	If the ISERROR function returns True, fill the cell with “” (empty string), else fill the cell with the Zip Code, which is the rightmost 5 characters of the Full Address

Step 6: Rename Columns

To rename a column, simply edit the cell in row 1 with the column name. It is important to give user-friendly names to all the variables that will appear on the dashboard. For example, “RECORD TYPE” should be renamed to “Permit Type” for ease of understanding. Be sure to standardize the formatting too, since databases can have inconsistent variable naming conventions, such as “ADDR FULL LINE” for full address, “Lic Prof” for licensed professionals, and “Sub Type” for permit subtype.

Part 3: Create Interactive Dashboards

Power BI (Power Business Intelligence) is a free data visualization tool by Microsoft.

Step 1: Set Up Power BI

Once open Power BI, we can import the data we collected and cleaned from “Get data”.

There are 3 tabs on the left side, the first one is ‘Report’, which is the place to design and create a dashboard:

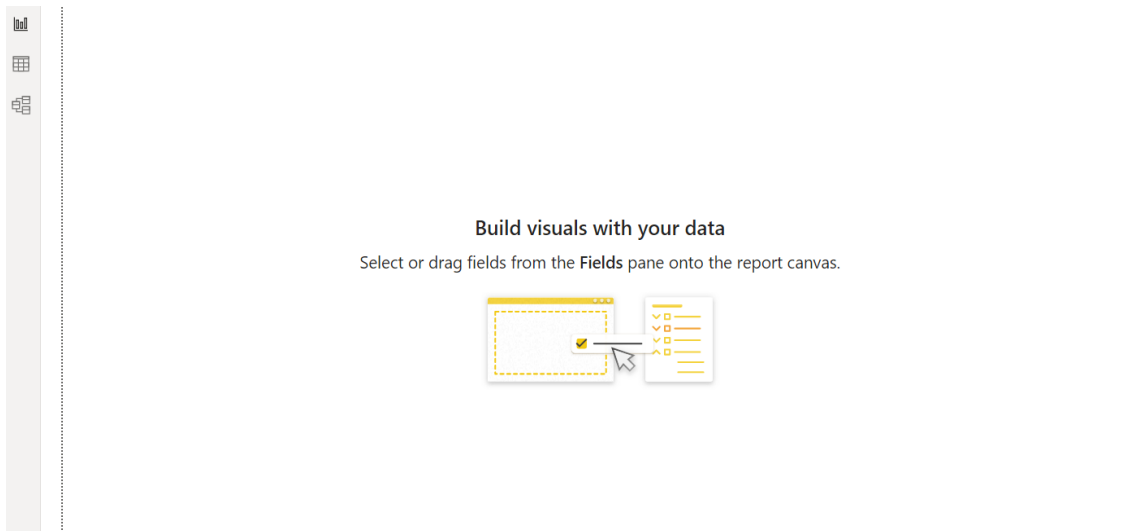


Figure 4 - Power BI Blank Report Tab

The second tab is ‘Data’, where we can check and manipulate the data we imported. We can change column names, change a column’s data type, and add new columns that contain useful information:

Column1	RECORD ID	RECORD TYPE	RECORD TYPE SUBTYPE	Sub Type	Class Desc	DESCRIPTION	RECORD STATUS	DATE STATUS	Date Applied	Date A
33461	W17-0669	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/16/2017 12:00:00 AM	9/19/2017 12:00:00 AM	
33469	W17-0677	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	3/15/2018 12:00:00 AM	9/21/2017 12:00:00 AM	
33470	W17-0678	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	3/1/2018 12:00:00 AM	9/22/2017 12:00:00 AM	
33472	W17-0680	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/3/2017 12:00:00 AM	9/22/2017 12:00:00 AM	
33475	W17-0683	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/10/2017 12:00:00 AM	9/25/2017 12:00:00 AM	
33483	W17-0691	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/13/2017 12:00:00 AM	9/26/2017 12:00:00 AM	
33516	W17-0724	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	5/24/2018 12:00:00 AM	10/4/2017 12:00:00 AM	
33520	W17-0728	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/16/2017 12:00:00 AM	10/6/2017 12:00:00 AM	
33533	W17-0741	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/26/2017 12:00:00 AM	10/12/2017 12:00:00 AM	
33553	W17-0761	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/9/2017 12:00:00 AM	10/17/2017 12:00:00 AM	
33559	W17-0767	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/2/2017 12:00:00 AM	10/17/2017 12:00:00 AM	
33565	W17-0773	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/6/2017 12:00:00 AM	10/19/2017 12:00:00 AM	
33567	W17-0775	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	12/18/2017 12:00:00 AM	10/19/2017 12:00:00 AM	
33568	W17-0776	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/30/2017 12:00:00 AM	10/19/2017 12:00:00 AM	
33575	W17-0783	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	12/5/2017 12:00:00 AM	10/20/2017 12:00:00 AM	
33577	W17-0785	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/7/2017 12:00:00 AM	10/20/2017 12:00:00 AM	
33578	W17-0786	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	10/31/2017 12:00:00 AM	10/23/2017 12:00:00 AM	
33579	W17-0787	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/21/2017 12:00:00 AM	10/23/2017 12:00:00 AM	
33580	W17-0788	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/9/2017 12:00:00 AM	10/23/2017 12:00:00 AM	
33683	W17-0790	Residential Utility Permit	Minor Utility	Plumbing	329 - Private Site Improvements	Replace 50 gal gas water heater like for like	Closed	11/13/2017 12:00:00 AM	10/23/2017 12:00:00 AM	

Figure 5 - Power BI Data (Power Query) Tab

The third tab is ‘Model’, where we can relate different fields across different datasets. The ‘Model’ tab is not utilized since we are only working with one dataset.

Step 2: Format the Dashboards

There are 2 things we need to keep in mind when formatting the dashboards:

First, we need to make sure our dashboard fits on the website. Since the users are able to scroll vertically on the website, our dashboard needs to have a slim shape or smaller width and larger height to fit the shape of the website.

Second, we want to present the data in a way that is easy for the public to understand. Interactive dashboards allow users to filter the data and get the information they need based on category and time. In each dashboard, there should be a ‘Slicer’ for each of the main categories, as well as one for time. Users are able to filter the data based on their interests, and the data displayed in the dashboard will change accordingly.

For example, in the New Residential Construction Dashboard, we have three filters at the top: Permit Subtype, Zip Code, and Year Permit was Issued.

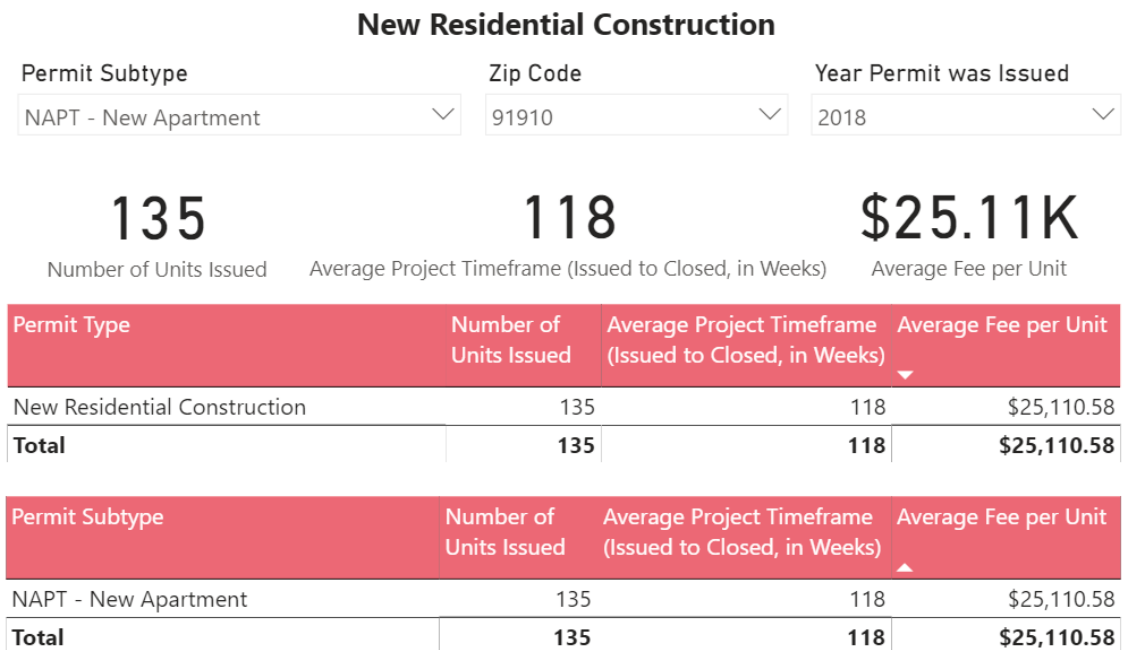


Figure 6 - Example Filtered Dashboard

When the user selects ‘NAPT - New Apartment’ for Permit Subtype , ‘91910’ for Zip Code, and ‘2018’ for Year Permit was Issued, the information in the dashboard will only include the data that satisfy the conditions of all three filters.

Step 3: Create the Dashboards

Power BI provides a wide variety of graphing tools in the Visualizations section. And for this project, we will be using the Card, Table, Bar Chart and Slicer.

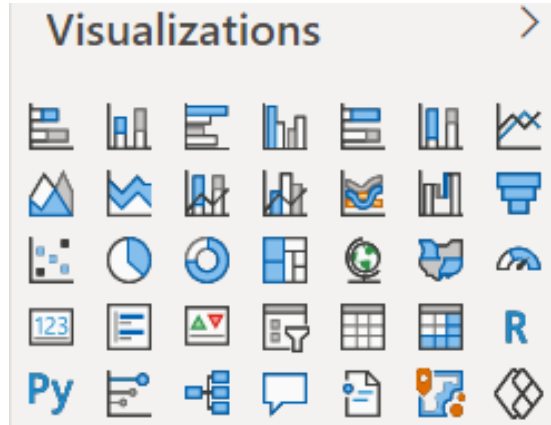


Figure 7 - Power BI Visualizations Sections

Here is an example Power BI dashboard showcasing the various ways to display information. (NOTE: the numbers shown in Figure 8 are not fully accurate because this old version of the building permit dataset contains more than one review cycle for each record. This issue has been fixed in later dashboards.)

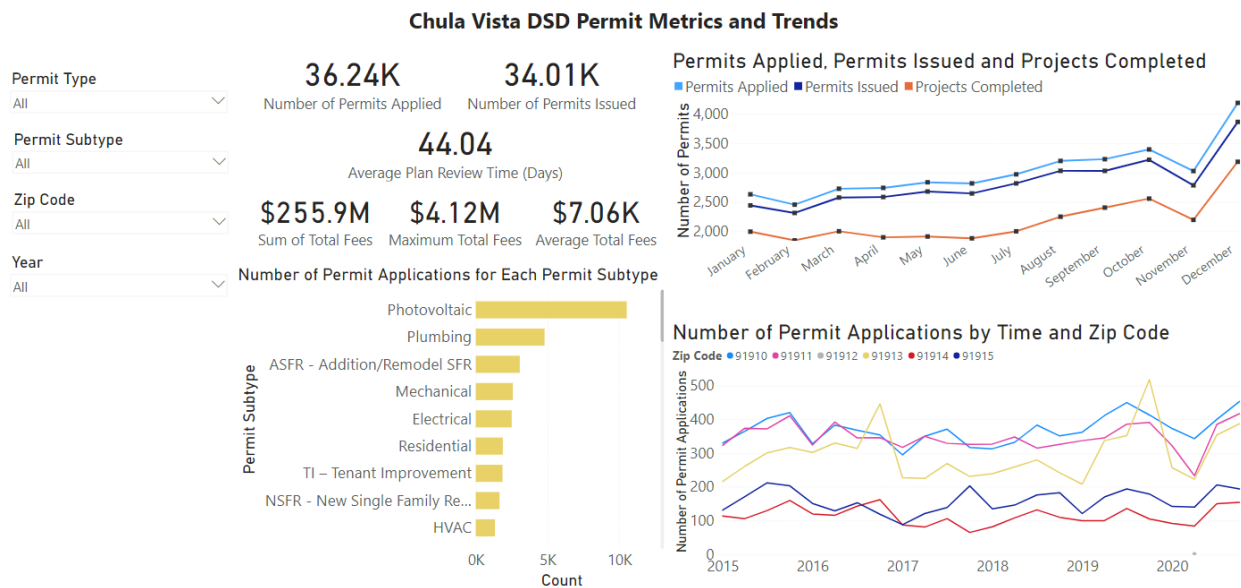


Figure 8 - Example Power BI Dashboard

Cards:

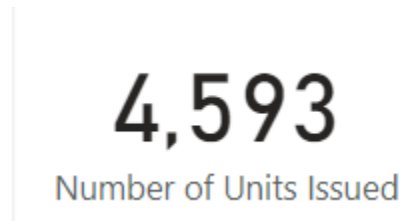


Figure 9 - Power BI Card

The most simple and the most straightforward way to present data is to display a single number or value. For questions like “How many units are issued” or “What is the average cost of the permit,” the answer is a single value. In Power BI, we can use the ‘Card’ function in the Visualization to display a single value. Card is the 1st element of the 4th row in the Visualizations section (Figure 7).

And then in the Fields section, select the data fields (i.e. variables in the dataset, as denoted by column headers) we want to display. For example, in order to display the total number of units issued, we need to insert the Housing Units to the Fields sections. Then, in the drop down menu, select the statistics of the data we want to present, which is Sum in this case. Furthermore, we can rename the data fields to be more user-friendly using the “Rename for this visual” function in the drop down menu.

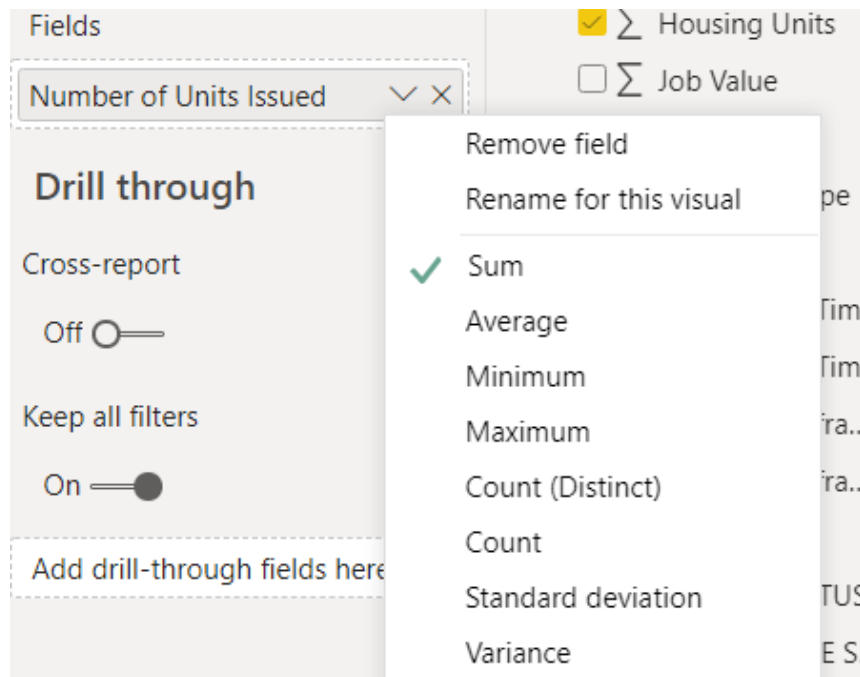


Figure 10 - Changing a Field's Statistics Type

Tables:

If we want to display specific number values for more than one data field, we can use a table to present the information. In Power BI, we can insert a table by selecting the 5th element on the 4th row in the Visualizations section (Figure 7).

Permit Subtype	Number of Units Issued	Average Project Timeframe (Issued to Closed, in Weeks)	Average Fee per Unit
NAPT - New Apartment	2,445	77	\$26,028.28
NASD - New Accessory 2nd Dwelling Unit	123	36	\$7,883.94
NCDO - New Condo	991	49	\$29,412.11
NDCO - New Detached Condo	90	51	\$50,264.93
NDUP - New Duplex	56	42	\$40,808.70
NMFG - New Manufactured Home	2	70	\$20,809.07
NSFR - New Single Family Residence	885	48	\$48,126.95
Private Site Improvements	0	57	
Total	4,592	49	\$40,572.05

Figure 11 - Power BI Table

Before inserting any data into the empty table, we need to make sure our data is clean, especially the units. Project Timeframe is in weeks, fee is in dollars. We are able to change the format or units of the data in the Data section as mentioned in Part 2.

After inserting an empty table, we can select the values/fields we want to display in the table. And each selected field will be a column in the table.

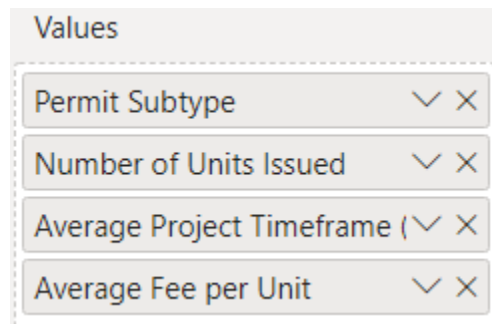


Figure 12 - Power BI Table Values

Notice how order matters—the data field on the top (Permit Subtype) will appear as the leftmost column next in the table, and the bottom field will be the rightmost column. In the drop down menu, we can select the different statistics (e.g. Sum, Average, Medium) of the data that we want to present.

Bar Charts:

If we want to compare different categories, we can use a bar chart. Bar charts can be vertical or horizontal. We used a horizontal Bar Chart to better display category names. Horizontal bar chart is the 3rd element of the 1st row in the Visualizations section (Figure 7).

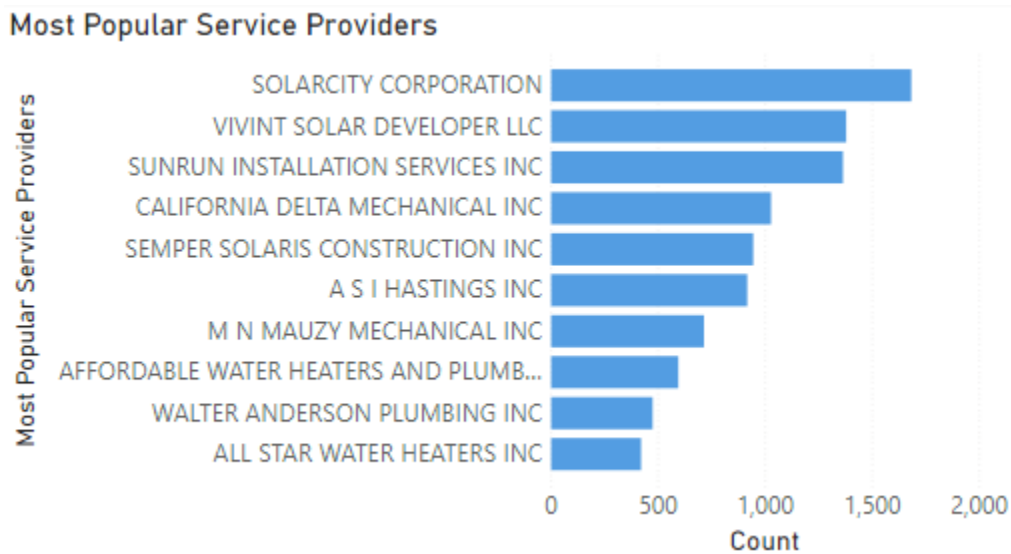


Figure 13 - Power BI Horizontal Bar Chart

After inserting an empty bar chart, we put the categorical variable (Service Providers in this case) in the Axis section, and we renamed it to “Most Popular Service Providers” to match the context of this bar chart. Next, we put the data field Permit ID (though technically any field works) under the Values section and select Count in the drop down menu, so that the bar chart counts the number of permit applications for each Service Provider.

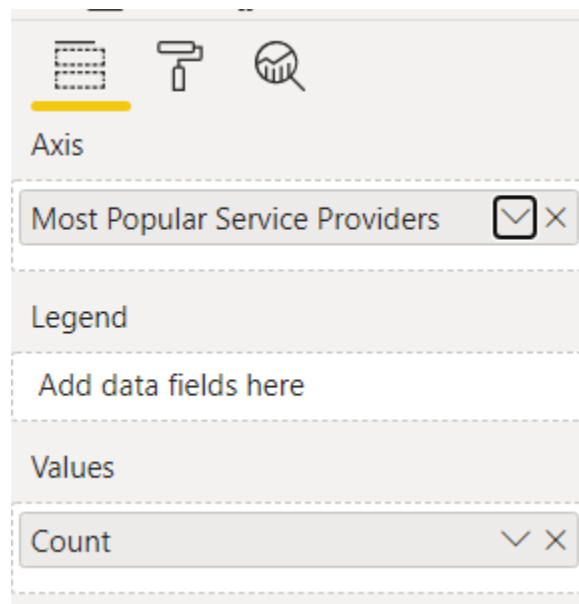


Figure 14 - Power BI Bar Chart Axis and Values

Slicers:

A 'Slicer' is a checkbox/ filter function that we can implement into the dashboard and make our dashboard interactive. Slicers allow the users to filter the data displayed on the dashboard in any way they like. For example, if we have a Slicer based on Zip Code, like the one shown in the image, when the user selects 91910, the dashboard will only display information in the zip code region 91910.

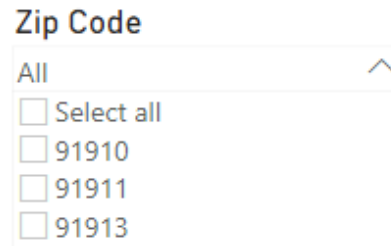


Figure 15 - Power BI Slicer

We can add a Slicer to the dashboard by clicking the 4th element on the 4th row in the Visualizations section (Figure 7). After that, we can simply insert Zip Code into the field section and we will have a slicer based on Zip Code.

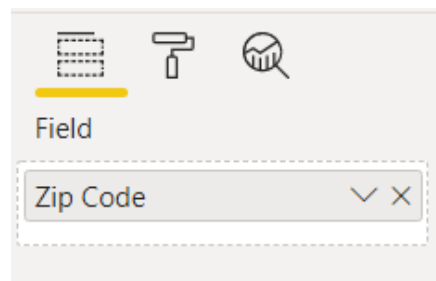


Figure 16 - Power BI Slicer Field

Furthermore, we can also edit which graphs will be affected by the Slicer. For instance, if we have a Bar Chart and a Table and we want our slicer to only change the data displayed by the Table and the Bar Chart doesn't change, we can select the Slicer, click the Format tab on the top of Power BI and then select Edit Interactions.

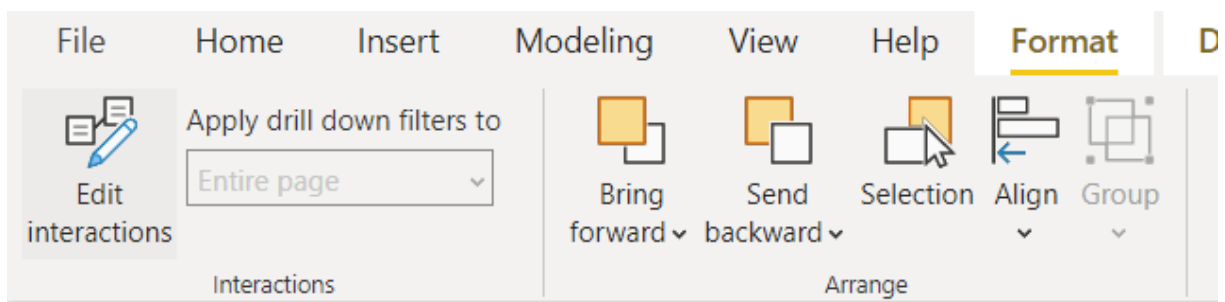


Figure 17 - Power BI Menu Bar, Edit Interactions

Then we will see there are 2 options in the top right corner of each graph we created. The one on the left is called ‘Filter’. If we have that selected, the Slicer will affect this graph. The one on the right is called ‘None’. If we have that selected, the Slicer will not affect this graph. So we are able to decide which graphs can be affected by the Slicers by selecting different options here.

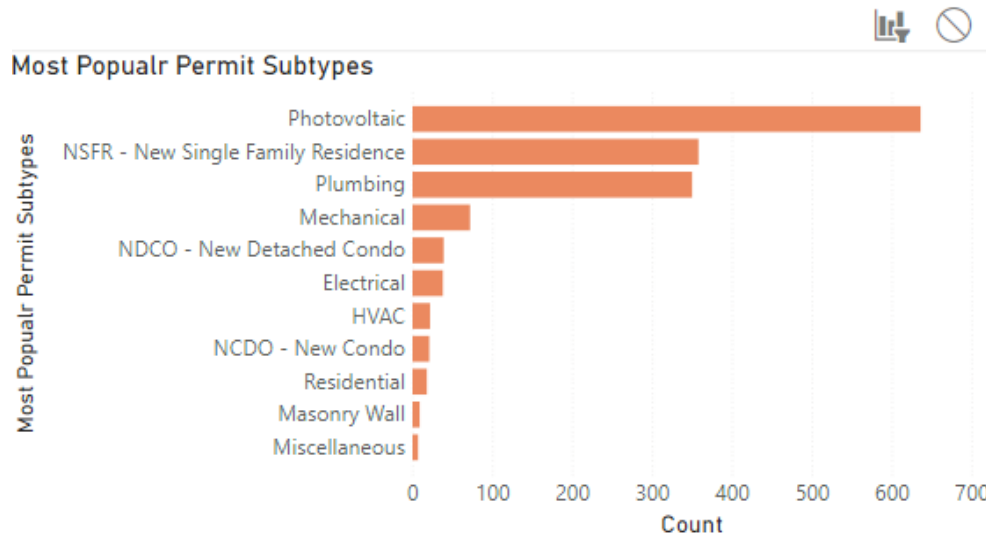


Figure 18 - Power BI Slicer On and Off

Part 4: Publish to Web Page

After designing the dashboard, the next step is to move the local .pbix file to your Power BI cloud space, then extract the HTML IFrame to embed into your web page .

Step 1: Publish to Power BI Workspace

Click the ‘Publish’ button, which will prompt you to ‘Save’ your file. Next, select the default workspace by clicking ‘Select’. If this dashboard has been published on your workspace in the past, clicking ‘Select’ will ask if you want to replace the older dashboard, which you can click ‘Replace’.

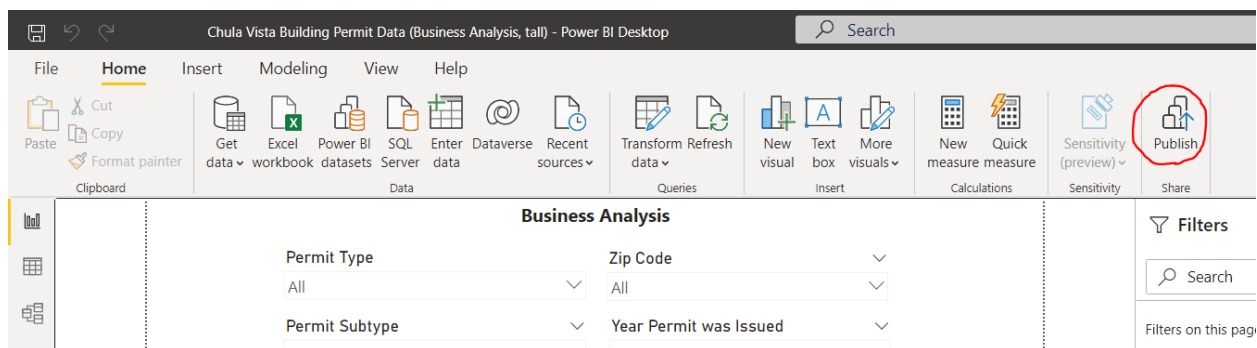




Figure 19 - Power BI Publish

Step 2: Get Each Dashboard's HTML IFrame

In your workspace, there are two versions of each dashboard you publish:  'report' and  'dataset'. Click on the 'report' version of a dashboard to access the interactive dashboard in your workspace.

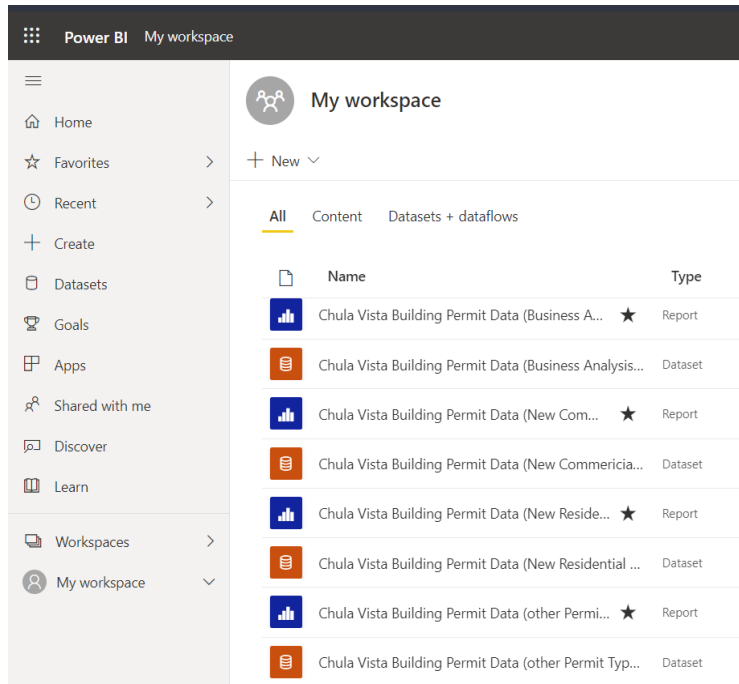


Figure 20 - Power BI Workspace

Next, click on File - Embed report - Publish to web (public)

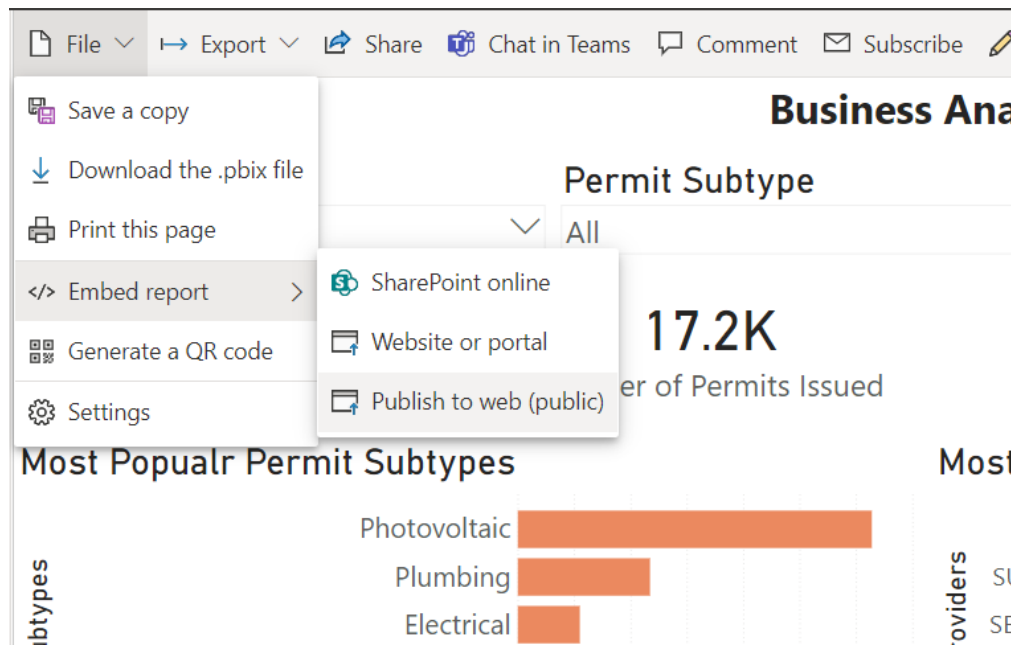


Figure 21 - Power BI Publish to Web

After clicking ‘Publish to web’, you will see 2 ‘Copy’ options. The first option is a link you can click on to open that dashboard in a browser, while the second option is the HTML IFrame that you can copy paste into your web page editor. Before copying the IFrame, change the width and height to suit the shape of your dashboard. We recommend at least 800 pixels wide and 636 or 836 pixels tall for best viewing.

Embed code

Link you can send in email

`https://app.powerbi.com/view?r=eyJrljoiY2MxYTYz`

Copy

HTML you can paste into a website

`<iframe width="600" height="373.5" src="https://a`

Copy

Figure 22 - Power BI Get HTML IFrame

Step 3: Embed into Web Page

After copying the IFrame, paste it into your web page editor and the interactive Power BI dashboard will show up on the web page. NOTE: The dashboard is still connected to the workspace you copied the IFrame from. If anyone updates that dashboard on that workspace, the dashboard on the web page will change to reflect the update.

The example Permit Data web page: [link](#)

This is the end of the “How to Create Interactive Dashboards” section. Have a great time making those dashboards!

Other Concerns

How often should we update the dashboards?

The city of Chula Vista already has a publicly-accessible permit database named [Accela](#). Citizens can access their permit records on Accela usually the day they apply. Therefore, it is not necessary that the permit data dashboards be updated daily or in real time. The update would unlikely affect the average plan review time. However, businesses might be interested in information from the most recent months, so it is helpful to make dashboards that pull data from live datasets and auto refresh to reflect the newest information. The permit data dashboards do not have auto refresh incorporated, but you can read more about [Data refresh and DirectQuery/LiveConnect mode in Power BI](#).

Are we relying too heavily on Power BI/ What if we stop using Power BI?

Besides Power BI, there are other commonly used platforms for creating interactive dashboards, such as Excel, Google Data Studio, Tableau, and iDashboard. You can even integrate Power BI and Google Data Studio with Panoply. Our “How to Create Interactive Dashboards” section solely uses Power BI, but it is by no means the only way to create embed-able live dashboards. So far, we have not found a convenient way to transfer dashboard designs from Power BI to other platforms, so if we stop using Power BI, it is likely someone will need to remake the dashboards.

ADA Compliance

According to [Interactive Accessibility](#), the Americans with Disabilities Act (ADA) states that “all electronic and information technology must be accessible to people with disabilities.” The permit data web page provides a general description for the dashboards and a text resizing option, but there is currently no method to audibly describe a dashboard interactive result.

Update to Chula Vista Color Scheme

Chula Vista is currently updating its website system. After the update is complete, it is best to standardize all fonts and color schemes to match the website design.

Appendix A - Tidy Data Problem Examples

Common dataset problems include:

- Column headers are not on row 1

Data Table #5			
Created on Jan 1, 2001			
ID	Name	Hometown	Number of pets owned
00001	A	Town 1	2
00002	B	Town 2	10
00003	C	Town 3	0

- Column headers are values, not variable names

Count of IPEDS institutions by sector and size (2017)

instsize	A Under 1,000	B 1,000 - 4,999	C 5,000 - 9,999	D 10,000 - 19,999	E 20,000 and above
sector					
Private for-profit, 4-yr or above	416	76	14	6	10
Private not-for-profit, 4-yr or above	785	711	108	53	22
Public, 4-yr or above	53	244	168	160	147

- Multiple variables are stored in one column

Country	Year	Column	Cases
A	2000	Male, ages 0-14	2
B	2000	Female, ages 14-28	10
C	2000	Female, ages 0-14	0

- Variables are stored in both rows and columns

City	date	Element	Temperature
A	2000-01-01	Minimum temperature	32
A	2000-01-01	Maximum temperature	57
B	2000-01-01	Minimum temperature	41
B	2000-01-01	Maximum temperature	65

- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

For more information on Tidy Data, please read this [Tidy Data guide](#) by Hadley Wickham.