Automating the Christmas Bird Count: Redux

John Hurley · Just now · 8 min read

What worked, what didn't

In a previous article, _Automating the Christmas Bird Count_, I described a vision of how this annual birding event could be automated. With the 121st Christmas Bird Count for 2020 behind us, let's see what worked and what didn't.

In this pandemic year, automation became a necessary part of the process. Some local counts, including San Francisco and Oakland, were completely cancelled due to public health orders. Other counts were able to take place but with a strict set of rules in place to ensure social distancing. It also meant that many counts actively encouraged the use of eBird for collecting data.

One consequence of this was that there were a large number of one or two person teams. This creates a lot more work for sector leaders and compilers. In more than one case, this lead to cancellation of the count. Using this tool, merging of the checklist becomes a task of minutes, not days.

Prior to the counts, I had prioritized work on the tools that I thought would be the most important to the count leaders. To use a popular paraphrase of a quote by Helmuth von Moltke the Elder, this plan didn't survive contact with the enemy. The software changed rapidly in the days leading up to the counts and afterward. At this point, it is producing accurate and useful data, and was greatly improved by the field testing.

The Name

Since it gets confusing just talking about "the tools", I was encourage to give the whole package a name. I'm calling it Gamma Corvi, after the brightest star in the constellation of Corvus. It's not such an unusual name, if you imagine that one of the purposes of the package is counting crows…

Summary of Gamma Corvi

The tools in Gamma Corvi are currently in the form of Jupyter Python notebooks. The two main tools that were used were Service-Parse and Service-Count. The primary task of Service-Parse is to produce an annotated, standardized version of the checklist for the count. This work was done ahead of time for the two counts I was participating in, San Jose (CASJ) and Calero-Morgan Hill (CACR). Many improvements came from similar work for Palo Alto (CAPA) and Monterey Peninsula (CAMP).

Group	CommonName	Total	Ad	Im	TaxonOrder	NACC_SORT _ORDE
Waterfowl	**Ross's Goose**	0			243	12
Waterfowl	**Cackling Goose**	0			286	21
Waterfowl	Canada Goose	0			297	22
Waterfowl	Wood Duck	0			397	34
Waterfowl	**Blue-winged Teal**	0			422	37
Waterfowl	Cinnamon Teal	0			423	38

Top of standardized checklist for CASJ 2020

The second tool, Service-Count, has multiple phases, which include retrieving data from eBird, creating detailed reports for each sector, then merging these into a summary report that can

be used to input data into the official Audubon site for compilers.

Merging

One of the greatest strengths of the whole package is the ability to merge count data into a master list, which is sorted taxonomically.

To illustrate this further, let's look at a FeederWatch example. The master checklist for CASJ has 206 species and variants. One set of FeederWatch data came in via email. This was converted manually to a CSV file with 32 species and counts, in no particular order. To enter this as a column in a count summary is very tedious: scanning the summary to find the species, entering the count and then on to the next species.

When using the Service-Count tool however, we just put the CSV file in the input directory and it was merged in as a separate column. Big win.

Summary Report

The summary report has six sheets, some of which I will show below. The main one of use for the compiler sums all the sub-totals from the sector summaries.

Section of the CASJ 2020 Final Checklist

Each sector has its own column that is adding in to the 'Total' column. When inputing data into the Audubon site, rows where total is zero and the 'Category' field is not "species" for are filtered out. Usually this is sorted by 'TaxonOrder', but sorting by 'NACC_SORT_ORDER' will arrange it in the same order as Audubon.

The rarities sheet shows a quick summary of species that were either explictily marked rare by the compiler, or were missing from the main checklist.

Rarities sheet for CASJ 2020

The individual details sheet shows useful information for every checklist filed within the count circle.

Section of Individual Details for CASJ 2020

Some key fields here are the location name, comments, and a clickable link to the checklist on eBird.

Autoparty

Autoparty is not some festival involving motor vehicles, it's the name I use to decscribe the automatic creation of parties based on eBird data. This is quite important this year, due to the large numbers of small parties forced by social distancing measures.

Section of the CASJ 2020 Autoparty sheet

This creates parties by grouping location ID and observation time with people on the corresponding checklists. Filling out party effort and time was identified as a major pain point by most of the leaders, and this year was worse. Autoparty can help.

Sector Summary

One feature of the sector summaries that would be very difficult to construct by hand is the grouping by location and time.

An illustration of this key feature is shown above, with some columns dropped for clarity. Bill birded two locations (L12060407 and L12988268) with his wife Kitty. Kitty is not shown because they created one checklist which they shared. The blue columns show a case where Bill and John both birded L370800. They were there at different times and didn't share a checklist. The first two locations use a SUM function, since there is no duplication. The last two columns use a MAX function, since we want to avoid counting birds twice. This leads us to this formula in the 'Total' column:
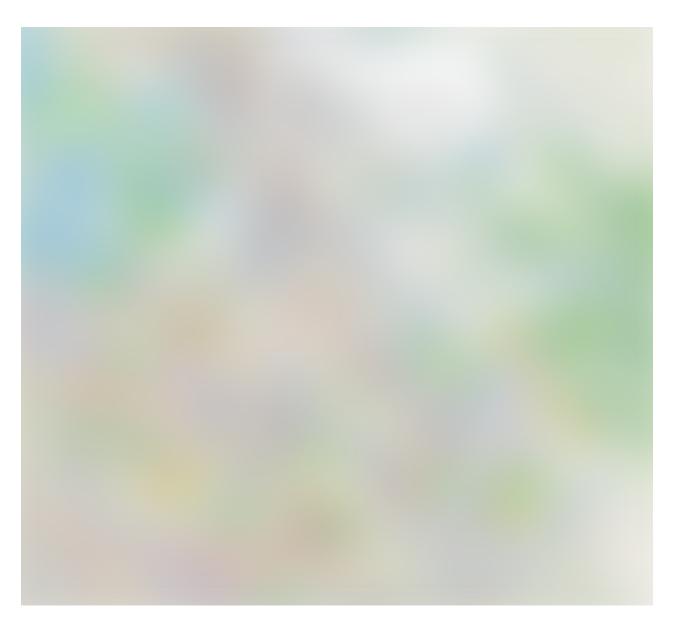
```
=SUM($I2:$Q2)+MAX($R2:$W2)
```

Note that I have hidden columns here, but $I through $Q are all of the non-duplicate locations, and $R through $W are for L370800. If there were more duplicate locations, those would each have a MAX grouping in the 'Total' formula.

If we had a good system in place for specifying parties, this could be used to generate the appropriate formula for the 'Total' field. In any event, this is certainly something to automate, given that 198 checklists were included in the CASJ 2020 count.

The Map

Often, the best way to present a large amount of data is with a graphic. This can be instructive, but an interactive graphic is even better. To get an overall view of where everyone was birding, I created a map using Folium.

CASJ CBC 2020 Activity

This shows every observation in eBird inside the count circle for San Jose, CA on 12–20–20. Circles with numbers are clusters of

observations, which can be revealed by clicking on them (in the interactive version of the map). There are several different layers, which can be toggled on and off.

As you can see from the map above, there were a lot of observers in the Evergreen sector, but no one birding the Berryessa and Calaveras sectors at all.

Individual observations have a marker with a pair of binoculars as an icon. The marker tooltip shows details of the observation, and includes a clickable link to the underlying eBird checklist. This article was the key to figuring out how to make the layers work properly.

I think this can be a great planning tool for the next year's count.

Rules for eBird

Cornell came out with an article recently _eBirding your Christmas Bird Count_. This mainly describes how to use eBird for accurate counting. In addition to their suggestions, I would add these:

- Start a new checklist at every new spot, or if the party splits.

- Compilers and sector leaders should distribute a list of preferred location/hotspot names. This makes it easier to decide if multiple people birded the same spot. Note that there is code that considers locations the same if they are within 150 meters of each other, so this helps somewhat.

- Don't use 'X' for species count; make a rough guess instead (10, 50, 100, 1000, etc.). I set 'X' equal to 3 (one, two, many); see *"The Pirahã people"*.

Wins

- Merging data from multiple checklists into a main one is a royal pain to do by hand. Using Gamma Corvi for this, even if a big chunk of data was not coming from eBird, is worth it all by itself.

- "Time saved by using Gamma Corvi let us participate in another count", from one participant.

- Gamma Corvi was used to supply data for three count circles. Not bad for an alpha version.

Losses

- Fillable PDFs should be an easy win, but people seemed reluctant to use them. At a minimum, Audubon should make the Rare Bird Form fillable (I have one in the GitHub repository).

- Additional data such as FeederWatch data supplied as CSV files still involves some manual steps.

- eBird has a limit of 200 checklists, which is for an entire county for a day. We luckily squeaked in under the wire for this, but this limit clearly has to be increased.

- The process for managing party hours and effort is still tedious, but the AutoParty feature helps. Participant list management is an area that needs attention.

- This code needs to be made into a web service to make it accessible to the sector leaders and compilers in the count. For example, Cornell may wish to pick this up and host it on their servers.

My favorite features

- The merging function for time savings

- The map for planning

- The AutoParty sheet

See the GitHub repository for code and examples. I plan to improve the code and hope to use it soon for Spring/Summer counts and Birdathons.

References

jhurley13/automatingcbc

Automating the Christmas Bird Count The Christmas Bird Count has been a tradition for 120 years. Started...

github.com

Christmas Bird Count

Christmas Bird Count Curious about what the
Christmas Bird Count data show? Audubon's Science...

www.audubon.org

eBirding your Christmas Bird Count [updated] - eBird

Christmas Bird Count (CBC) season is here! Each year,
tens of thousands of birders throughout the western...

ebird.org

Christmas Bird Count   Audubon   Data Science   Ebird   Towards Data Science