Version and share data and models: remote storage

In the previous lessons, we explored how to add data to our DVC cache through the `dvc add` command. What we did not mention yet, however, is that DVC also tracks data automatically. As we learned in module 3, the `dvc.yaml` defines our pipeline through dependencies and outputs. All outputs in our pipeline are automatically added to the DVC cache and versioned.

The DVC cache we have thus far been using is located on our local machine. Typically, however, we would also want to version our data on a remote storage (or simply: *remote*).

The remote is where DVC sends our data for persistent storage. It is analogous to a remote Git repository (e.g. a repository on GitHub or GitLab). DVC supports a wide variety of remote storages, including Amazon S3, Azure Blob Storage, and Google Cloud Storage. For a full list of the supported remotes, please refer to the DVC documentation.

`dvc remote add`

We can add a remote using the `dvc remote add` command (optionally with the `-d` option to set it as our new default). The required details vary slightly for every kind of remote. In the example below we create a directory that will serve as a *local remote*. While located on the same machine, the remote is in a separate location from our DVC cache. Once we have added our remote (here named *myremote*), we commit this change in our DVC configuration to our Git history.

```
mkdir -p /tmp/dvc
dvc remote add -d myremote /tmp/dvc
git commit .dvc/config -m "Configure local remote"
```

Adding a remote on a different machine is quite similar, although it requires a little more configuration. Take an Amazon S3 remote, for example. This can be added to our DVC configuration through the following command:

```
dvc remote add -d myremote s3://mybucket/path
```

In this case, however, we would also need to create a bucket, set our AWS CLI authentication credentials, and set a few specific permissions on AWS. Refer to the DVC docs for comprehensive instructions for your remote of choice.

`dvc push` and `dvc pull`

Once we have set up our remote, we can synchronize the data between our local cache and our remote for persistent storage.

We do so just like how we would synchronize the code between our local Git repository and our remote Git repository. While we use `git push` and `git pull` for code, we use `dvc push` and `dvc pull` respectively for our data. Similarly, `dvc add` and `dvc commit` exist as counterparts to `git add` and `git commit`.