Let’s go into my Photos folder in my home directory and take a look around:

```
pwd

## /Users/sean

ls

## Code
## Documents
## Photos
## Desktop
## Music
## todo-2017-01-24.txt
cd Photos
ls

## 2016-06-20-datasci01.png
## 2016-06-20-datasci02.png
## 2016-06-20-datasci03.png
## 2016-06-21-lab01.jpg
## 2016-06-21-lab02.jpg
## 2017-01-02-hiking01.jpg
## 2017-01-02-hiking02.jpg
## 2017-02-10-hiking01.jpg
## 2017-02-10-hiking02.jpg
```

I’ve just been dumping pictures and figures into this folder without organizing them at all! Thankfully (in the words of Dr. Jenny Bryan) I have an unwavering commitment to the ISO 8601 date standard so at least I know when these photos were taken. Instead of using mv to move around each individual photo I can select groups of photos using the * wildcard. A **wildcard** is a character that represents other characters, much like how joker in a deck of cards can represent other cards in the deck. Wildcards are a subset of metacharacters, a topic which we will discuss in detail later on in this chapter. The * (“star”) wildcard represents *zero or more of any character*, and it can be used to match names of files and folders in the command line. For example if I wanted to list all of the files in my Photos directory which have a name that starts with “2017” I could do the following:

```
ls 2017*

## 2017-01-02-hiking01.jpg
## 2017-01-02-hiking02.jpg
## 2017-02-10-hiking01.jpg
## 2017-02-10-hiking02.jpg
```

Only the files starting with “2017” are listed! The command ls 2017* literally means: list the files that start with “2017” followed by zero or more of any character. As you can imagine using wildcards is a powerful tool for working with groups of files that are similarly named.

Let’s walk through a few other examples of using the star wildcard. We could only list the photos starting with “2016”:

```
ls 2016*

## 2016-06-20-datasci01.png
## 2016-06-20-datasci02.png
## 2016-06-20-datasci03.png
## 2016-06-21-lab01.jpg
## 2016-06-21-lab02.jpg
```

We could list only the files with names ending in .jpg:

```
ls *.jpg

## 2016-06-21-lab01.jpg
## 2016-06-21-lab02.jpg
## 2017-01-02-hiking01.jpg
## 2017-01-02-hiking02.jpg
## 2017-02-10-hiking01.jpg
## 2017-02-10-hiking02.jpg
```

In the case above the file name can start with a sequence of zero or more of any character, but the file name must end in .jpg. Or we could also list only the first photos from each set of photos:

```
ls *01.*

## 2016-06-20-datasci01.png
## 2016-06-21-lab01.jpg
## 2017-01-02-hiking01.jpg
## 2017-02-10-hiking01.jpg

```

All of the files above have names that are composed of a sequence of characters, followed by the adjacent characters 01., followed by another sequence of characters. Notice that if I had entered ls *01* into the console every file would have been listed since 01 is a part of all of the file names in my Photos directory.

Let’s organize these photos by year. First let’s create one directory for each year of photos:

```
mkdir 2016
mkdir 2017

```

Now we can move the photos using wildcards:

```
mv 2017-* 2017/
ls

## 2016
## 2016-06-20-datasci01.png
## 2016-06-20-datasci02.png
## 2016-06-20-datasci03.png
## 2016-06-21-lab01.jpg
## 2016-06-21-lab02.jpg
## 2017
```

Notice that I've moved all files that start with "2017-" into the 2017 folder! Now let's do the same thing for files with names starting with "2016-":

```
mv 2016-* 2016/
ls

## 2016
## 2017
```

Finally my photos are somewhat organized! Let's list the files in each directory just to make sure all was moved as planned:

```
ls 2016/

## 2016-06-20-datasci01.png
## 2016-06-20-datasci02.png
## 2016-06-20-datasci03.png
## 2016-06-21-lab01.jpg
## 2016-06-21-lab02.jpg

ls 2017/

## 2017-01-02-hiking01.jpg
## 2017-01-02-hiking02.jpg
## 2017-02-10-hiking01.jpg
## 2017-02-10-hiking02.jpg
```

Looks good! There are a few more wildcards beyond the star wildcard which we'll discuss in the next section where searching file names gets a little more advanced.

Summary

- Wildcards can represent many kinds and numbers of characters.
- The star wildcard (*) represents zero or more of any character.
- You can use wildcards on the command line in order to work with multiple files and folders.

Mark as completed