**#TODO: 5: Calculating Results**

**Implement the calculates\_results\_stats() function to calculate the results of run and put statistics in a results statistics dictionary (results\_stats\_dic).**

**Coding within the *check\_images.py* and *calculates\_results\_stats.py***

**Code to Edit**

This section will help you code the *undefined* function **calculates\_results\_stats** within ***calculates\_results\_stats.py***. With this function you will be inputing the results dictionary to create a dictionary of results statistics. This results statistics dictionary will contain the *statistic's name* as the *key* and the *value* will simply be the *statistic's numeric* value.

You will be creating the results statistics dictionary within **calculates\_results\_stats**. This means we recommend that you create this dictionary as the first line in your function and return it's value with the last line in your function. For a more detailed explanation regarding why this dictionary is returned by the function, review the section *Mutable Data Types and Functions*.

* Code for the function definition def calculates\_results\_stats(): as indicated by #TODO: 5 within ***calculates\_results\_stats.py***
  + Using the comments and the docstring within ***calculates\_results\_stats.py*** to define **calculates\_results\_stats**

**Expected Outcome**

When completed, this code will be able to provide counts and percentages that will be used to answer the objectives of this lab. The percentages provided by the *results\_stats* dictionary will answer objectives 1 and 2. The counts will be used to compute the percentages.

**Checking your code**

The **check\_calculating\_results** function within ***check\_images.py*** will check your code by recalculating the following results statistics (counts and percentages) and comparing them to the results statistics that you calculated and stored within the results statistics dictionary.

Results Statistics checked:

* **Counts**:
  + Number of Images
  + Number of Dog Images
  + Number of "Not-a" Dog Images
* **Percentages**:
  + % Correctly Classified Dog Images
  + % Correctly Classified "Not-a" Dog Images
  + % Correctly Classified Breeds of Dog Images

Visually check that the results from this *checking* code for these six statistics above *match* the results you computed with **calculates\_results\_stats** within ***calculates\_results\_stats.py***.

**Project Workspace - Calculating Results**

* The next concept will have your workspace to work on #TODO: 5
* Editing of ***check\_image.py*** and ***calculates\_results\_stats.py*** can be done within the **Project Workspace - Calculating Results**

**For additional information and help on #TODO: 5, please look at the information below:**

**How to Define Percentages for Summarizing the Results**

**Principle Objectives \_1\_ and \_2\_:**

1. Correctly identify which pet images are of dogs (even if breed is misclassified) and which pet images aren't of dogs.
2. Correctly classify the breed of dog, for the images that are of dogs.

To achieve objectives \_1\_ and \_2\_, your program will need to be able to calculate the following percentages based upon the results of comparing the labels contained within the results dictionary.

The **results dictionary** will have the following format:

* *key* = pet image filename (ex: Beagle\_01141.jpg)
* *value* = List with:
  + index 0 = Pet Image Label (ex: beagle)
  + index 1 = Classifier Label (ex: english foxhound)
  + index 2 = 0/1 where 1 = labels match , 0 = labels don't match (ex: 0)
  + index 3 = 0/1 where 1= Pet Image Label is a dog, 0 = Pet Image Label isn't a dog (ex: 1)
  + index 4 = 0/1 where 1= Classifier Label is a dog, 0 = Classifier Label isn't a dog (ex: 1)
* example\_dictionary = {'Beagle\_01141.jpg': ['beagle', 'walker hound, walker foxhound', 0, 1, 1]}

You will be storing these calculations (counts & percentages) in the results statistics dictionary. We recommend using the same prefix for all counts (e.g. *n\_*) and percentages (e.g. *pct\_*) in the statistic's name(*key*) as to make it easier to print all of them for each group.

The **results statistics dictionary** will have the following format:

* *key* = statistic's name (e.g. n\_correct\_dogs, pct\_correct\_dogs, n\_correct\_breed, pct\_correct\_breed)
* *value* = statistic's value (e.g. 30, 100%, 24, 80%)
* example\_dictionary = {'n\_correct\_dogs': 30, 'pct\_correct\_dogs': 100.0, 'n\_correct\_breed': 24, 'pct\_correct\_breed': 80.0}

**Counts Computed from the *Results* dictionary for input into the *Results Statistics* dictionary:**

* **Z**: Number of Images
  + length of *results\_dic*, because filenames = key
* **A**: Number of Correct Dog matches
  + Both labels are of dogs: *results\_dic*[key][3] = 1 and *results\_dic*[key][4] = 1
* **B**: Number of Dog Images
  + Pet Label is a dog: *results\_dic*[key][3] = 1
* **C**: Number of Correct Non-Dog matches
  + Both labels are NOT of dogs: *results\_dic*[key][3] = 0 and *results\_dic*[key][4] = 0
* **D**: Number of Not Dog Images
  + number images - number dog images --OR--
  + Pet Label is NOT a dog: *results\_dic*[key][3] = 0
* **E**: Number of Correct Breed matches
  + Pet Label is a dog & Labels match: *results\_dic*[key][3] = 1 and *results\_dic*[key][2] = 1
* (*Optional*) **Y**: Number of label matches
  + Labels match: *results\_dic*[key][2] = 1

**Compute a Summary of the Percentages from the *Results Statistics* dictionary counts:**

* **Objective \_1\_a**: Percentage of Correctly Classified Dog Images
  + **A** Correctly classified *dog* images.
  + **B** Number of *dog* images
  + Percentage of correctly classified "*dog*" images: **A**/**B** \* 100
* **Objective \_1\_b**: Percentage of Correctly Classified Non-Dog Images
  + **C** Correctly classified *NOT a dog* images.
  + **D** Number of *NOT a dog* images
  + Percentage of correctly classified "*Non-dog*" images: **C**/**D** \* 100
* **Objective \_2\_**: Percentage of Correctly Classified Dog Breeds
  + **E** Correctly classified as a particular breed of *dog* images.
  + **B** Number of *dog* images
  + Percentage of correctly classified *Dog* Breed images: **E**/**B** \* 100
* (*Optional*): Percentage Label Matches ( regardless if they're a dog)
  + **Y** Number of label matches
  + **Z** Number of images
  + Percentage of correctly Matched Images ( regardless if they are a dog): **Y**/**Z** \* 100

**Important Notes**:

* You will need to initialize all the counts to a value of zero before iterating through *results* dictionary. As you iterate through *results* dictionary, if certain criterion are met you will need to increment these counters by 1.
* The percentages (and total number of images) can be generated from the counts (see percentage & count calculations above); therefore, these values should be calculated *after* counts have been calculated by iterating through the *results* dictionary.
* When calculating the percentage of correctly classified Non-Dog Images, use a conditional statement to check that **D**, the number of "not-a-dog" images, is greater than zero. To avoid division by zero error, only if **D** is greater than zero should **C**/**D** be computed; otherwise, this should be set to 0.
* Because the *Results Statistics* dictionary is created inside of the function and is a mutable object, you will need to *return* it's value at the end of the function (see section *Mutable Data Types and Functions*).