

Laboratory Manual
of
BIG DATA ANALYTICS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AVANTHI INSTITUTE OF ENG & TECH

Guntapally(V), Abdullaput met (M), RR Dist.

LIST OF EXPERIMENTS

1. JAVA/HADOOP INSTALLATION
2. IMPLEMENT SIMPLE MAP REDUCE JOB
3. HBASE INSTALLATION
4. PROCESS BIG DATA IN HBASE
5. INSTALLATION OF PIG
6. STORE AND RETREIVE DATA IN PIG
7. USING POWER PIVOT(EXCEL SHEET) PERFORM
 - A) BIG DATA ANALYTICS
 - B) BIG DATA CHARTING

EXPERIMENT 1

JAVA/HADOOP INSTALLATION

Steps to Install Hadoop

- Install Java JDK 1.8
- Download Hadoop and extract and place under C drive
- Set Path in Environment Variables
- Config files under Hadoop directory
- Create folder datanode and namenode under data directory
- Edit HDFS and YARN files
- Set Java Home environment in Hadoop environment
- Setup Complete. Test by executing start-all.cmd

There are two ways to install Hadoop, i.e.

- **Single node**
- **Multi node**

Single node cluster means only one DataNode running and setting up all the NameNode, DataNode, ResourceManager and NodeManager on a single machine.

This is used for studying and testing purposes.

- So for testing whether the Oozie jobs have scheduled all the processes like collecting, aggregating, storing and processing the data in a proper sequence, we use single node cluster.
- It can easily and efficiently test the sequential workflow in a smaller environment as compared to large environments which contains terabytes of data distributed across hundreds of machines.

While in a Multi node cluster, there are more than one DataNode running and each DataNode is running on different machines. The multi node cluster is practically used in organizations for analyzing Big Data. In real time when we deal with petabytes of data, it needs to be distributed across hundreds of machines to be processed. Thus, here we use multi node cluster.

Setting up a single node Hadoop cluster

Prerequisites to install Hadoop on windows

- *VIRTUAL BOX (For Linux)*: it is used for installing the operating system on it.

- *OPERATING SYSTEM*: You can install Hadoop on Windows or Linux based operating systems. Ubuntu and CentOS are very commonly used.
- *JAVA*: You need to install the Java 8 package on your system.
- *HADOOP*: You require Hadoop latest version


1. Install Java

– Java JDK Link to download

<https://www.oracle.com/java/technologies/javase-jdk8-downloads.html>

– extract and install Java in C:\Java

– open cmd and type -> javac -version

 Command Prompt

```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\asus>javac -version
javac 1.8.0_241
```

2. Download Hadoop

– <https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

– extract to C:\Hadoop

EXPERIMENT 2

Write a simple program for Word Count Using Map Reduce Programming

OBJECTIVE

In MapReduce word count example, we find out the frequency of each word. Here, the role of Mapper is to map the keys to the existing values and the role of

Reducer is to aggregate the keys of common values. So, everything is represented in the form of Key-value pair.

THEORY

In Hadoop, Map Reduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to compute final results.

Pre-requisite

Make sure that Hadoop is installed on your system with the Java SDK, ECLIPSE editor. For all Experiment.

Steps

1.Open Eclipse> File > New > Java Project >(Name it – MRProgramsDemo) > Finish.

2.Right Click > New > Package (Name it - PackageDemo) > Finish.

3.Right Click on Package > New > Class (Name it - WordCount).

4.Add Following Reference Libraries:

1.Right Click on Project > Build Path> Add External

2./usr/lib/hadoop-0.20/hadoop-core.jar

3.Usr/lib/hadoop-0.20/lib/Commons-cli-1.2.jar

5.Type the following code: package PackageDemo;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

```
import org.apache.hadoop.io.IntWritable; import
org.apache.hadoop.io.LongWritable; import
org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job; import
org.apache.hadoop.mapreduce.Mapper; import
org.apache.hadoop.mapreduce.Reducer;
import
org.apache.hadoop.mapreduce.lib.input.FileInputForma
t; import
org.apache.hadoop.mapreduce.lib.output.FileOutputFor
mat; import
org.apache.hadoop.util.GenericOptionsParser;
```

```
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration(); String[] files=new
GenericOptionsParser(c,args).getRemainingArgs(); Path
input=new Path(files[0]);
Path output=new Path(files[1]); Job j=new
Job(c,"wordcount"); j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends
Mapper<LongWritable, Text, Text, IntWritable>{
```

```

public void map(LongWritable key, Text value, Context
con) throws IOException, InterruptedException
{
String line = value.toString(); String[]
words=line.split(",");

{
Text outputKey = new Text(word.toUpperCase().trim());
IntWritable outputValue = new IntWritable(1);
con.write(outputKey, outputValue);
}
}
}

```

```

public static class ReduceForWordCount extends
Reducer<Text, IntWritable, Text, IntWritable>
{
public void reduce(Text word, Iterable<IntWritable>
values, Context con) throws IOException,
InterruptedException
{
int sum = 0;
for(IntWritable value : values)
{
sum += value.get();
}
con.write(word, new IntWritable(sum));
}

}

```

```
}
```

The above program consists of three classes:

- Driver class (Public, void, static, or main; this is the entry point).
- The Map class which extends the public class `Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT>` and implements the Map function.
- The Reduce class which extends the public class `Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT>` and implements the Reduce function.

6. Make a jar file

Right Click on Project > Export > Select export destination as Jar File > next > Finish.

7. Take a text file and move it into HDFS format:

To move this into Hadoop directly, open the terminal and enter the following commands:

9. Open the result:

```
[training@localhost ~]$ hadoop fs -ls MRDir1
```

Found 3 items

```
-rw-r--r-- 1 training supergroup    0 2022-02-23 03:36
```

```
/user/training/MRDir1/_SUCCESS
```

```
drwxr-xr-x - training supergroup    0 2022-02-23 03:36
```

```
/user/training/MRDir1/_logs
```

```
-rw-r--r-- 1 training supergroup   20 2022-02-23 03:36
```

```
/user/training/MRDir1/part-r-00000
```

```
[training@localhost ~]$ hadoop fs -cat MRDir1/part-r-
```

```
00000 BUS      7
```

```
CAR 4
```

```
TRAIN    6
```

EXPERIMENT 3

HBase Installation Steps:

Step 1:

Unzip the downloaded Hbase and place it in some common path, say C:/Document/hbase-2.2.5

Step 2:

Create a folders as shown below inside root folder for HBase data and zookeeper

-> C:/Document/hbase-2.2.5/hbase

-> C:/Document/hbase-2.2.5/zookeeper

Step 3:

Open C:/Document/hbase-2.2.5/bin/hbase.cmd in notepad++. Search for below given lines and remove %HEAP_SETTINGS% from that line as dictated in the video embedded with this blog

```
set
java_arguments=%HEAP_SETTINGS% %HBASE_OPTS% -
classpath "%CLASSPATH%" %CLASS% %hbase-command-
arguments%
```

Step 4:

Open C:/Document/hbase-2.2.5/conf/hbase-env.cmd in notepad++. Add the below lines to the file after the comment session as shown in the YT video given with this tutorial.

```
set JAVA_HOME=%JAVA_HOME%
set HBASE_CLASSPATH=%HBASE_HOME%\lib\client-
facing-thirdparty\*
set HBASE_HEAPSIZE=8000
set HBASE_OPTS="-XX:+UseConcMarkSweepGC" "-
Djava.net.preferIPv4Stack=true"
set SERVER_GC_OPTS="-verbose:gc" "-
XX:+PrintGCDetails" "-
XX:+PrintGCDateStamps" %HBASE_GC_OPTS%
set HBASE_USE_GC_LOGFILE=true
```

```
set HBASE_JMX_BASE="-
Dcom.sun.management.jmxremote.ssl=false" "-
Dcom.sun.management.jmxremote.authenticate=false"
```

```

set HBASE_MASTER_OPTS=%HBASE_JMX_BASE% "-
Dcom.sun.management.jmxremote.port=10101"
set HBASE_REGIONSERVER_OPTS=%HBASE_JMX_BASE%
"-Dcom.sun.management.jmxremote.port=10102"
set HBASE_THRIFT_OPTS=%HBASE_JMX_BASE% "-
Dcom.sun.management.jmxremote.port=10103"
set HBASE_ZOOKEEPER_OPTS=%HBASE_JMX_BASE% -
Dcom.sun.management.jmxremote.port=10104"
set
HBASE_REGIONSERVERS=%HBASE_HOME%\conf\region
servers
set HBASE_LOG_DIR=%HBASE_HOME%\logs
set HBASE_IDENT_STRING=%USERNAME%
set HBASE_MANAGES_ZK=true

```

Step 5:

Open C:/Document/hbase-2.2.5/conf/hbase-site.xml
notepad++. Add the below lines inside <configuration>
tag. Refer YT video given in this tutorial.

```

<property>
  <name>hbase.rootdir</name>
  <value>file:///C:/Documents/hbase-
2.2.5/hbase</value>
</property>
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/C:/Documents/hbase-
2.2.5/zookeeper</value>
</property>
<property>

```

```
<name> hbase.zookeeper.quorum</name>  
<value>localhost</value>  
</property>
```

Step 6:

Setup the Environment variable for HBASE_HOME and add bin to the path variable as shown in the below image.

We are done with the Setup for HBase on Windows 10 and to verify the setup follow the video and also try creating a table inside hbase shell. Hope you made a complete standalone setup of HBase in windows machine successfully. If in case of any issue, please let me know through the below comment box.

Data Definition Commands

Following are some Data Definition HBase Commands:

i. alter

We use alter command to add/modify/delete column families, also to change table configuration.

a. Add/Change column family

Let's see an example, in order to change or add the 'f1' column family in table 't1' from the current value to keep a maximum of 5 cell VERSIONS, use this command-

```
hbase> alter 't1', NAME => 'f1', VERSIONS => 5
```

Moreover, on several column families, we can operate it:

```
hbase> alter 't1', 'f1', {NAME => 'f2', IN_MEMORY => true}, {NAME => 'f3', VERSIONS => 5}
```

b. Delete column family

Use one of following to delete the 'f1' column family in table 'ns1:t1':

```
hbase> alter 'ns1:t1', NAME => 'f1', METHOD => 'delete'  
hbase> alter 'ns1:t1', 'delete' => 'f1'
```

c. Alter Table Properties

Also, it is possible to change table-scope attributes such as MAX_FILESIZE, READONLY, MEMSTORE_FLUSH_SIZE, DEFERRED_LOG_FLUSH, etc. we can put these at the end;

for example:

Now, in order to change the max size of a region to 128MB, use this command:

```
hbase> alter 't1', MAX_FILESIZE => '134217728'
```

ii. alter_async

Only one difference between alter command and alter_async is, alter_async does not wait for all regions to receive the schema changes.

iii. alter_status

Alter_status command gets the status of the alter command. Moreover, it indicates the number of regions of the table which have received the updated schema.

For Example:

```
hbase> alter_status 't1'  
hbase> alter_status 'ns1:t1'
```

iv. Create

For Creating tables, we use it. Also, we can Pass a table name, and a set of column family specifications (at least one), and, optionally, table configuration as arguments.

For Examples:

a. Create a table;

Along with the namespace=ns1 and table
qualifier/name=t1

```
hbase> create 'ns1:t1', {NAME => 'f1', VERSIONS => 5}
```

b. Create a table;

Along with namespace=default and table qualifier=t1

```
hbase> create 't1', {NAME => 'f1'}, {NAME => 'f2'},  
{NAME => 'f3'}
```

hbase> # The above in shorthand would be the following:

```
hbase> create 't1', 'f1', 'f2', 'f3'
```

```
hbase> create 't1', {NAME => 'f1', VERSIONS => 1, TTL =>  
2592000, BLOCKCACHE => true}
```

```
hbase> create 't1', {NAME => 'f1', CONFIGURATION =>  
{ 'hbase.hstore.blockingStoreFiles' => '10' }}
```

c. Table configuration options can be put at the end.

```
hbase> create 'ns1:t1', 'f1', SPLITS => ['10', '20', '30', '40']
```

```
hbase> create 't1', 'f1', SPLITS => ['10', '20', '30', '40']
```

```
hbase> create 't1', 'f1', SPLITS_FILE => 'splits.txt',  
OWNER => 'johndoe'
```

```
hbase> create 't1', {NAME => 'f1', VERSIONS => 5},  
METADATA => { 'mykey' => 'myvalue' }
```

hbase> # Optionally pre-split the table into
NUMREGIONS, using

```
hbase> # SPLITALGO ("HexStringSplit", "UniformSplit" or  
classname)
```



```
hbase> create 't1', 'f1', {NUMREGIONS => 15, SPLITALGO  
=> 'HexStringSplit'}  
hbase> create 't1', 'f1', {NUMREGIONS => 15, SPLITALGO  
=> 'HexStringSplit', CONFIGURATION =>  
{'hbase.hregion.scan.loadColumnFamiliesOnDemand' =>  
'true'}}
```

d. Also, it is possible to keep around a reference to the created table.

```
hbase> t1 = create 't1', 'f1'
```

Which gives a reference to the table named 't1', on which we can then call methods `t1.scan`, `t1.get`.

EXPERIMENT 5

Installing Apache Pig 0.17.0 on Windows 10

This article is a part of a series that we are publishing on TowardsDataScience.com that aims to illustrate how to install Big Data technologies on Windows operating system.

Previously published:

- [Installing Hadoop 3.2.1 Single node cluster on Windows 10](#)
- [Installing Apache Hive 3.1.2 on Windows 10](#)

In this article, we will provide a step-by-step guide to install Apache Pig 0.17.0 on Windows 10.

1. Prerequisites

1.1. Hadoop Cluster Installation

Apache Pig is a platform build on the top of Hadoop. You can refer to our previously published article to install a Hadoop single node cluster on Windows 10.

Note that the Apache Pig latest version 0.17.0 [supports Hadoop 2.x versions](#) and [still facing some compatibility](#)

[*issues with Hadoop 3.x*](#). In this article, we will only illustrate the installation since we are working with Hadoop 3.2.1

1.2. 7zip

7zip is needed to extract .tar.gz archives we will be downloading in this guide.

2. Downloading Apache Pig

To download the Apache Pig, you should go to the following link:

- <https://downloads.apache.org/pig/>

Pig Releases

Please make sure you're downloading from [a nearby mirror site](#), not from www.apache.org.

Older releases are available from the [archives](#).






Name	Last modified	Size	Description
 Parent Directory		-	
 latest/	2018-05-04 17:41	-	
 pig-0.16.0/	2018-05-04 17:38	-	
 pig-0.17.0/	2018-05-04 17:41	-	
 KEYS	2017-06-19 08:12	11K	

Figure 1 — Apache Pig releases directory

If you are looking for the latest version, navigate to “latest” directory, then download the pig-x.xx.x.tar.gz file.

Index of /pig/latest








Name	Last modified	Size	Description
 Parent Directory		-	
 RELEASE_NOTES.txt	2017-06-16 18:10	1.9K	
 pig-0.17.0-src.tar.gz	2017-06-16 18:11	15M	
 pig-0.17.0-src.tar.gz.asc	2017-06-16 18:11	488	
 pig-0.17.0-src.tar.gz.md5	2017-06-16 18:11	56	
 pig-0.17.0.tar.gz 	2017-06-16 18:10	220M	
 pig-0.17.0.tar.gz.asc	2017-06-16 18:11	488	
 pig-0.17.0.tar.gz.md5	2017-06-16 18:11	52	

Figure 2 — Download Apache Pig binaries

After the file is downloaded, we should extract it twice using 7zip (*using 7zip: the first time we extract the .tar.gz file, the second time we extract the .tar file*). We will extract the Pig folder into “E:\hadoop-env” directory as used in the previous articles.

3. Setting Environment Variables

After extracting Derby and Hive archives, we should go to Control Panel > System and Security > System. Then Click on “Advanced system settings”.

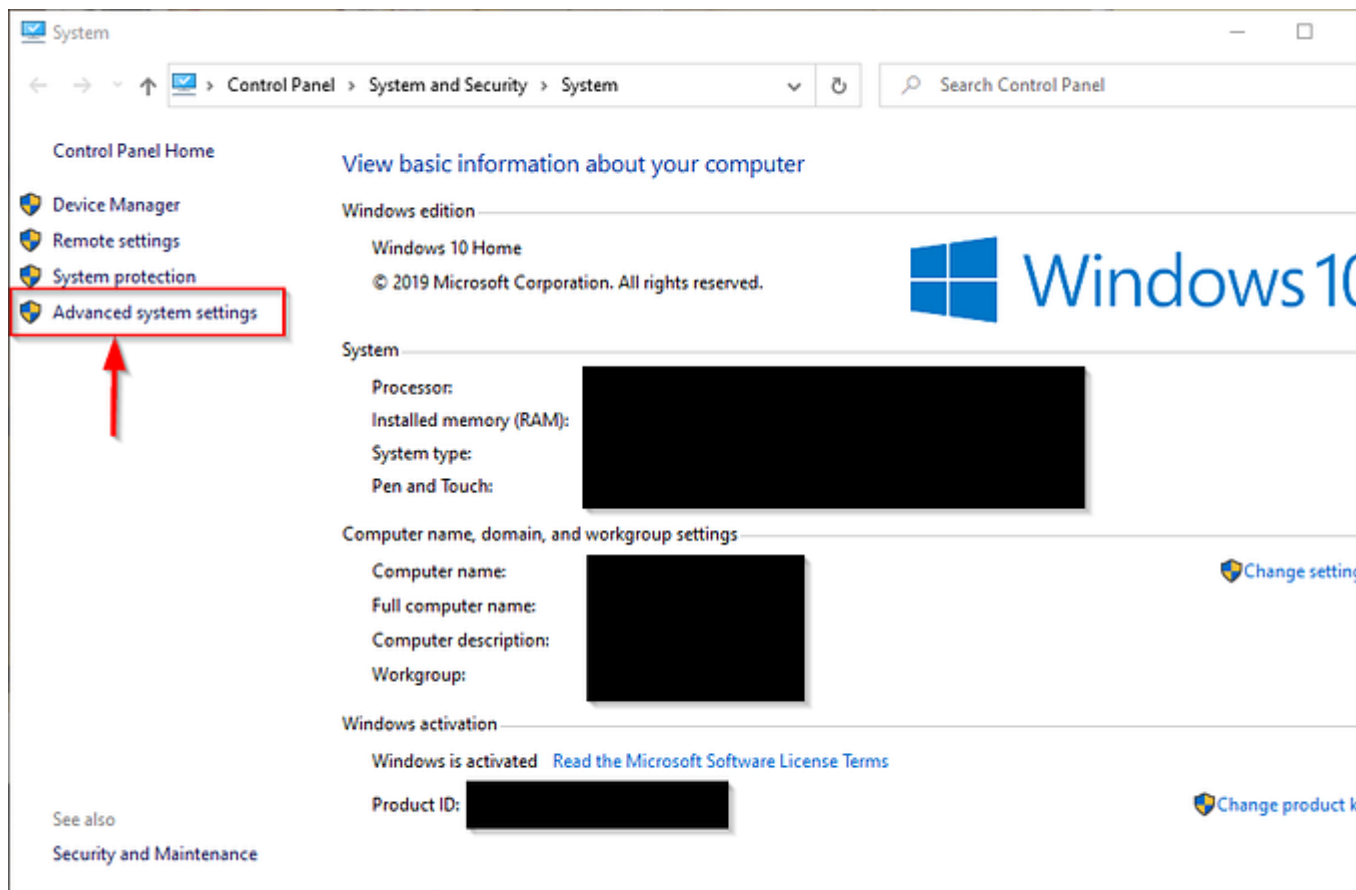


Figure 3 — Advanced system settings

In the advanced system settings dialog, click on “Environment variables” button.

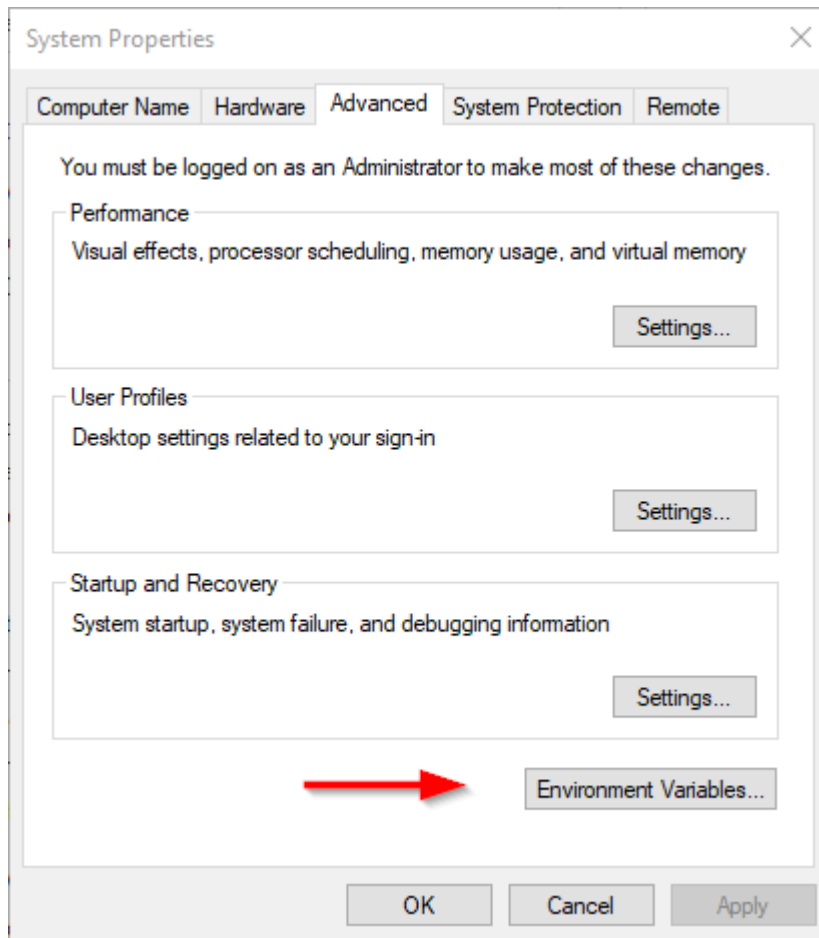


Figure 4 — Opening environment variables editor

Now we should add the following user variables:

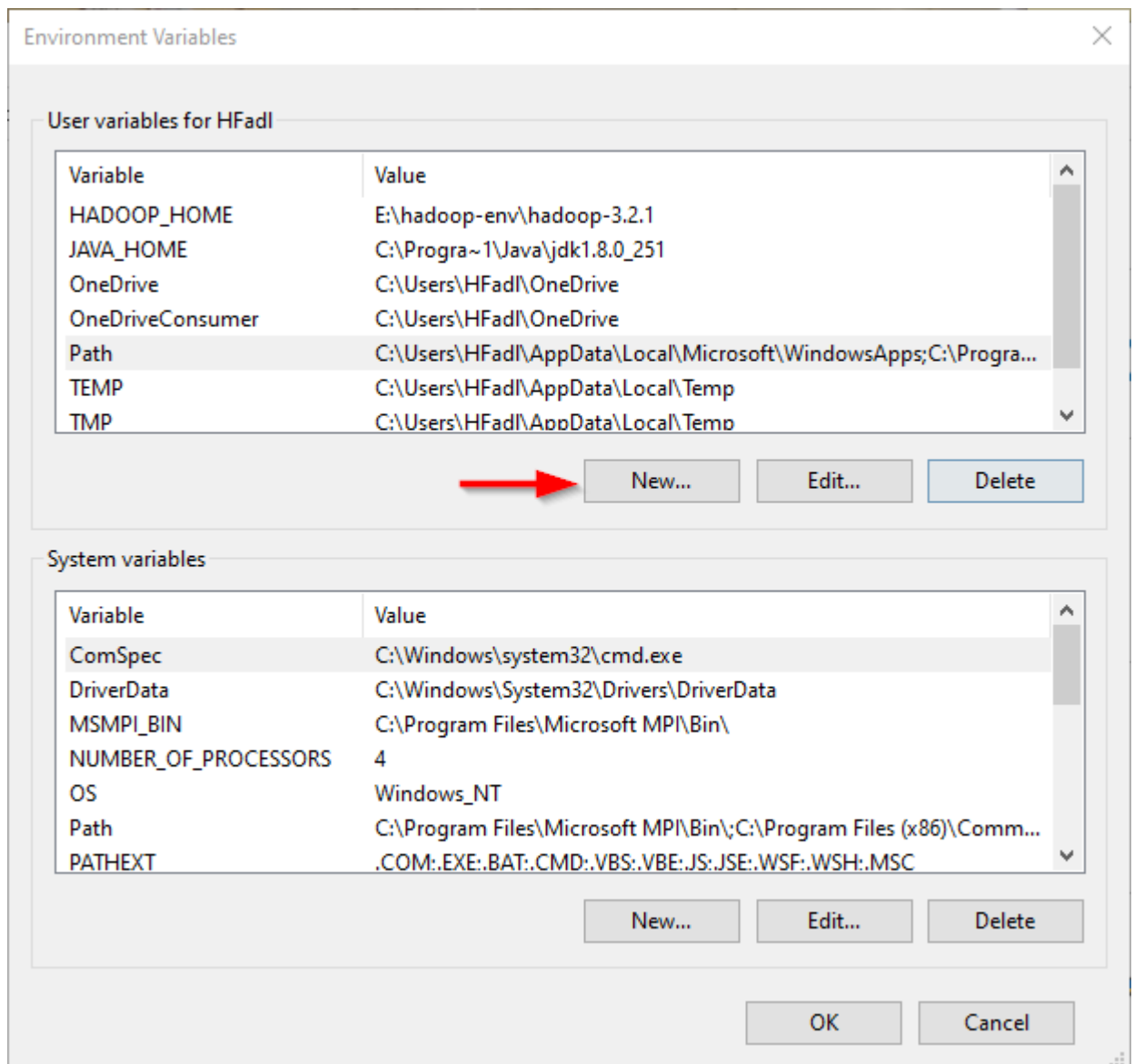


Figure 5 — Adding user variables

- PIG_HOME: “E:\hadoop-env\pig-0.17.0”

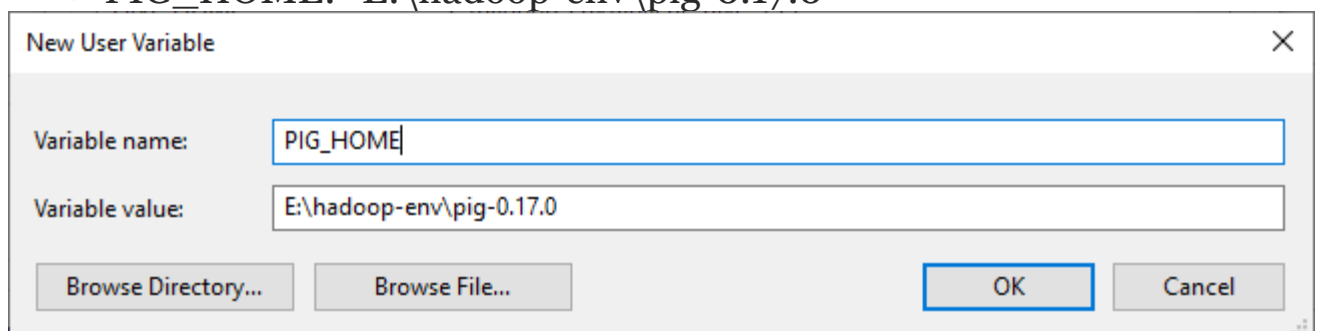


Figure 6 — Adding PIG_HOME variable

Now, we should edit the Path user variable to add the following paths:

- %PIG_HOME%\bin

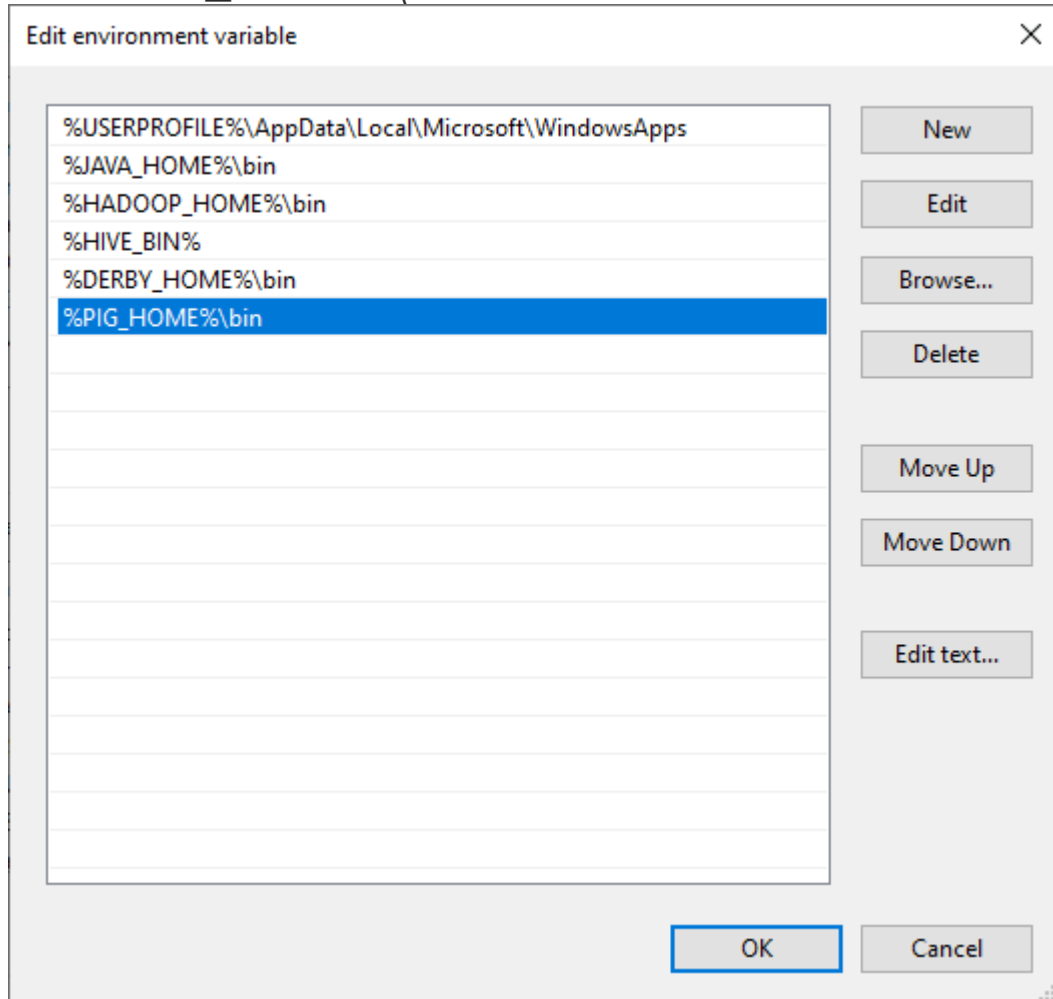


Figure 7 — Editing Path variable

4. Starting Apache Pig

After setting environment variables, let's try to run Apache Pig.

Note: *Hadoop Services must be running*

Open a command prompt as administrator, and execute the following command

```
pig -version
```

You will receive the following exception:

```
'E:\hadoop-env\hadoop-3.2.1\bin\hadoop-config.cmd' is not
recognized as an internal or external command,
operable program or batch file.
'-Xmx1000M' is not recognized as an internal or external
command,
operable program or batch file.
```

```
E:\>pig -version
'E:\hadoop-env\hadoop-3.2.1\bin\hadoop-config.cmd' is not recognized as an internal or external command,
operable program or batch file.
'-Xmx1000M' is not recognized as an internal or external command,
operable program or batch file.
```

Figure 8 — Pig exception

To fix this error, we should edit the `pig.cmd` file located in the “`pig-0.17.0\bin`” directory by changing the `HADOOP_BIN_PATH` value from “`%HADOOP_HOME%\bin`” to “`%HADOOP_HOME%\libexec`”.

Now, let's try to run the “`pig -version`” command again:

```
E:\>pig -version
Apache Pig version 0.17.0 (r1797386)
compiled Jun 02 2017, 15:41:58
```

Figure 9 — Pig installation validated

The simplest way to write PigLatin statements is using Grunt shell which is an interactive tool where we write a statement and get the desired output. There are two modes to involve Grunt Shell:

1. Local: All scripts are executed on a single machine without requiring Hadoop. (command: pig -x local)
2. MapReduce: Scripts are executed on a Hadoop cluster (command: pig -x MapReduce)

Since we have installed Apache Hadoop 3.2.1 which is not compatible with Pig 0.17.0, we will try to run Pig using local mode.

```
E:\>pig -x local
2020-05-05 03:22:24,894 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2020-05-05 03:22:24,895 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2020-05-05 03:22:25,246 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 12 2019
2020-05-05 03:22:25,246 [main] INFO org.apache.pig.Main - Logging error messages to: E:\hadoop-env\hadoop-log\hadoop-log.txt
2020-05-05 03:22:25,282 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file C:\Users\HFadl/.pigrc
2020-05-05 03:22:25,495 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker=mapred.job.tracker.address
2020-05-05 03:22:25,501 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to HDFS
2020-05-05 03:22:25,912 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum=4096
2020-05-05 03:22:25,960 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-1588641145
2020-05-05 03:22:25,962 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled=false
grunt>
```

Figure 10 — Starting Grunt Shell in local mode

EXPERIMENT 7

Basic Pig Commands

Let's take a look at some of the Basic Pig commands which are given below:-

1. Fs: This will list all the file in the HDFS

```
grunt> fs -ls
```

2. Clear: This will clear the interactive Grunt shell.

```
grunt> clear
```

3. History:

This command shows the commands executed so far.

```
grunt> history
```

4. Reading Data: Assuming the data resides in HDFS, and we need to read data to Pig.

```
grunt> college_students = LOAD  
'hdfs://localhost:9000/pig_data/college_data.txt'
```

```
USING PigStorage(',')
```

```
as ( id:int, firstname:chararray, lastname:chararray, phone:chararray,  
city:chararray );
```

PigStorage() is the function that loads and stores data as structured text files.

5. Storing Data: Store operator is used to storing the processed/loaded data.

```
grunt> STORE college_students INTO ' hdfs://localhost:9000/pig_Output/ '  
USING PigStorage (',');
```

Here, "/pig_Output/" is the directory where relation needs to be stored.

6. Dump Operator: This command is used to display the results on screen.

It usually helps in debugging.

```
grunt> Dump college_students;
```

7. Describe Operator: It helps the programmer to view the schema of the relation.

```
grunt> describe college_students;
```

8. Explain: This command helps to review the logical, physical and map-reduce execution plans.

```
grunt> explain college_students;
```

9. Illustrate operator: This gives step-by-step execution of statements in Pig Commands.

```
grunt> illustrate college_students;
```

EXPERIMENT 7

Pivot Tables

[Insert a Pivot Table](#) | [Drag fields](#) | [Sort](#) | [Filter](#) | [Change Summary Calculation](#) | [Two-dimensional Pivot Table](#)

Pivot tables are one of Excel's most powerful features. A pivot table allows you to extract the significance from a large, detailed data set.

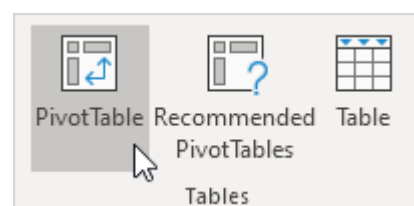
Our data set consists of 213 records and 6 fields. Order ID, Product, Category, Amount, Date and Country.

	A	B	C	D	E	F	G	H
1	Order ID	Product	Category	Amount	Date	Country		
2	1	Carrots	Vegetables	\$4,270	1/6/2016	United States		
3	2	Broccoli	Vegetables	\$8,239	1/7/2016	United Kingdom		
4	3	Banana	Fruit	\$617	1/8/2016	United States		
5	4	Banana	Fruit	\$8,384	1/10/2016	Canada		
6	5	Beans	Vegetables	\$2,626	1/10/2016	Germany		
7	6	Orange	Fruit	\$3,610	1/11/2016	United States		
8	7	Broccoli	Vegetables	\$9,062	1/11/2016	Australia		
9	8	Banana	Fruit	\$6,906	1/16/2016	New Zealand		
10	9	Apple	Fruit	\$2,417	1/16/2016	France		
11	10	Apple	Fruit	\$7,421	1/16/2016	Canada		

Insert a Pivot Table

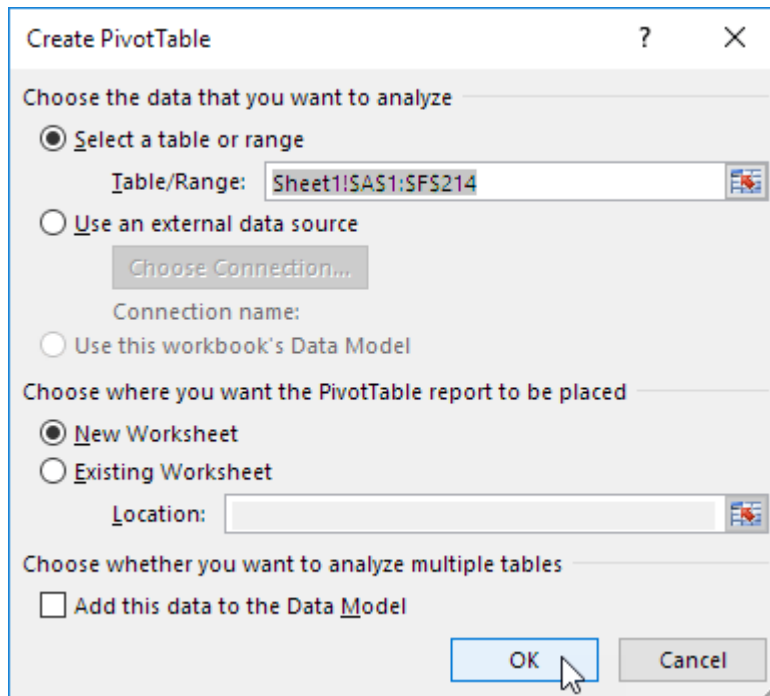
To insert a pivot table, execute the following steps.

1. Click any single cell inside the data set.
2. On the Insert tab, in the Tables group, click PivotTable.



The following dialog box appears. Excel automatically selects the data for you. The default location for a new pivot table is New Worksheet.

3. Click OK.



Drag fields

The PivotTable Fields pane appears. To get the total amount exported of each product, drag the following fields to the different areas.

1. Product field to the Rows area.
2. Amount field to the Values area.
3. Country field to the Filters area.

PivotTable Fields

Choose fields to add to report:

☐ Order ID
☒ **Product**
☐ Category
☒ **Amount**
☐ Date
☒ **Country**

Drag fields between areas below:

Filters

Country

Columns

Rows

Product

Values

Sum of Amou...

☐ Defer Layout Update

Update

Below you can find the pivot table. Bananas are our main export product. That's how easy pivot tables can be!

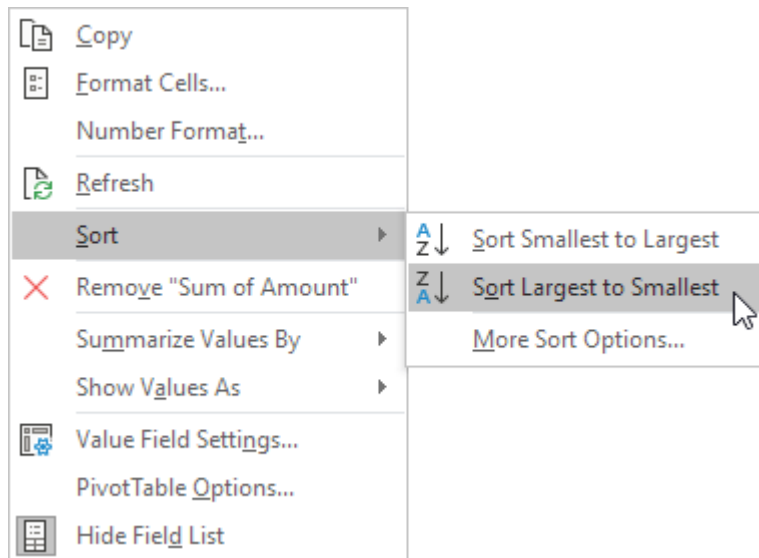
	A	B	C
1	Country	(All)	
2			
3	Row Labels	Sum of Amount	
4	Apple	191257	
5	Banana	340295	
6	Beans	57281	
7	Broccoli	142439	
8	Carrots	136945	
9	Mango	57079	
10	Orange	104438	
11	Grand Total	1029734	
12			

Sort

To get Banana at the top of the list, sort the pivot table.

1. Click any cell inside the Sum of Amount column.

2. Right click and click on Sort, Sort Largest to Smallest.



Result.

	A	B	C
1	Country	(All) ▼	
2			
3	Row Labels ▼	Sum of Amount	
4	Banana	340295	
5	Apple	191257	
6	Broccoli	142439	
7	Carrots	136945	
8	Orange	104438	
9	Beans	57281	
10	Mango	57079	
11	Grand Total	1029734	
12			

Filter

Because we added the Country field to the Filters area, we can filter this pivot table by Country.

For example, which products do we export the most to France?

1. Click the filter drop-down and select France.

Result. Apples are our main export product to France.

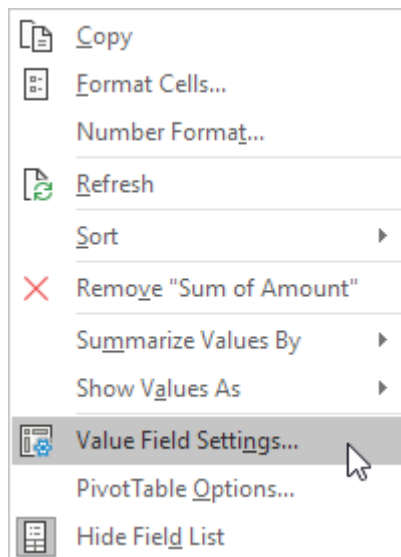
	A	B	C
1	Country	France	
2			
3	Row Labels	Sum of Amount	
4	Apple	80193	
5	Banana	36094	
6	Carrots	9104	
7	Mango	7388	
8	Broccoli	5341	
9	Orange	2256	
10	Beans	680	
11	Grand Total	141056	
12			

Note: you can use the standard filter (triangle next to Row Labels) to only show the amounts of specific products.

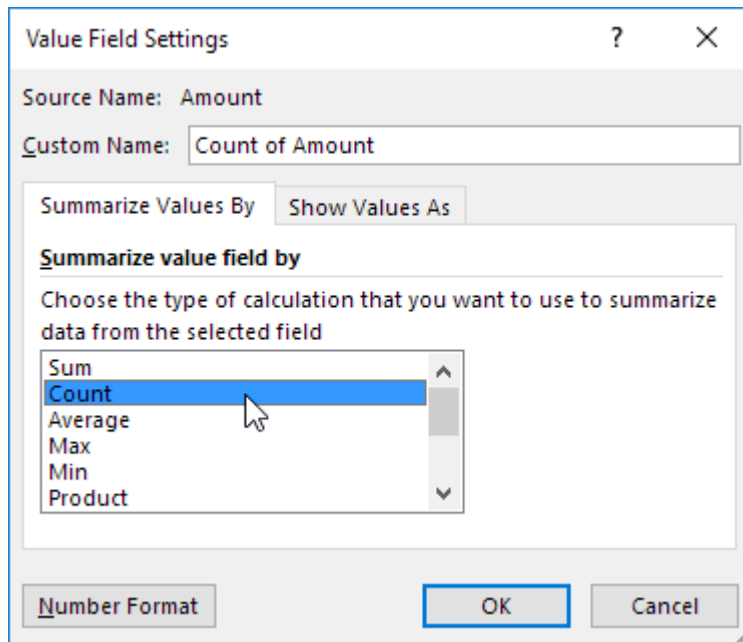
Change Summary Calculation

By default, Excel summarizes your data by either summing or counting the items. To change the type of calculation that you want to use, execute the following steps.

1. Click any cell inside the Sum of Amount column.
2. Right click and click on Value Field Settings.



3. Choose the type of calculation you want to use. For example, click Count.



4. Click OK.

Result. 16 out of the 28 orders to France were 'Apple' orders.

	A	B	C
1	Country	France	
2			
3	Row Labels	Count of Amount	
4	Apple	16	
5	Banana	7	
6	Carrots	1	
7	Mango	1	
8	Orange	1	
9	Beans	1	
10	Broccoli	1	
11	Grand Total	28	
12			

Two-dimensional Pivot Table

If you drag a field to the Rows area and Columns area, you can create a two-dimensional pivot table. First, [insert a pivot table](#). Next, to get the total amount exported to each country, of each product, drag the following fields to the different areas.

1. Country field to the Rows area.
2. Product field to the Columns area.
3. Amount field to the Values area.
4. Category field to the Filters area.

PivotTable Fields

Choose fields to add to report:

☐ Order ID
☒ Product
☒ Category
☒ Amount
☐ Date
☒ Country

Drag fields between areas below:

Filters

Category

Columns

Product

Rows

Country

Values

Sum of Amou...

☐ Defer Layout Update

Update

	A	B	C	D	E	F	G	H	I	J
1	Category	(All)								
2										
3	Sum of Amount	Column								
4	Row Labels	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange	Grand Total	
5	Australia	20634	52721	14433	17953	8106	9186	8680	131713	
6	Canada	24867	33775		12407		3767	19929	94745	
7	France	80193	36094	680	5341	9104	7388	2256	141056	
8	Germany	9082	39686	29905	37197	21636	8775	8887	155168	
9	New Zealand	10332	40050		4390			12010	66782	
10	United Kingdom	17534	42908	5100	38436	41815	5600	21744	173137	
11	United States	28615	95061	7163	26715	56284	22363	30932	267133	
12	Grand Total	191257	340295	57281	142439	136945	57079	104438	1029734	
13										

To easily compare these numbers, create a [pivot chart](#) and apply a filter. Maybe this is one step too far for you at this stage, but it shows you one of the many other powerful pivot table features Excel has to offer.

