

# Report Final project Summer School "Arqueology of the future"

Maria Coto-Sarmiento

November 29, 2016

## 1 Introduction

### 1.1 Dataset

We analyzed a dataset with amphorae from different places in order to understand the evolution of the amphorae made in different pottery workshops. Our dataset was collected from the project *Roman Amphorae: a digital recourse* within Archaeology data service at University of Southampton (fig.1)

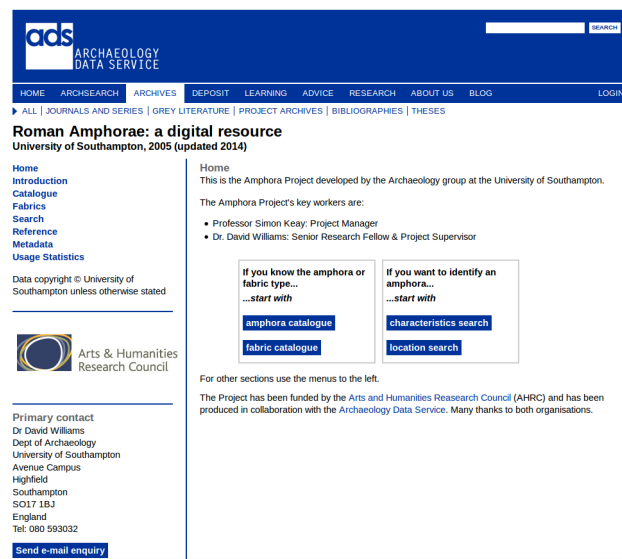


Figure 1: Webpage Roman Amphorae: a digital resource (Archaeology Data Service)

This database is a catalogue of around 375 amphorae of different types and divided in different places and chronology during the Roman Empire. It was taken some characteristics on the dataset as the different parts of the shape of amphorae.

Our dataset was divided in 24 columns based on the web page.

- **ID:** number of identification
- **Name:** type of amphorae
- **Rim type:** divided into triangular (the rim has two straight, angled sides, resembling two partial sides of a triangle), everted (the rim becomes gently wider towards the top), collar (the rim is noticeably thickened in the form of a collar around the neck of the amphora), beaded (the rim has a simple rounded lip), rounded (the rim is gently rounded), flaring (the rim flares out sharply, in a more pronounced manner than an everted rim) and pulley wheel (this distinctive rim resembles a pulley-wheel).
- **Shoulder type:** divided into rounded (the shoulder is noticeably present, usually supporting the handles, but there is no ridge), carinated (the shoulder displays distinct carination: a ridge where the shoulder meets the body of the amphora) and none/smooth (there is no shoulder: the body of the amphora is uninterrupted).
- **Handle profile:** divided into ear-shaped (these handles are similar to the 'curved' variety, but are more reminiscent of the shape of a human ear), bowed (the handles form a broad, curving sweep away from the body and neck of the amphora. They are generally longer than the 'curved' handles), curved (the handles gently curve from the neck to shoulder), peaked (the distinctive profile of these handles rises to a peak, often above the rim of the amphora. They are especially characteristic of the Rhodian type), ring (These handles are generally smaller than the 'curved' handles, forming a small semi-circular profile), short vertical (these handles travel upwards vertically from the shoulder, but only a short distance before turning inwards to the neck), arched (the handles form high arches, without coming to a point), long vertical (these handles generally appear on long-necked amphorae, attaching near the top of the neck progressing vertically downwards to the shoulder).
- **Handle section:** divided into ovoid/elliptical (the handle appears to be ovoid or elliptical in section), ridged (the handle has one or more ridges running down it), grooved (the handle is circular, or nearly circular, in section), round (the handle is circular, or nearly circular, in section), bifid (this is a distinctive feature of the Dressel 2-4 type. The handle is formed into two rods, appearing as a figure-8 in section).
- **Neck type:** divided into short/narrow (the neck is disproportionately short and/or narrow relative to the size of the amphora), cylindrical (the neck is cylinder-shaped), conical (the neck is cone-shaped - it tapers upwards), hourglass (the neck takes the form of an hourglass, narrowing at its mid-point), none (there is no distinct neck: the body of the amphora progresses smoothly upwards to the rim) and broad (the neck tends to be wide)
- **Body type:** divided into cylindrical (the body of the amphora is cylinder-shaped, displaying little curvature), tapered (the body tapers downwards:

it is wider at the top), globular (the body is wide, round and bag-shaped or globular), ovoid (the body is ovoid or elliptical in profile), piriform (the body is pear-shaped: it is wider towards the bottom of the amphora), narrow (the body is disproportionately narrow compared with the height of the amphora)

- **Base type** divided into short hollow (the base is short and hollow), spike/-tapered (there is a solid spike at the base of the amphora), knobbed (there is small knob at the base of the amphora), ringed (the base has a foot ring), button (there is a noticeable button-shape at the base of the amphora), flat (the base has been flattened so the amphora will stand unsupported), long hollow (the base and foot are long and hollow), pointed (the base comes to a point), rounded basal point (the base extends into a small, pronounced point not as long as a spike)
- **Capacity**: capacity of the amphorae
- **Height min**: minimum height
- **Height max**: maximum height
- **Width min**: minimum width
- **Rim diameter min**: minimum rim diameter
- **Rim diameter max**: maximum rim diameter
- **Manufacture**: workshops where amphorae come from

## 2 Objectives

The aim of this project can be divided in different objectives:

- A simple exploration of our dataset. We explore our dataset to seek the kind of information that we are going to use. So we will need to fit our database to each exercises.
- In particular, we want to understand the differences among workshops from different places. We can detect measurable differences in the amphorae production over time as the result of different pattern of production.
- The exercises will be different adapted to each section. For instance, we use a different database on MySQL and R commander than on QGIS. On the one hand, specifically on R program, we wanted to focus this exercise in *Dressel* production. For that, we wanted to use amphorae measurements to analyze the different production. On the other hand, we created a new amphorae workshop dataset from Andalusia (south Spain) adapted to the QGIS exercise.

## 3 MySQL

We used phpMyAdmin, a free software tool to handle the administration of MySQL with Linux and perform our data analysis (fig.2)

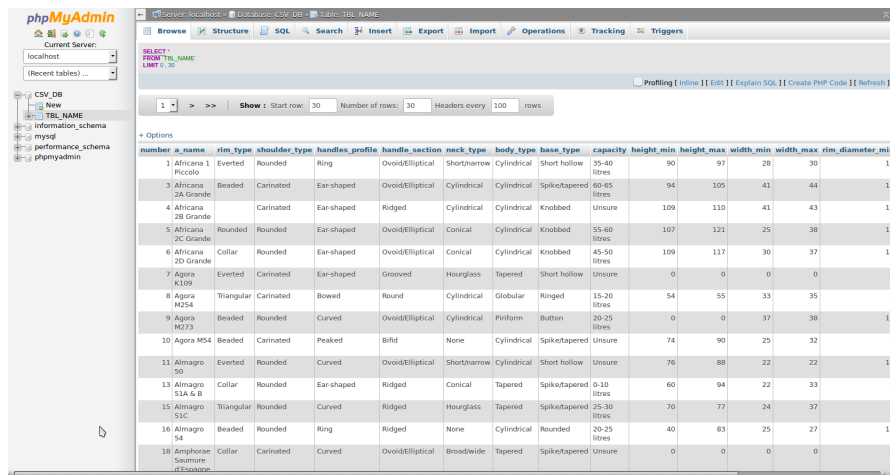


Figure 2: phpMyAdmin program with the database

## 3.1 MySQL exercises

We perform some exercises with mySQL and query the dataset.

### 3.1.1 Exercise 1

We want to know how many rim shapes are "everted". For that, we used the following code

```
SELECT *
FROM 'TBL_NAME'
WHERE 'rim_type' = "everted"
LIMIT 0 , 30
```

We selected all from our dataset but we wanted specifically to select from the column rim type the types which are everted. We also specified a limit by 30 (fig.3)

<pre>SELECT * FROM 'TBL_NAME' WHERE 'rim_type' = "everted" LIMIT 0 , 30</pre>														
Profiling [ Inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]														
+ Options														
number	a_name	rim_type	shoulder_type	handles_profile	handle_section	neck_type	body_type	base_type	capacity	height_min	height_max	width_min	width_max	rim_diameter_min
1	Africana 1 Piccolo	Everted	Rounded	Ring	Ovoid/Elliptical	Shortnarrow	Cylindrical	Short hollow	35-40 litres	90	97	28	30	12
3	Africana 2A Grande	Beaded	Carinated	Ear-shaped	Ovoid/Elliptical	Cylindrical	Cylindrical	Spike/tapered	60-65 litres	94	105	41	44	13
4	Africana 2B Grande	Carinated	Carinated	Ear-shaped	Ridged	Cylindrical	Cylindrical	Knobbed	Unsure	109	110	41	43	12
5	Africana 2C Grande	Rounded	Rounded	Ear-shaped	Ovoid/Elliptical	Conical	Cylindrical	Knobbed	55-60 litres	107	121	25	38	11
6	Africana 2D Grande	Collar	Rounded	Ear-shaped	Ovoid/Elliptical	Conical	Cylindrical	Knobbed	45-50 litres	109	117	30	37	11
7	Agora K109	Everted	Carinated	Ear-shaped	Grooved	Hourglass	Tapered	Short hollow	Unsure	0	0	0	0	0
8	Agora M254	Triangular	Carinated	Bowed	Round	Cylindrical	Globular	Ringed	15-20 litres	54	55	33	35	9
9	Agora M273	Beaded	Rounded	Curved	Ovoid/Elliptical	Cylindrical	Piriform	Button	20-25 litres	0	0	37	38	11
10	Agora M54	Beaded	Carinated	Peaked	Bifid	None	Cylindrical	Spike/tapered	Unsure	74	90	25	32	0
11	Almagro 50	Everted	Rounded	Curved	Ovoid/Elliptical	Shortnarrow	Cylindrical	Short hollow	Unsure	76	88	22	22	14
13	Almagro 51A & B	Collar	Rounded	Ear-shaped	Ridged	Conical	Tapered	Spike/tapered	0-10 litres	60	94	22	33	9
15	Almagro 51C	Triangular	Rounded	Curved	Ridged	Hourglass	Tapered	Spike/tapered	25-30 litres	70	77	24	37	8
16	Almagro 54	Beaded	Rounded	Ring	Ridged	None	Cylindrical	Rounded	20-25 litres	40	83	25	27	10
18	Amphora Seaurum d'Esage	Collar	Carinated	Curved	Ovoid/Elliptical	Broadwide	Tapered	Spike/tapered	Unsure	0	0	0	0	0

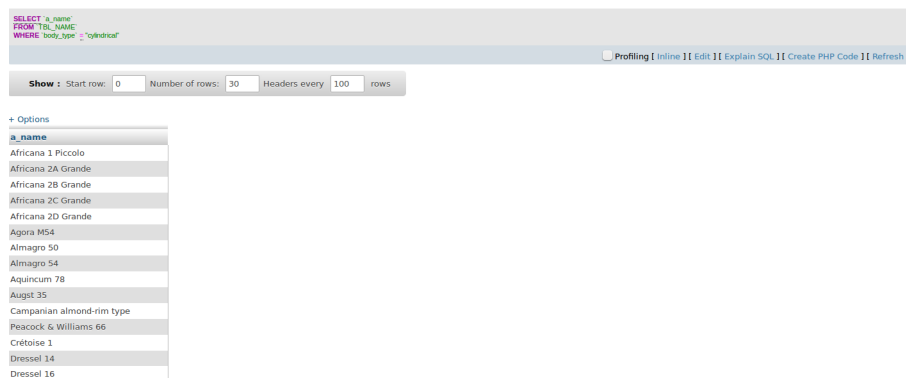
Figure 3: query selecting "everted"

### 3.1.2 Exercise 2

We want to know how many amphorae have a cylindrical body type. We use the following code

```
SELECT 'a_name'
FROM 'TBL_NAME'
WHERE 'body_type' = "cylindrical"
```

We selected the column *a\_name* from the general table on the dataset. We also selected from the body type column which amphorae are cylindrical (fig. 4).



The screenshot shows a SQL query interface with the following SQL code:

```
SELECT 'a_name'
FROM 'TBL_NAME'
WHERE 'body_type' = "cylindrical"
```

Below the query, there are controls for the query execution: "Show : Start row: 0 Number of rows: 30 Headers every 100 rows".

The results are displayed in a table with the following columns and rows:

a_name
Africana 1 Piccolo
Africana 2A Grande
Africana 2B Grande
Africana 2C Grande
Africana 2D Grande
Agora IM54
Almagro 50
Almagro 54
Aquincum 78
Augst 35
Campanian almond-rim type
Peacock & Williams 66
Crétolise 1
Dressel 14
Dressel 16

Figure 4: query selecting cylindrical column

### 3.1.3 Exercise 3

We want to know which amphorae have the largest diameter. We use the following code

```
SELECT 'a_name' , 'rim_diameter_max'
FROM 'TBL_NAME'
ORDER BY 'rim_diameter_max' DESC
LIMIT 100
```

We selected the columns and ORDER BY in order to know which diameter is the largest. We can choose between ASC (ascending order) or DESC (descending order) (fig. 5).

```
SELECT 'a_name', 'rim_diameter_max'
FROM 'TBL_NAME'
ORDER BY 'rim_diameter_max' DESC
LIMIT 100
```

Profiling [ Inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

+ Options

a_name	rim_diameter_max
Gauloise 4/5/6 similis in Verulamium White Ware	176
Gauloise Amphora in Verulamium White Slipped Ware	150
Gauloise 1 similis in Verulamium White Ware	130
Van der Werff 1	27
Beltrijn 2B	25
Beltrijn 2A	25
Dressel 9	24
Lomba do Canho 67	24
Dressel 10	22
Dressel 14	22
Dressel 8	22
Opait F9	22
Dressel 11	22

Figure 5: query selecting two columns ordered by maximum diameter

### 3.1.4 Exercise 4

We want to seek which amphorae have cylindrical body type and carinated shoulder type with a limit of 100. We also want to order by alphabetical name of amphorae.

```
SELECT 'a_name' , 'shoulder_type' , 'body_type'
FROM 'TBL_NAME'
WHERE 'shoulder_type' = "carinated"
AND 'body_type' = "cylindrical"
ORDER BY 'a_name'
LIMIT 0 , 100
```

We selected three columns from the original table and we specifically chose body type, shoulder type and *a\_name* ordered by names of amphorae (fig. 6).

```
SELECT 'a_name' , 'shoulder_type' , 'body_type'
FROM 'TBL_NAME'
WHERE 'shoulder_type' = "carinated"
AND 'body_type' = "cylindrical"
ORDER BY 'a_name'
LIMIT 0 , 30
```

Profiling [ Inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

+ Options

a_name	shoulder_type	body_type
Africana 2A Grande	Carinated	Cylindrical
Africana 2B Grande	Carinated	Cylindrical
Africana 3B	Carinated	Cylindrical
Agora G199	Carinated	Cylindrical
Agora M54	Carinated	Cylindrical
Dressel 1	Carinated	Cylindrical
Dressel 16	Carinated	Cylindrical
Dressel 1B similis - Lyon	Carinated	Cylindrical
Dressel 2-4 African	Carinated	Cylindrical
Dressel 2-4 Almadava	Carinated	Cylindrical
Dressel 2-4 Baetican	Carinated	Cylindrical
Dressel 2-4 Catalan	Carinated	Cylindrical
Dressel 2-4 Cilician	Carinated	Cylindrical
Dressel 2-4 Cos	Carinated	Cylindrical
Dressel 2-4 Egyptian	Carinated	Cylindrical
Dressel 2-4 Italian	Carinated	Cylindrical

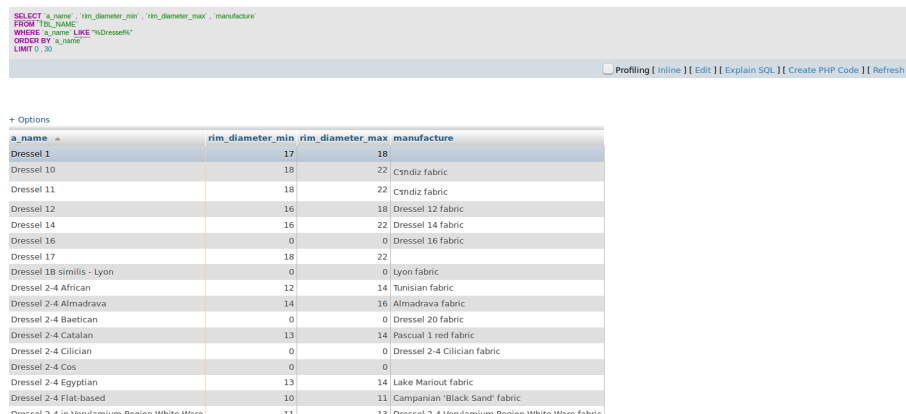
Figure 6: query selecting shoulder type, body type and type of amphorae

## 3.2 Querying the data

For our database, we want to select the Dressel types and its maximum and minimum diameters. We are going to analyze on R program the differences between diameters and whether they are related with the workshops of each type of amphorae.

```
SELECT 'a_name' , 'rim_diameter_min' , 'rim_diameter_max' , 'manufacture'
FROM 'TBL_NAME'
WHERE 'a_name' LIKE "%Dressel%"
ORDER BY 'a_name' ASC
LIMIT 0 , 100
```

To do that, we select the type of amphorae, maximum and minimum diameter and manufacture columns from the general table. We only want to select Dressel types so we use LIKE \$Dressel\$ to choose only the Dressel types in the database. The database will be ordered by names with the command ORDER BY and with ascending order. The limit must be until 100 to select all the amphorae as possible (fig. 7)



The screenshot shows a database query interface. At the top, the SQL query is entered: `SELECT 'a_name' , 'rim_diameter_min' , 'rim_diameter_max' , 'manufacture' FROM 'TBL_NAME' WHERE 'a_name' LIKE "%Dressel%" ORDER BY 'a_name' ASC LIMIT 0 , 100`. Below the query, there are links for `Profiling`, `Inline`, `Edit`, `Explain SQL`, `Create PHP Code`, and `Refresh`. The results are displayed in a table with the following columns: `a_name`, `rim_diameter_min`, `rim_diameter_max`, and `manufacture`. The table contains 20 rows of data, including Dressel types and their associated diameters and manufacture details.

a_name	rim_diameter_min	rim_diameter_max	manufacture
Dressel 1	17	18	
Dressel 10	18	22	Csrdiz fabric
Dressel 11	18	22	Csrdiz fabric
Dressel 12	16	18	Dressel 12 fabric
Dressel 14	16	22	Dressel 14 fabric
Dressel 16	0	0	Dressel 16 fabric
Dressel 17	18	22	
Dressel 1B similis - Lyon	0	0	Lyon fabric
Dressel 2-4 African	12	14	Tunisian fabric
Dressel 2-4 Almadra	14	16	Almadra fabric
Dressel 2-4 Baetican	0	0	Dressel 20 fabric
Dressel 2-4 Catalan	13	14	Pascual 1 red fabric
Dressel 2-4 Cilician	0	0	Dressel 2-4 Cilician fabric
Dressel 2-4 Cos	0	0	
Dressel 2-4 Egyptian	13	14	Lake Mariout fabric
Dressel 2-4 Flat-based	10	11	Campanian 'Black Sand' fabric

Figure 7: Database selected with Dressel amphorae

## 4 R programming language exercises

We explored the database produced by MySQL with R program. Thus, we have used R programming language in the command line in order to analyse the data. However, R Studio was not used in any exercise.

### 4.1 Calculate the mean with R program

We want to calculate the mean of maximum and minimum diameter of the amphorae to know the differences of the diameters.

We use the following code with R to calculate the mean maximum diameter

```
project= project$rim_diameter_max  
mean (project)  
[1] 13.1
```

We did the same but we want to know the mean minimum diameter using the same code

```
project= project$rim_diameter_min  
mean (project)  
[1] 10.95
```

We also wanted to calculate the frequency of maximum diameter in our database to know which measurement of diameter is more frequent in our database.

We perform with the following code

```
hist(project$rim_diameter_max, border= "white", col= "black",  
main= "Frecuency of maximum diameter")
```

We created a histogram with the frequency of the maximum diameter. We performed the histogram with black color and white borders. We added a title with the command "main" (fig. 8).

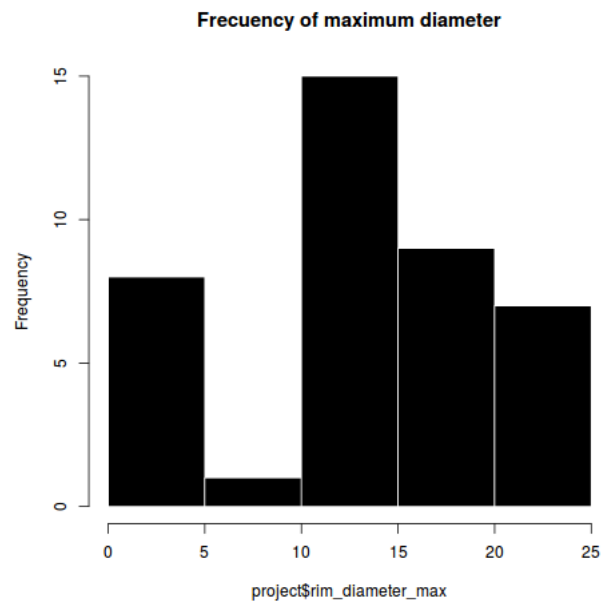


Figure 8: Histogram with the frequency of maximum diameter



Likewise, the same code was also used to calculate the frequency of minimum diameter as well.

```
boxplot(project$rim_diameter_min, main= "Frecuency of minimum diameter")
```

We used the boxplot graphic to visualize the frequency of minimum diameter (fig. 9)

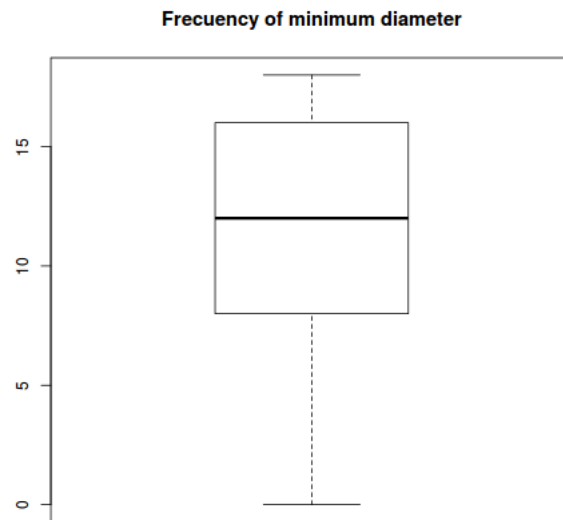


Figure 9: Boxplot with the frequency of minimum diameter

## 4.2 Exercises using different graphing packages/libraries

### 4.2.1 qplot

We used the graphing package `qplot` to plot the results of diameter with the pottery workshops to know if there are differences among them (fig. 10). We also performed the color scale to differentiate the pottery workshops

```
qplot(x= rim_diameter_max, y= rim_diameter_min, data= project,  
main= "Frecuency", color= manufacture)
```

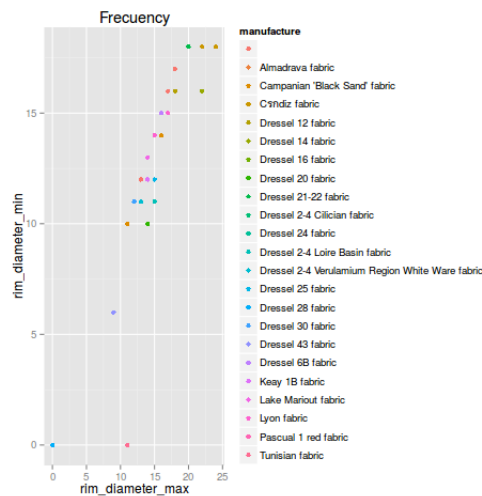


Figure 10: Using `qplot` graphing package

### 4.2.2 ggplot2

We wanted to see the diameter differences among pottery workshops ("manufacture") and type of amphorae. We had to install the package `ggplot2` before performing the data to use it as graphing library

```
install.packages ("ggplot2")  
library (ggplot2)
```

Then, we use the following code:

We are going to select different types of pottery workshops such as Cádiz fabric, Tunisian fabric, Dressel 20 fabric and Dressel 24 fabric because we want to know the differences in the pottery production

```
myData= subset(project, manufacture %in% c("Cádiz fabric",  
"Tunisian fabric", "Dressel 20 fabric", "Dressel 24 fabric"))
```

We selected from our database the type of amphorae and the pottery workshops ("manufacture") and we plot the results

```
ggplot(myData, aes(x=rim_diameter_max, y=rim_diameter_min, colour= manufacture))
+ geom_point() + facet_wrap(~a_name)
```

We divided the different amphorae types to see the differences among diameter related with the pottery workshops (manufacture in the picture fig. 11)

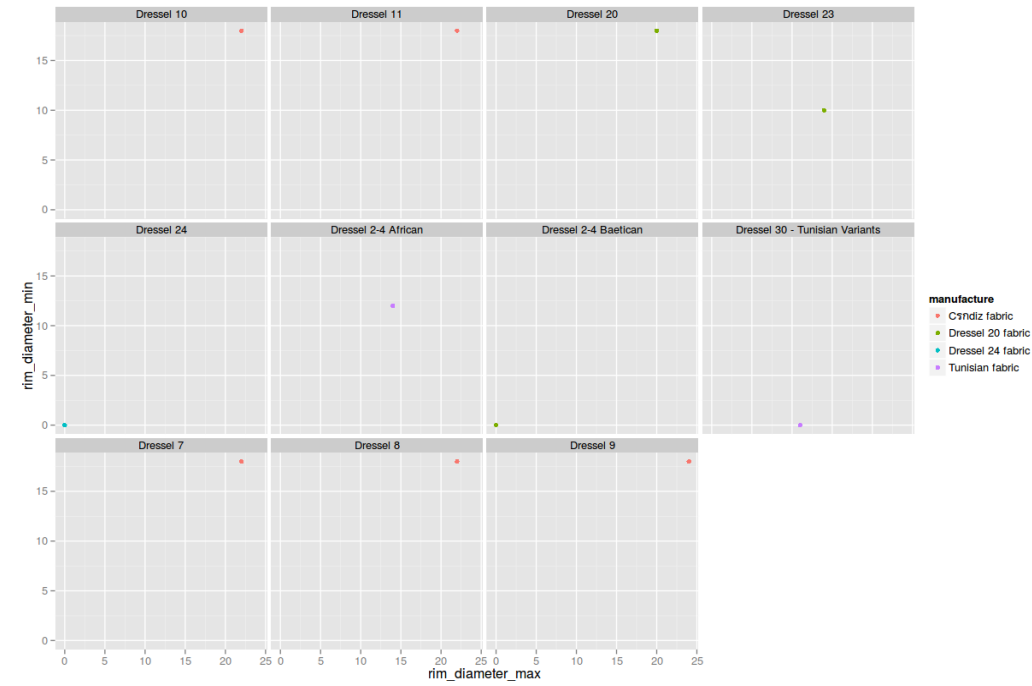


Figure 11: Using ggplot graphics

## 5 Mapping the data. Ontological section

### 5.1 CIDOC-CRM exercises

We want to develop the ontological process based on the pottery production. We created the following ontological entities from CIDOC-CRM, a semantic framework used in cultural heritage, to define our data (fig. 12). Then we represented our result in a graphic with the schema mapping.

- **E.55 Type** (because we have represented a pottery database)
- **E.57 Material** (represented by amphorae)
- **P.108 was produced by** (represented by potters)
- **E.12 Production** (pottery workshops)
- **P.31 Has modified by** (made and modified by potters)
- **P.14 Carry out by** (represented by potters and workshops)

- **E.24 Physical made man things** (represented by potters)
- **E.39 Actor** (represented by potters who make pottery)
- **E.53 Place** (represented by the production site)
- **P.53 has former or current location** (production site)

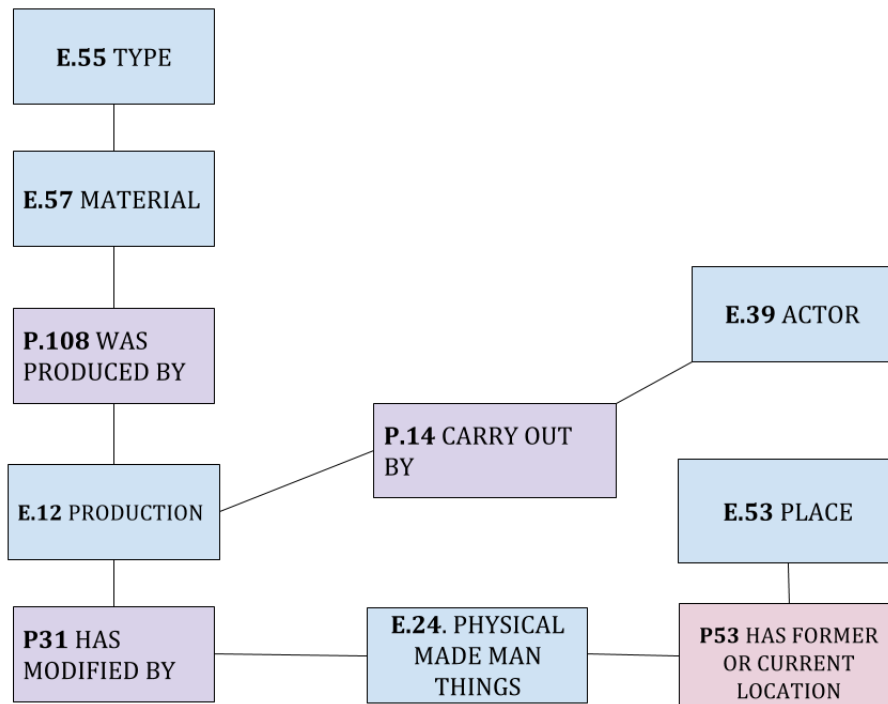


Figure 12: Mapping representing data with CIDOC-CRM

## 6 Licenses and data

We have selected for our data the licence created by Creative Common. In our case, we believe in the free distribution of open data so we chose the licence CC BY-SA. This kind of license allows to modify, develop and use the data, even with commercial purposes. This license also allows to use new creations under the identical terms and indicate if changes were made in our data (fig. 13).

We think that it could be a useful licence for our data because it allows to share our data for free.

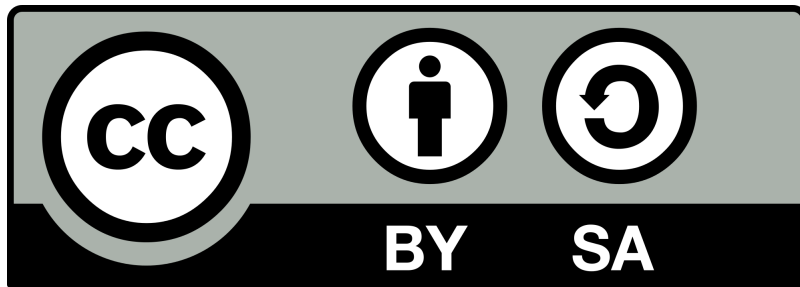


Figure 13: Type of license CC BY-SA

## 7 Analyze data on QGIS

### 7.1 Performing the data with Open Refine

We created a new dataset to transform with Open Refine and geolocalize the results with QGIS. We used the data from the pottery workshops of *Baetica* (currently Andalusia), ancient province within Roman Empire. In particular, we collected in the database the workshops of amphorae of olive oil. This commerce was particularly important due to the high demand of olive oil in the Roman Empire. As result, a huge network of pottery workshops was created to satisfy this demand.

As a first step, we perform our data with Open Refine:

- We open our data on Open Refine server
- We select the column where we want to edit. In the column, we select Edit Column and the option Add column by fetching URLs for the geolocalization
- We add in expression option the following code (fig. 14)

```
"http://maps.google.com/maps/api/geocode/json?sensor=false&address=" + escape(value, "url")
```

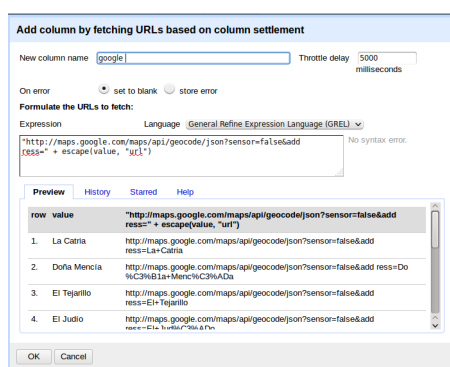


Figure 14: Add geolocalization on Open Refine

- When the file finishes loading, we select Edit cells and Transform. We write the next code to transform in latitude and longitude (fig. 15)

```
with(value.parseJson().results[0].geometry.location, pair, pair.lat +",  
" + pair.lng)
```

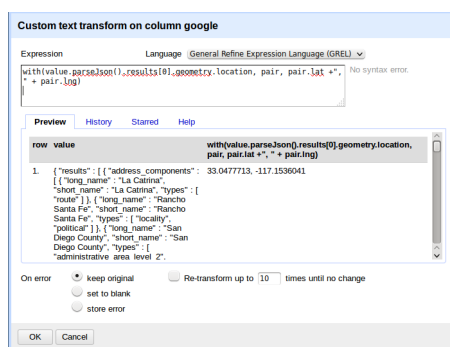


Figure 15: Transform the data in latitude and longitude

- We select edit column and the option Split into several column to separate into two different columns (fig. 16)
- We export the results as csv file

▼ All	▼ settlement	▼ google 1	▼ google 2
☆ ↗	1. La Catria	33.0477713	-117.1536041
☆ ↗	2. Doña Mencía	37.5544848	-4.3577976
☆ ↗	3. El Tejarillo	-34.3975996	-59.0661438
☆ ↗	4. El Judío	20.1340949	-98.82180509999999
☆ ↗	5. Las Ánimas	38.0666735	-103.222708
☆ ↗	6. El Tesoro	-34.9008819	-54.8703715
☆ ↗	7. Almodóvar del Río	37.8131513	-5.014567
☆ ↗	8. Azanaque	37.6107212	-5.6101709
☆ ↗	9. huerta del río	22.1272004	-101.0333942
☆ ↗	10. Peña de la Sal	43.323041	-4.1939408
☆ ↗	11. El Mohino	37.7329532	-5.2215292000000001
☆ ↗	12. Castillejo	40.6830576	-5.6548247
☆ ↗	13. Villaseca	41.31609539999999	-3.8261115
☆ ↗	14. Cerro de los Pesebres	19.6007194	-99.2317882
☆ ↗	15. Casa del Encinarejo	38.5732358	-4.0083161

Figure 16: Transform the data in latitude and longitude

## 7.2 QGIS

We used the database performed by Open Refine to produce a spatial visualization on QGIS. We had to modify our dataset because some pottery workshops did not detect real coordinates. These pottery workshops were deleted and we only included the pottery workshops correctly geolocalized. Ultimately, our data were performed with QGIS version 2.0.1 Dufour on Linux Mint.

We follow these steps below:

- We choose in Layer option Add Delimiter Text Layer to attach the csv file. You also can do it in the Toolbars on the left (fig. 17).

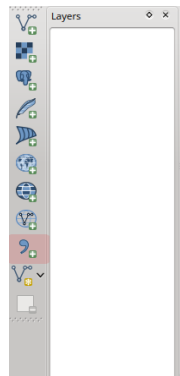


Figure 17: Option (in red) to attach the document file

- We attach the file that we want to use it and the format. In geometrical definition option we have to be sure for choosing the accurate latitude and longitude (fig. 18).
- We need to map our result. So we have to select Manage in the plugins option and install plugins and "get more" to install Open Layers. Open Layers is a plugin library to show and manipulate maps from different

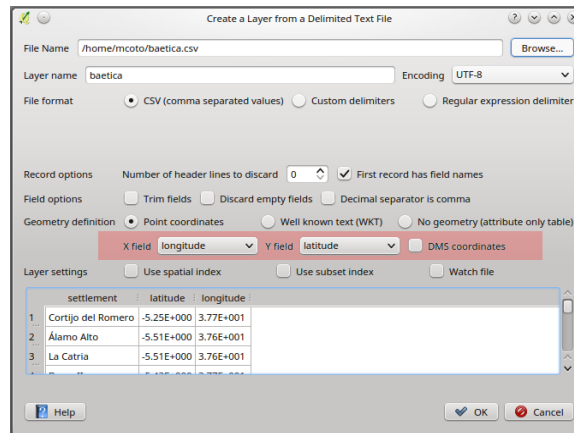


Figure 18: Option of geometry definition and coordinates

application on internet. On Linux, this API has to be installed through Python. So we have to install first OpenLayers plugin on Python (fig. 19).

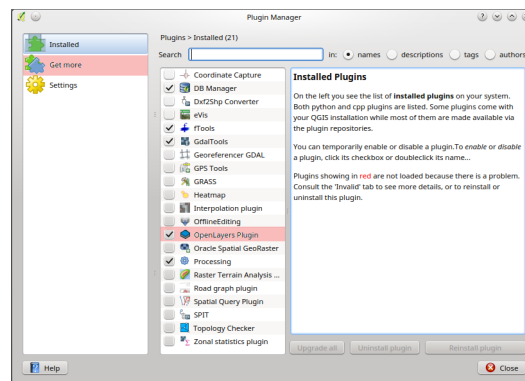


Figure 19: Install and select OpenLayers Plugin

- Once installed, we choose the most accurate map server to geolocalize our results. In our case, we chose Open Street Maps Layer.



- We map the database (fig. 20).

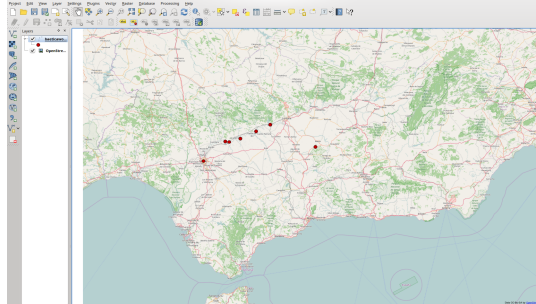


Figure 20: Mapping of the database

### 7.3 Discussion and Conclusion

We wanted to do a simple exploration of data so our results are not finished yet. In the case of mySQL exercises, we wanted to explore the data from our dataset.

We practised some codes in order to get data. We finally created a new dataset with dressel type to study the evolution of the production. The data obtained with R program allowed to identify measurable differences in the pottery workshops. For instance, the dressel types which were made in the same pottery workshops such as Dressel 7, Dressel 8, Dressel 9, Dressel 10, Dressel 11 shared the same measures in the diameter than the rest. By contrast, the types Dressel 20 and Dressel 23 did not share the same measures in spite of belonging to the same pottery workshops.

Otherwise, we created a new dataset to be performed with QGIS. Firstly, we modified our data with Open Refine to add the geographical coordinates. However, we had to delete some bugs from the pottery workshops because the coordinates were wrong. The results showed a homogeneous dispersion of the pottery workshops. This dispersion concurred with the necessity to build pottery workshops next to rivers in order to pick up clay from the rivers and make pottery artifacts.

We have identified new useful features from our database however this results were done as a simple exploration of data. Anyway, here we have showed a simple baseline for the exploration of the production processes. In a future, we expect to develop this study more deeply.

## 8 Apendix

We attached the files that we used to analyse the data. Here,