Welcome to Software Carpentry Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.  You can add your name in the top right corner, as well as changing the color of your typing.

Use of this service is restricted to members of the Software Carpentry and Data Carpentry community; this is not for general purpose use (for that, try etherpad.wikimedia.org).

Users are expected to follow our code of conduct: <http://software-carpentry.org/conduct.html>

All content is publicly available under the Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>

\*\*\*\*\*\*\*\*\*\*\*\*\*

**Data Carpentry @ UW Madison**

August 22-23, 2017

Location: 3rd Floor teaching lab, Wisconsin Institutes for Discovery (<http://map.wisc.edu/s/viuw59ds> )

**Useful Links**

Workshop website: <https://uw-madison-aci.github.io/2017-08-22-uwmadison-dc/>

This etherpad: <http://pad.software-carpentry.org/2017-08-22-uwmadison-dc>

Data for whole workshop: <http://kbroman.org/datacarp/>

Lesson material we'll be using: <http://www.datacarpentry.org/lessons/#ecology-workshop>

Tweet us at #dcuwm

**Day 1 schedule:**

    Install help 8:30 - 9am

    Intros 9 - 9:15am

    Spreadsheets 9:15am - 10:30am

    OpenRefine 10:45am - 12pm

    SQL 1pm - 4pm

**Install issues**

**Intros**

* Name // Discipline // Research/work in 3 words
* Ed Boswell//Soil Science//Winter Soil Biophysics
* Richard Barker // Botany // Plant Transcription Orbit
* Yaqi Zhang // Mechanical Engineering // 3D Printing Simulation
* Elizabeth Larson//Biochemistry//Transcription in Drosophila
* Samantha Fye//Pediatric Immunology//Cytometry Maturation Markers
* Isabel Rojas// Forest and Wildlife Ecology//Forest landscape connectivity
* Sohyun Kang // Educational Policy Studies // Stratification in highered
* Shu Wang // Mechanical Engineering// Molecular Dynamics
* Dave Bloom // Science & Engineering Librarian // Assisting researchers . . . research
* Angela Breard-Chen//Undergraduate Economics Student//
* Sam Hartke // Water Resource Engineering // Satellite precipitation errors
* Mingwei Huang // Biomolecular Chemistry // Human fungal pathogen
* K. Draeger // Forest Fungal Communities
* Jiarui Li//Undergraduate student computer science
* Julie Cheung//Biochemistry//Plant Membrane Trafficking
* Rebecca Hutcheson // Cellular and Molecular Pathology // Evolution of Burkitt Lymphoma
* Jaimie West // Soil Science and Agronomy //  soil & agriculture connection
* Devon Bulman // Environmental Chemistry and Technology // Chlorine Photolysis Kinetics
* Megan Fleming// Psychological Science// Addiction and Stress
* Huibin Ke//Materials modelling//nuclear reactor life extension
* Jonathan Lombardino// Botany //Plant Radiation Response
* Sarah Stevens // Microbiology/genomics // Bacteria Lakes Populations
* Brian Yandell // Systems genetics & data science // model phenotype relationships
* Dan Dryer// Soil Science Undergraduate Student//Soil aggregation research
* Christina Koch // CS // computing enabling research
* Alex Bajcz // Soil Sciences Postdoc // Also plant ecologist and simulation modeler in R
* Trisha Adamus // Medical Library // research data and analytics

What are you hoping to learn?

**Data Organization in Spreadsheets Lesson**

<https://pollev.com/trishaadamus254> <- link to poll for spreadsheets

Spreadsheet to download: <https://ndownloader.figshare.com/files/2252083>

**Open Refine Lesson**

 To open Open Refine [http://127.0.01:3333](http://127.0.0.1:3333/)

Spreadsheet to download:

<https://www.dropbox.com/s/kbb4k00eanm19lg/Portal_rodents_19772002_scinameUUIDs.csv?dl=0>

Data for Exercise:

<https://data.cityofnewyork.us/social-services/sandyrelated/fs5z-tpv4>

Select Export ->.csv

Great slides on naming things: <http://www2.stat.duke.edu/~rcs46/lectures_2015/01-markdown-git/slides/naming-slides/naming-slides.pdf>

Want more examples of OpenRefine? Do you like the Grateful Dead? Visit <https://github.com/scottythered/gratefuldata/wiki>

For background, read <https://github.com/scottythered/gratefuldata/wiki/About-Data-Cleaning-(and-the-Dead)>

If you want to learn even more about the GD band, see <https://www.amazon.com/Long-Strange-Trip/dp/B072HHSYTQ/>

**SQL**

Download the portal\_mammals.sqlite file and .csv files from <http://kbroman.org/datacarp/>

SQL commands:

SELECT year, month, day FROM surveys

* - shows year, month, and day columns from the surveys table

SELECT DISTINCT year, month, day FROM surveys

* - shows only the distinct entrys from the columns year, month, day from the surveys table

SELECT \* FROM species ORDER BY taxa DESC

* - shows all columns from speices table and orders them in decending order by their taxa column

SELECT \* FROM surveys WHERE year >= 2000 AND species\_id = 'DM'

* - shows all columns but only the rows of surveys where the year is greater than or equal to 2000 and species id is DM
* - DM needs quotation marks since it is text

SELECT \* FROM surveys WHERE species\_id IN ('DM', 'DO','DS')

* - Selects rows where species ID is either 'DM', 'DO', or 'DS'

SELECT month, day, year, species\_id, weight/1000.0

FROM surveys

WHERE year = 1999

ORDER BY species\_id

SELECT COUNT(\*) FROM surveys

* - counts number of rows in surveys table

SELECT COUNT(\*), SUM(weight) FROM surveys

* - counts number of rows in surveys table and sums all the weights

SELECT MAX(weight), MIN(weight), AVG(weight) FROM surveys

WHERE species\_id IN ('DO','DM','DS') AND year = 1999

SELECT year, MAX(weight), MIN(weight), AVG(weight) FROM surveys

WHERE species\_id IN ('DO','DM','DS') AND year = 1999 GROUP BY year

SELECT year, speices\_id, COUNT(\*) FROM surveys GROUP BY year, species\_id

SELECT species\_id, COUNT(species\_id) AS num\_species FROM surveys

GROUP BY species\_id HAVING num\_species > 10 ORDER BY num\_species DESC

Other SQL notes:

    SQL is not case sensitive

    The SQL Manger (Firefox extention) does not save history - chosen because it works across operating systems

    You can split up the query into multipule lines to help organize them - Christina doesn't because then she would have to scroll for you to see the whole query

It is hard to find specific rules on names of columns. The following two links say avoid reserved words:

<https://mariadb.com/kb/en/sql-99/naming-rules/>

<http://mdc.jackbe.com/prestodocs/v3.8/raql/validNames.html> (RAQL seems to be a variation on SQL)

You can also use quotes around column names or use SQL Aliases (<https://www.w3schools.com/sql/sql_alias.asp)> if your data has column names such as "distinct".

For more information on SQL in general see [www.lynda.com/SQL-tutorials/SQL-Essential-Training/139988-2.html](http://www.lynda.com/SQL-tutorials/SQL-Essential-Training/139988-2.html) or [www.w3schools.com/sql](http://www.w3schools.com/sql)

SELECT sv.year, sv.month, sv.day, sv.weight, sv.species\_id, sp.genus

FROM surveys AS sv JOIN species AS sp ON sv.species\_id = sp.species\_id

Challenge

1.

SELECT COUNT(sv.plot\_id), sv.plot\_id

FROM surveys AS sv JOIN plots AS pl ON sv.plot\_id = pl.plot\_id

GROUP BY sv.plot\_id

2.

SELECT year, species\_id, sex, COUNT (species\_id) FROM surveys GROUP BY species\_id, year, sex

3. SELECT sv.species\_id, pl.plot\_type, COUNT(sv.species\_id) FROM surveys AS sv JOIN plots AS pl ON sv.plot\_id = pl.plot\_id GROUP BY pl.plot\_type, sv.species\_id

Notes for R:

R style guides:

Google's - <https://google.github.io/styleguide/Rguide.xml>

tidyverse's - <http://style.tidyverse.org/>

Jean Fan's - <http://jef.works/R-style-guide/>

#Reading into object surveys directly from internet

surveys <- read.csv(file = "<http://kbroman.org/datacarp/portal_data_joined.csv>")

OR

#Download the file to our computer, then read it into the survey's object

download.file(url = "<http://kbroman.org/datacarp/portal_data_joined.csv>", destfile = "raw\_data/portal\_data\_joined.csv")

surveys <- read.csv(file = "raw\_data/portal\_data\_joined.csv")

head(surveys) # first 6 rows of the dataframe

tail(surveys) # bottom 6 rows of the dataframe

dim(surveys) # shows you the number of rows and cols your dataframe has

rnow(surveys) # number of rows data frame

ncol(surveys) # number of cols in dataframe

names(surveys) # col names in dataframe

rownames(surveys) # row names in dataframe

str(surveys) # stucture of data frame

summary(suverys) # stats about your numerical data, and number of observations for factors

Indexing

surveys[1, 5] # 1st row, 5th column

surveys[5, 1] # 5th row, 1st column

surveys[1, ] # all cols for first row, leave col blank

surveys[ , 7] # all rows, 7th column

#can use col names

surveys$sex # returns the sex column

sex <- surveys$sex # assigns the sex column to new object/variable, this is a vector (only 1 dim, doesn't have both rows and cols)

vec1 <- c(5, 50, 500) # making a vector with 'combine' function

dplyr -

**SQL cmd                                              --> dplyr cmd  (action)**

SELECT                                                 --> select         (select columns)

WHERE                                                  --> filter           (filter rows)

SELECT + function/operation                --> mutate       (does operation/function on column)

GROUP BY                                            --> group\_by   (groups by some categorical data to then perform some operation/function on the groups)

aggregators such as sum, min, sd, etc --> summarize (perform aggregation functions on grouped data)

%>% (pipe) shortcuts

Win: ctrl-shift-m

Mac: cmd-shift-m

**Advanced Information on dplyr**angling Cheat Sheet: <https://www.rstudio.com/resources/cheatsheets/> (same sheet you have on paper)

* Databases using dplyr: <https://db.rstudio.com/dplyr/> (this shows how to use SQL via dplyr)

# Downloading the cleaned and reduced data set to plot with ggplot2

download.file(url = "<http://kbroman.org/datacarp/portal_data_reduced.csv>", destfile = "raw\_data/portal\_data\_reduced.csv")

surveys <- read.csv(file = "raw\_data/portal\_data\_reduced.csv")

Rcolors - <http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

**Rmarkdown section**

If you use other languages such as bash, python, stan or SQL, see the following page for information about how to use them with knitr/rmarkdown:

<http://rmarkdown.rstudio.com/authoring_knitr_engines.html>

Computational Biology, Ecology, Evolution group - <https://combee-uw-madison.github.io/>

Research Data Sevices <http://researchdata.wisc.edu/>

Help email:

swc-dc-help@lists.wisc.edu

Interested in becoming a Data Carpentry helper/instructor?

Email Christina - ckoch5@wisc.edu

Great listserv about campus resources can be found at the link below:

<http://aci.wisc.edu/>

Center for High Throughput Computing

chtc.cs.wisc.edu