

Welcome to Software Carpentry Etherpad for the May 1st workshop at the University of Connecticut

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.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try <https://etherpad.wikimedia.org/>).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

All content is publicly available under the Creative Commons Attribution License:

<https://creativecommons.org/licenses/by/4.0/>

We will use this Etherpad during the workshop for chatting, asking questions, taking notes collaboratively, and sharing URLs or bits of code.

Todo list for participants:

- Go to the workshop website: <https://carpentries-uconn.github.io/2020-05-01-UConn-online> (link in chat, too)
- Click the link under the Collaborative Notes section to get to this page
- Name yourself in this page in the top right corner where it says Enter your name
- Add your name, university, & operating system (try to match the helper's OS) under a breakout room.
- Open up RStudio. In the Console window (bottom left quarter) run the following command:
`install.packages(c("ggplot2", "gapminder", "cowplot", "plotly"))`
- Open a tab with <https://b.socrative.com/login/student/>, and join room: SWCUCONN
- Take the pre-workshop survey on the workshop website if you haven't already:
- Introduce yourselves in the chat (on the right), so we know who you are

Instructors:

- * James Mickley - Ecology and Evolutionary Biology (james.mickley@uconn.edu)
- * Dyanna Louyakis - Molecular and Cell Biology (artemis.louyakis@uconn.edu)
- * Timothy Moore - COR2E Statistical Consulting Services & UConn Carpentries (timothy.e.moore@uconn.edu)
- * Kendra Maas - COR2E MARS (kendra.maas@uconn.edu)
- * Jeremy Teitelbaum - Math (jeremy.teitelbaum@uconn.edu)

For participants - Choose your breakout rooms:

Breakout room Tim

Helper: **Timothy Moore** - Statistical Consulting Services & UConn Carpentries (Windows)

1. Dennis-UConn, Psychological Sciences, OSX
2. Nikola Vukovic (OSX)

Breakout room Jeremy

Helper: **Jeremy Teitelbaum** - Math (Linux & OSX)

1. Siliva - UConn Psycholgoical Sciences - OSX
2. Matt- UCSF-OSX
3. Oliver- UConn, Psychological Sciences, OSX

Breakout room Kendra & Megan

Helper: Kendra Maas MARS (Windows) & Megan Chiovaro - Psychological Sciences - PAC-E (OSX)

1. Leah - UConn- OSX
2. Rebecca - UMich - OSX

Breakout room Eliza

Helper: **Eliza Grames** - Ecology and Evolutionary Bio (Linux or Windows or OSX)

1. Olga Kepinska - UCSF/UConn (OSX)
2. Shaan Kamal (OSX)
3. Florence Bouhali UCSF (OSX)

Breakout room Michael

Helper: **Michael LaScaleia** - Ecology and Evolutionary Bio (Windows)

1. Natasza Marrouch, UConn (OSX)
2. Jieyin - UConn - Windows

Breakout room Jie

Helper: **Jie Chen**- Nursing (Linux or OSX)

1. Jocelyn Caballero (OSX)
2. Chloe Jones UConn (OSX)

Workshop Website: <https://carpentries-uconn.github.io/2020-05-01-UConn-online>

Socrative Login (for quizzes): <https://b.socrative.com/login/student/>

Room: SWCUCONN

Download gapminder_data.csv here (Click download button at top right, and choose Direct Download)

https://www.dropbox.com/s/xeheqo6iyysz2i6/gapminder_data.csv?dl=0

Follow along with Dropbox script:

<https://www.dropbox.com/s/qkb5jyp58z38g7e/ggplot.R?dl=0>

follow-up

- getting involved
- etherpad export
- resources

Beginning of Workshop

NOTES:

use etherpad for collaborative note taking

Socrative is a way to give you all a chance to test what you've learned so far.

In Zoom, you can raise your hand if you have a question. Kendra will also monitor the etherpad chat if you have questions there.

If you only have one screen, we suggest you put zoom and rstudio side by side and change zoom to either "fit to screen" or 150%

Check your R version and or package versions

```
>R.version()
```

```
>packageVersion("ggplot2")
```

```
# Creating a project will help you organize your analysis for yourself and enable you to share a project  
(code and data) with a collaborator+1
```

```
# we're going to create a 'data' and 'figures' folders. also create a new R Script and name it 'ggplot.R'  
### move the gapminder_data.csv into the 'data' folder
```

```
"#" is a comment in R, leave yourself and you collaborators lots of comments explaining what you are  
doing!
```

```
>?read.csv # bring up help on a specific function
```

```
# check your data when you read it in
```

- head() # see first 6 rows
- str() # look at the structure of the data-gives you more info on each variable
- rStudio also shows you very basic info about your data in the 'Environment' tab (default setup has Environment in the upper right panel). This shows you the size of the data-check that you have as many rows (obs.) and columns (variables) that you expect.

```
# ggplot Grammer of Graphics
```

```
### ggplot uses slightly different syntax as base R, this will take a bit to get used to. But is super powerful  
once you get it.+
```

```
### just like you can structure a sentence in many ways, you can structure a ggplot command in many  
ways. We're going to put the "noun", the data within ggplot() function. RStudio has really handy  
cheatsheets for some major packages like ggplot2, you can get to it in the Help menu.
```

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp))
```

```
# this gives an empty plot because you haven't told ggplot the "verb" or what you want ggplot to do with  
that data. geom are the main type of verb in ggplot
```

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp))+  
  geom_point()
```

```
# You can map more than x & y position, add color to your mapping
```

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp, color = continent))+  
  geom_point()
```

```
# maybe we can see the data better as lines rather than points. To do that we need to tell ggplot how to  
group the data.
```

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp, color = continent, group_by = country))+  
  geom_line()
```

```
#you can also put more than one geom (or layer) on a plot
```

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp, color = continent, group_by = country))+  
  geom_line(mapping = aes(color = continent)) +
```

```
geom_point(color = "blue")
```

**** You can think of ggplot as taking on layers:**

- the base layer is the geom
- you can add various layers to your plots using '+' and different **geom functions** (e.g., geom_line, geom_point)
- Help on geometry layers: <https://ggplot2.tidyverse.org/reference/#section-layer-geoms>
- Common geometry layers:
 - **geom_point()** # Scatterplot
 - **geom_jitter()** a special type of scatterplot, that adds some random noise to points so they don't plot exactly on top of each other
 - **geom_line()** # Line plot
 - **geom_barplot()** # Bar graph
 - **geom_boxplot()** # Boxplots
 - **geom_smooth()** # Trend lines
 - Lots of different kinds of smoothers or trendlines here. The default is loess, which is a wavy curved line
 - The straight line we're all used to is **method = "lm"** for linear model
 - **geom_histogram()** # Histogram
 - **geom_density()** # Smoothed histograms
- You can change aesthetics of specific layers of the plot, by adding 'aes' to the layer you want to customise

Hadley Wickham quote:

- *“In brief, the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinates system.”*

NOTE 'gg' in ggplot stands for grammar of graphics.

So far we've seen the noun and verb of our grammar. now we can add in the adjectives and adverbs.

- Scales change the coordinate system

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp))+  
  geom_point()+  
  scale_x_log10()
```

since ggplot is a **grammar** there is often more than one way to accomplish the graph that you want. You can specify mapping = aes(???) in the main ggplot() or in a specific geom_X() for example, if you want to color the points by continent and run a linear model for each continent you can do that in a few different ways.

```
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp))+  
  geom_point(aes(color = continent))+  
  scale_x_log10()+  
  geom_smooth(aes(group = continent), method = "lm")
```

the order of the geom control which is layer is on top

```
# you can add more than one mapping to a geom
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp))+
  geom_point(aes(color = continent, shape = continent), size = 2, alpha = 0.5)+
  scale_x_log10()+
  geom_smooth(aes(group = continent), method = "lm")

# Now to clean this figure up for publication. Control the axis labels and breaks, change the background
and guide lines, add nicer title and guide (legend)
ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(mapping = aes(shape = continent), size = 2) +
  scale_x_log10() +
  geom_smooth(method = "lm") +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 10)) +
  theme_minimal() +
  labs(title = "Effects of per-capita GDP", x = "GDP per Capita ($)", y = "Life Expectancy (yrs)", color
= "Continents", shape = "Continents")

# exporting your plots. Best practice is to not use the "Export" button because that isn't reproducible

ggsave(file = "figures/life_expectancy.png")
ggsave(file = "figures/life_expectancy.pdf")
ggsave(file = "figures/life_expectancy.pdf", width = 10, height = 6, dpi = 300)

# when you specify the width and height you are changing the ratio between the plot and text, you may
need to play with the values for width and height if your text is too big or small

# you can save plots to a variable then explicitly name that plot in the ggsave()
lifeExp_plot <- geom_point(mapping = aes(shape = continent), size = 2) +
  scale_x_log10() +
  geom_smooth(method = "lm") +
  scale_y_continuous(limits = c(0, 100), breaks = seq(0, 100, by = 10)) +
  theme_minimal() +
  labs(title = "Effects of per-capita GDP", x = "GDP per Capita ($)", y = "Life Expectancy (yrs)", color
= "Continents", shape = "Continents")

ggsave(lifeExp_plot, file = "figures/life_expectancy.pdf", width = 10, height = 6, dpi = 300)

# split your data into different panels within a larger plot by "facet"ing on data

ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp)) +
  facet_wrap(~ continent, ncol = 2, scales = "free") +
  geom_point(alpha = 0.5) +
  scale_x_log10() +
  geom_smooth(method = "lm")

# within facet_wrap you can set whether you want the coordinates to be the same for each of the smaller
plots or let them be "free" from the others. I use scales = "free" carefully because often I want to use
facets to be able to easily compare the different subsets of data.
```

Please give feedback for your instructors: <https://forms.gle/6JaT94UxFLqn4eo76>

Take the post-workshop survey for Software Carpentry: <https://carpentries.typeform.com/to/UgVdRQ?slug=2020-05-01-UConn-online>

Resources

- R Graph catalog: <http://shiny.stat.ubc.ca/r-graph-catalog/>
- GGPlot2 online help: <http://ggplot2.tidyverse.org>
- R Graph Cookbook: <http://www.cookbook-r.com/Graphs/>
- ggplot2 essentials: <http://www.sthda.com/english/wiki/ggplot2-essentials>
- Rstudio cheatsheets: <https://www.rstudio.com/resources/cheatsheets/>
- Cowplot: <https://cran.r-project.org/web/packages/cowplot/vignettes/introduction.html>

Themes

- Rather than using `theme_minimal()` or `theme_cowplot()` by itself, we can customize that, too
- <http://ggplot2.tidyverse.org/reference/ggtheme.html>
- <http://ggplot2.tidyverse.org/reference/theme.html>

Most of what we covered is here:

<http://swcarpentry.github.io/r-novice-gapminder/08-plot-ggplot2/index.html>

Full Reproducible Research lesson: <http://swcarpentry.github.io/r-novice-gapminder/>

GGplotly

-bookdown

<https://plotly-r.com/index.html>

Becoming a UConn Carpentries Instructor/Helper

<https://docs.google.com/forms/d/e/1FAIpQLSdpQE7fYW9S9I1PRWg1WXyIFLmKjufw6lpcsIOgUZigCU1j7g/viewform>

Also see: <https://carpentries-uconn.github.io/>

bonus more plots

Cowplot theme allows you to easily make multipanel plots with different data. Make your plots and save them to an object.

```
plotA <- ggplot(data = gap, mapping = aes(x = gdpPercap, y = lifeExp)) +
  geom_point()+
  scale_x_log10()+
  theme_cowplot()
```

```
plotB <- ggplot(data = gap, mapping = aes(x = continent, y = lifeExp))+
  geom_boxplot()+
  theme_cowplot()
```

now combine the plots

```
plot_grid(plotA, plotB, labels = c("A", "B"))
ggsave(file = "figures/combined_plot.pdf", width = 10, height = 4, units = "in")
```

if you want to make one subplot bigger than the other

```
ggdraw()+
  draw_plot(plotA, x = 0, y = 0, width = 0.3, height = 1)
  draw_plot(plotB, x = 0.3, width = 0.7, height = 1)
# your plot space is 0 - 1 in both x and y.
```

```
# vignettes are a great way to learn how to use a new-to-you package  
browseVignettes("cowplot")
```

```
# interactive graphs  
yearLifeExp <- ggplot(data = gap, mapping = aes(x = year, y = lifeExp, group = country))+  
  facet_wrap(~ continent)+  
  geom_line() +  
  scale_x_log10()
```

```
yearLifeExp # view the plot object
```

```
ggplotly(yearLifeExp)
```

Now when you hover over your plot, a pop up window will show up with data. The data that is in the aes is show.