Welcome to Workshop: Making Graphs

I. Please sign in on the sign in sheet (so I can send you slides & follow up for feedback).

II. Download materials you'll need from my website (http://faculty.washington.edu/jhrl/Teaching.html or google Janneke HilleRisLambers at University of Washington – go to Teaching tab, scroll down (zip file under workshop III). Or ask me for a USB stick.

III. You'll need one script (ChickenScript_wk3.R), and data files (Chickens.csv, ChickenDiet.csv) which are identical to those you downloaded in previous weeks.

Graphs

I. Wrapup from last week
II. Data visualizations
    A. Goals
    B. Principles of effective graphs
III. Graphs (4 ‘assignments’)
    A. Scatterplots, line graphs, and boxplots
    B. Multi-panel graphs
IV. Additional Resources

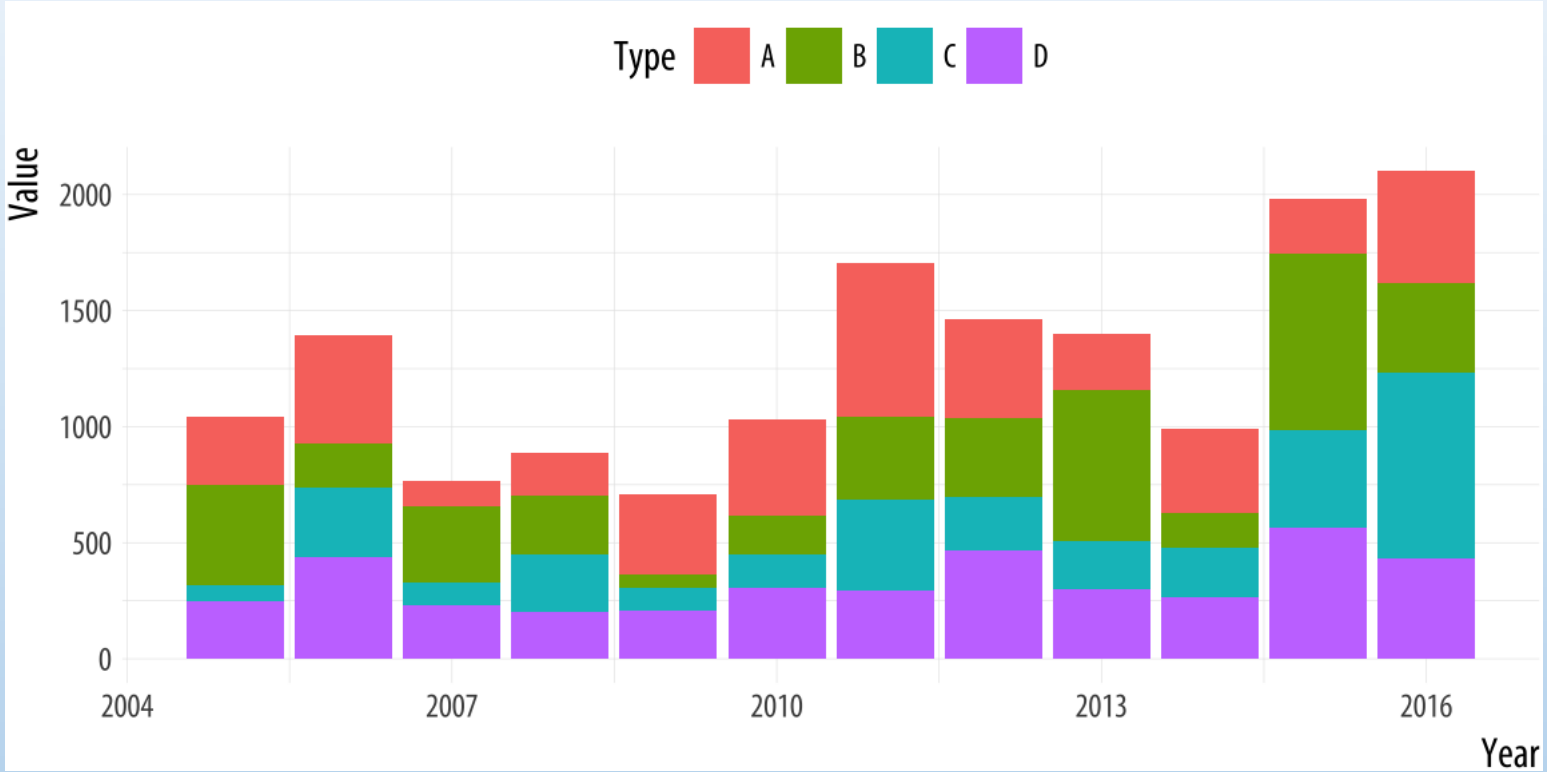
I. Wrapup from last week (before we move on): Your responsibility when coding…

- You must understand the statistics underlying the code you've written.
- This is true even if (and especially if) you 'pirate' code from someone else / the web (which is perfectly reasonable!).
- You must error proof your code and make sure your results are 'reproducible'). Run it by someone for review (just like you would a manuscript).
- You must write clear and well commented code that you must be willing to share (increasingly, a requirement for journals).

II. Graphs: what makes a ‘bad’ graph?

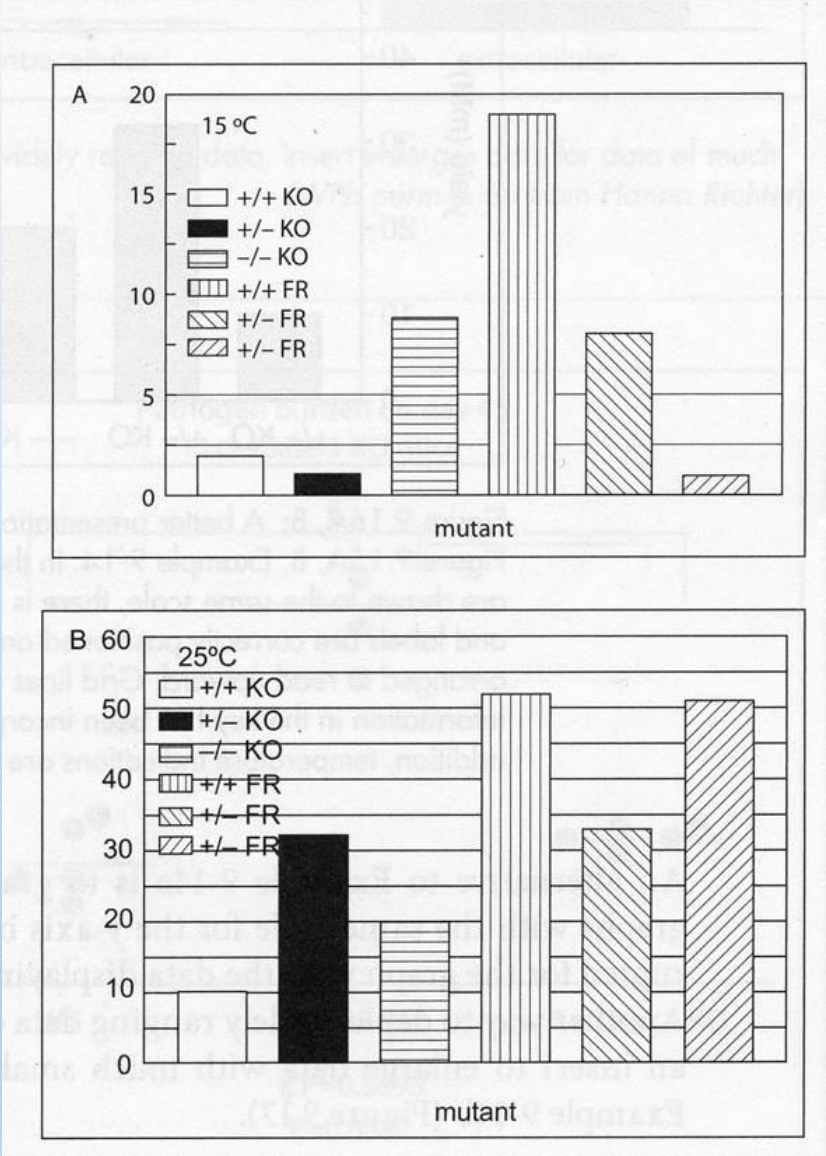
- How could this graph be improved?

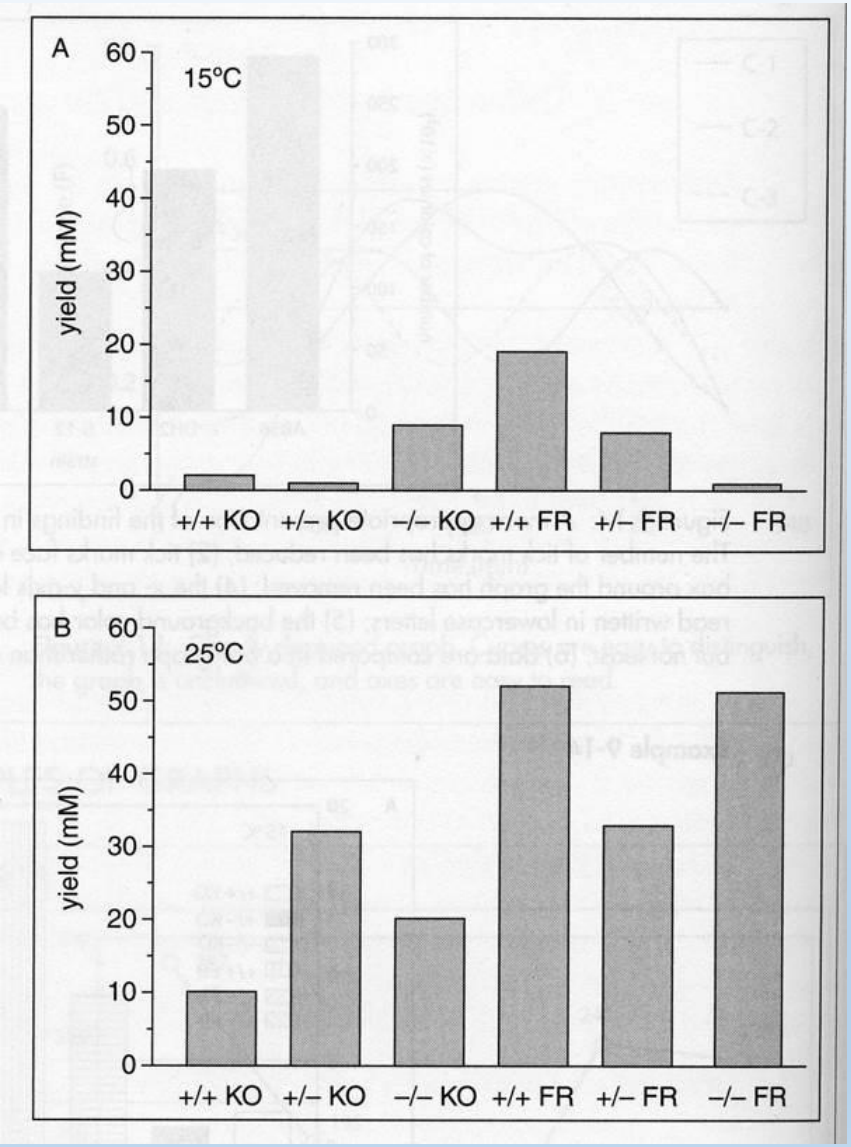
II. Graphs: what makes a ‘bad’ graph?



- How could this graph be improved?

II. Graphs: what makes a ‘bad’ graph?



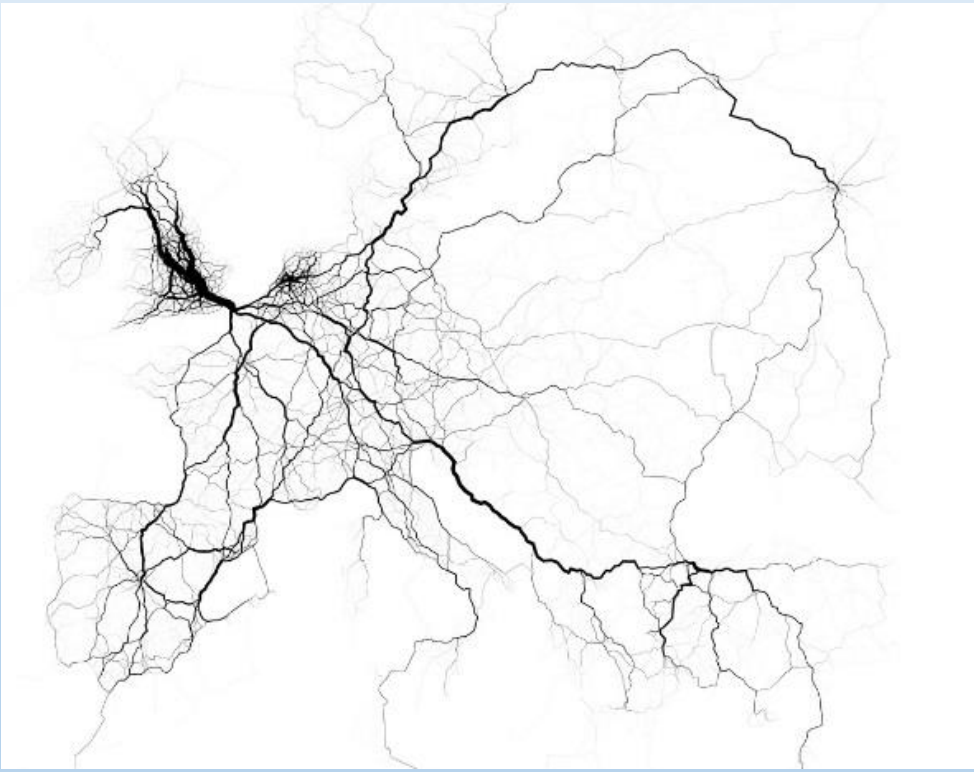


- How could this graph be improved?

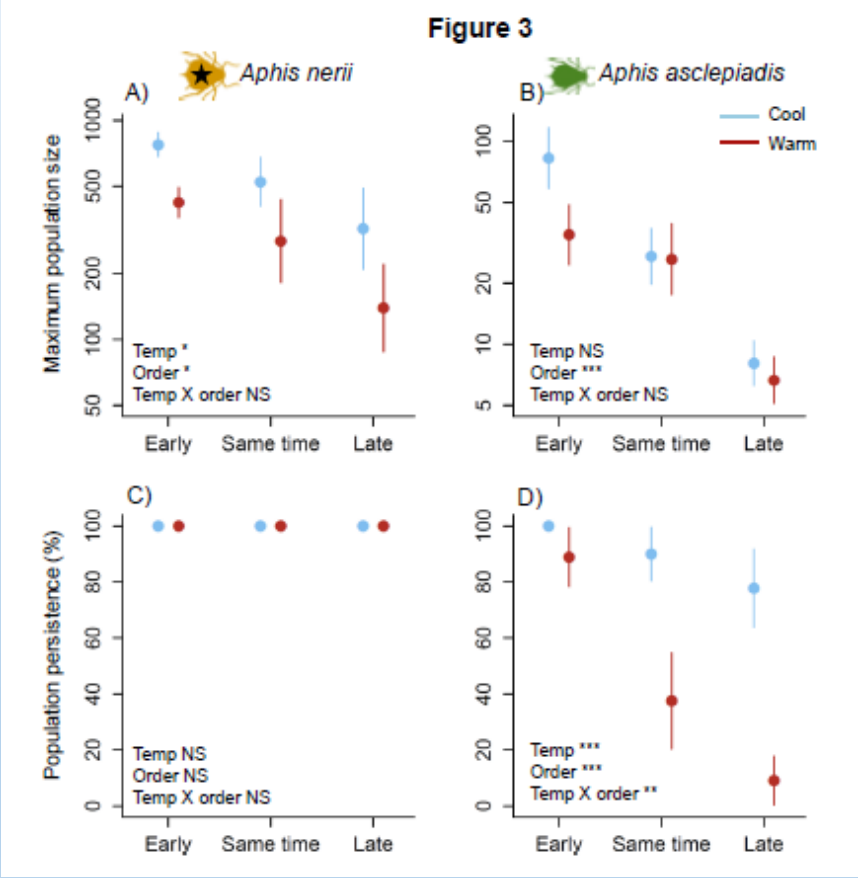
Workshop 3 (29/03/2018)

II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc). Exception: useful guides… data ‘art’…



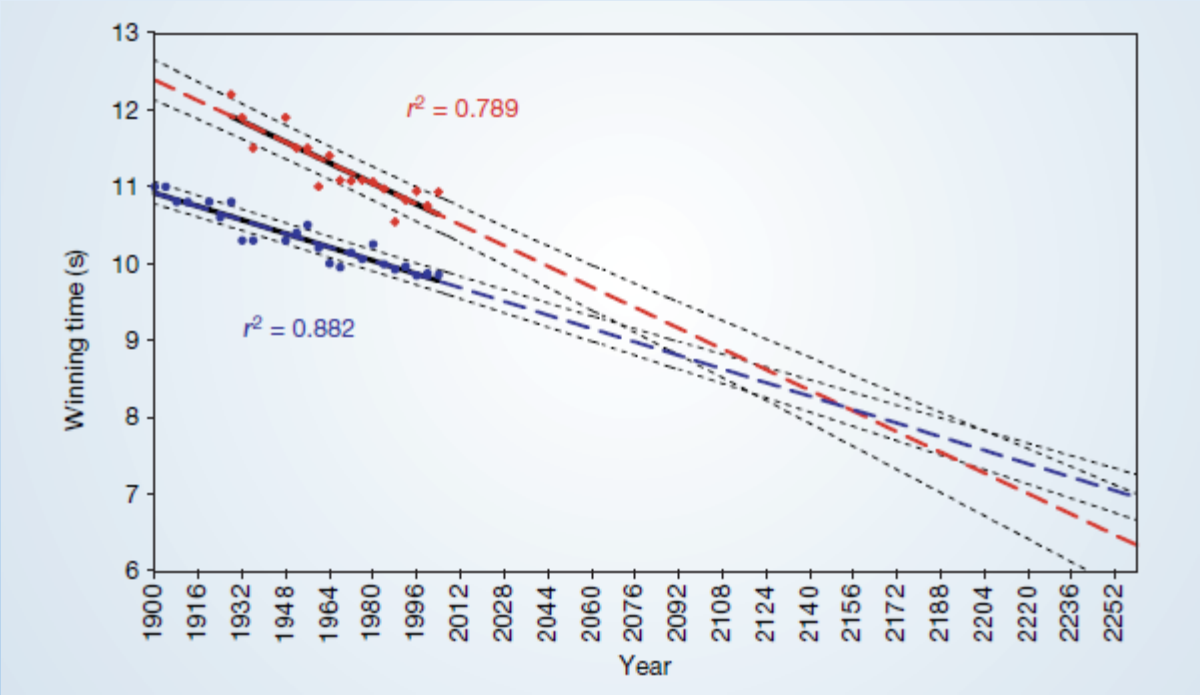
d.code: geo-coded tweets
https://news.sap.com/big-data-art-gallery/



Check out Tufte’s classic work

II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.



Should these trends continue, the projections will intersect at the 2156 Olympics, when — for the first time ever — the winning women’s 100-metre sprint time of 8.079 seconds will be lower than that of the men’s winning time of 8.098 seconds (Fig. 1).

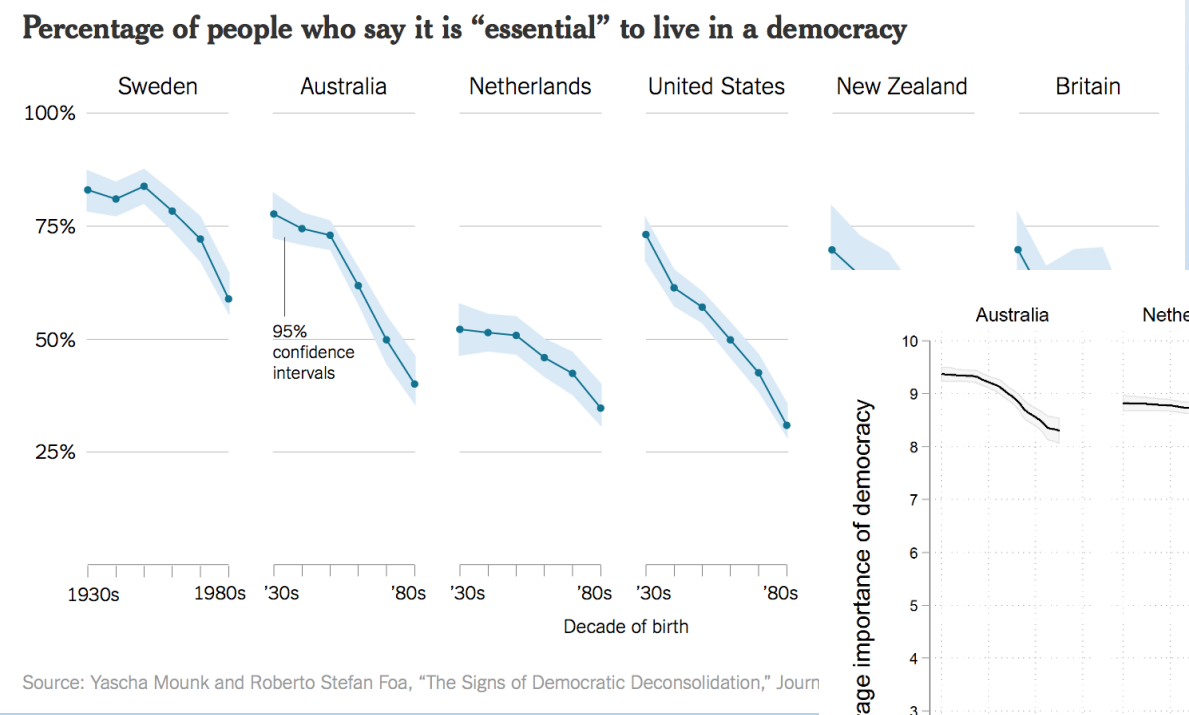
*Sir — A. J. Tatem and colleagues calculate that women may out-sprint men by the middle of the twenty-second century (Nature **431**,525; 2004). They omit to mention, however, that (according to their analysis) a far more interesting race should occur in about 2636, when times of less than zero seconds will be recorded.*

In the intervening 600 years, the authors may wish to address the obvious challenges raised for both time-keeping and the teaching of basic statistics.

Kenneth Rice

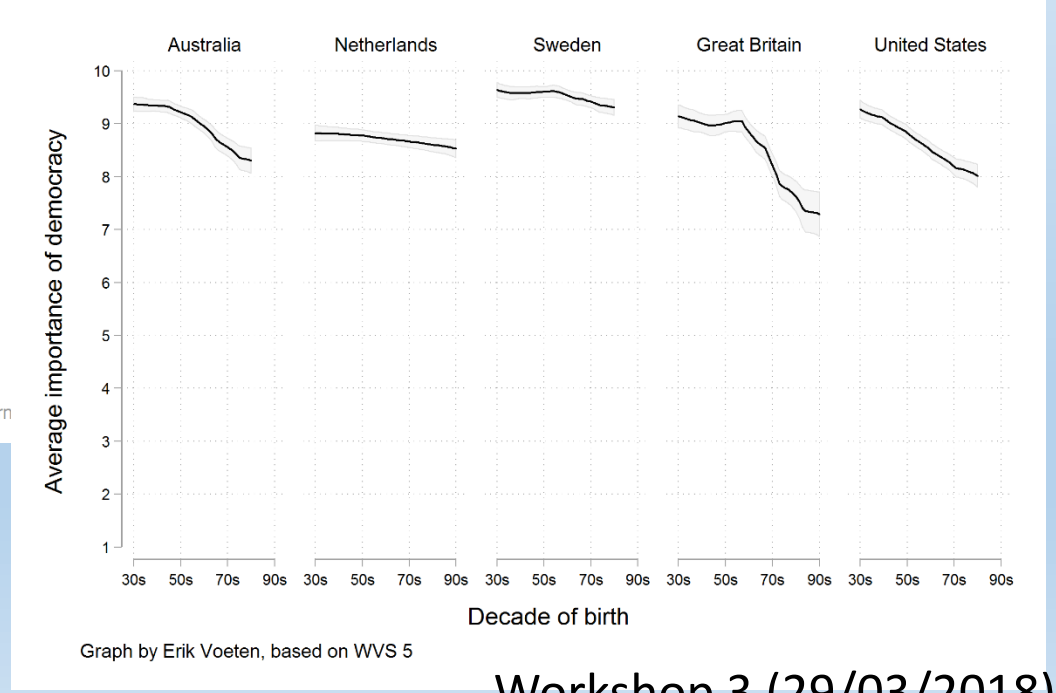
II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.



How Stable Are Democracies? ‘Warning Signs Are Flashing Red’” (New York Times, Taub, 2016).

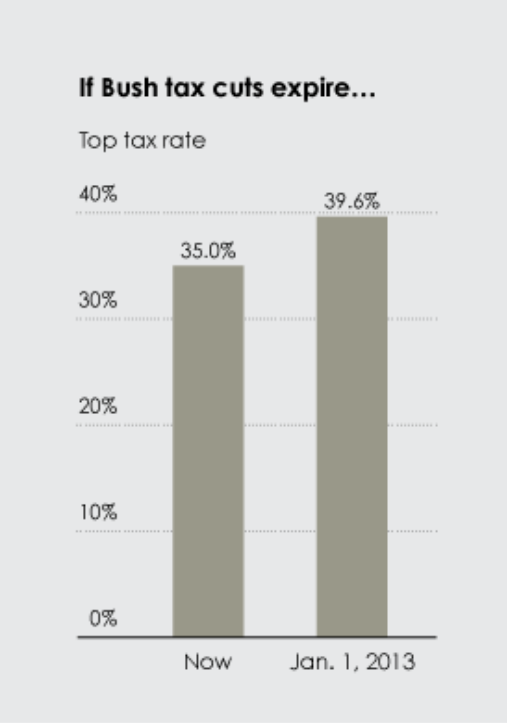
Perhaps the crisis is overblown
Erik Voeten



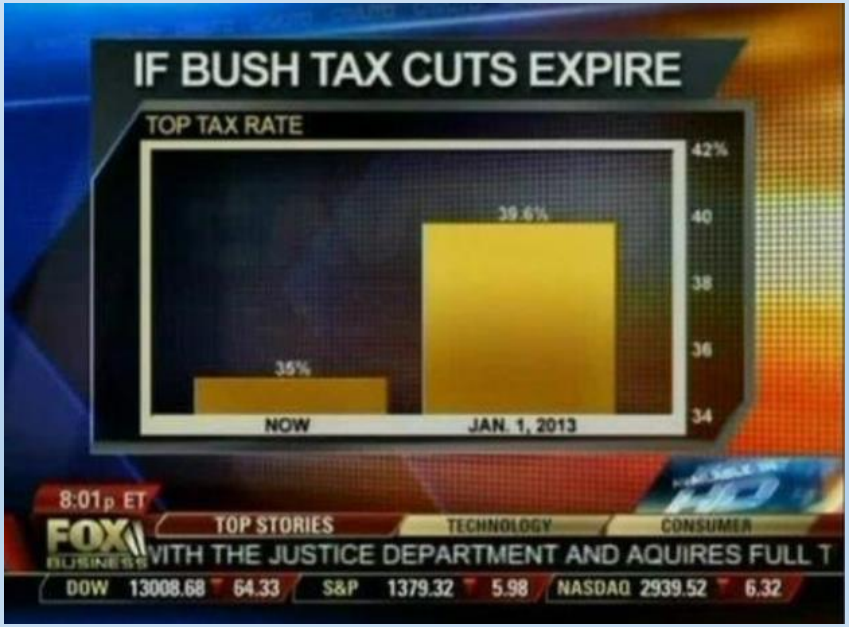
II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: (in)advertently misleading

Well maybe not…



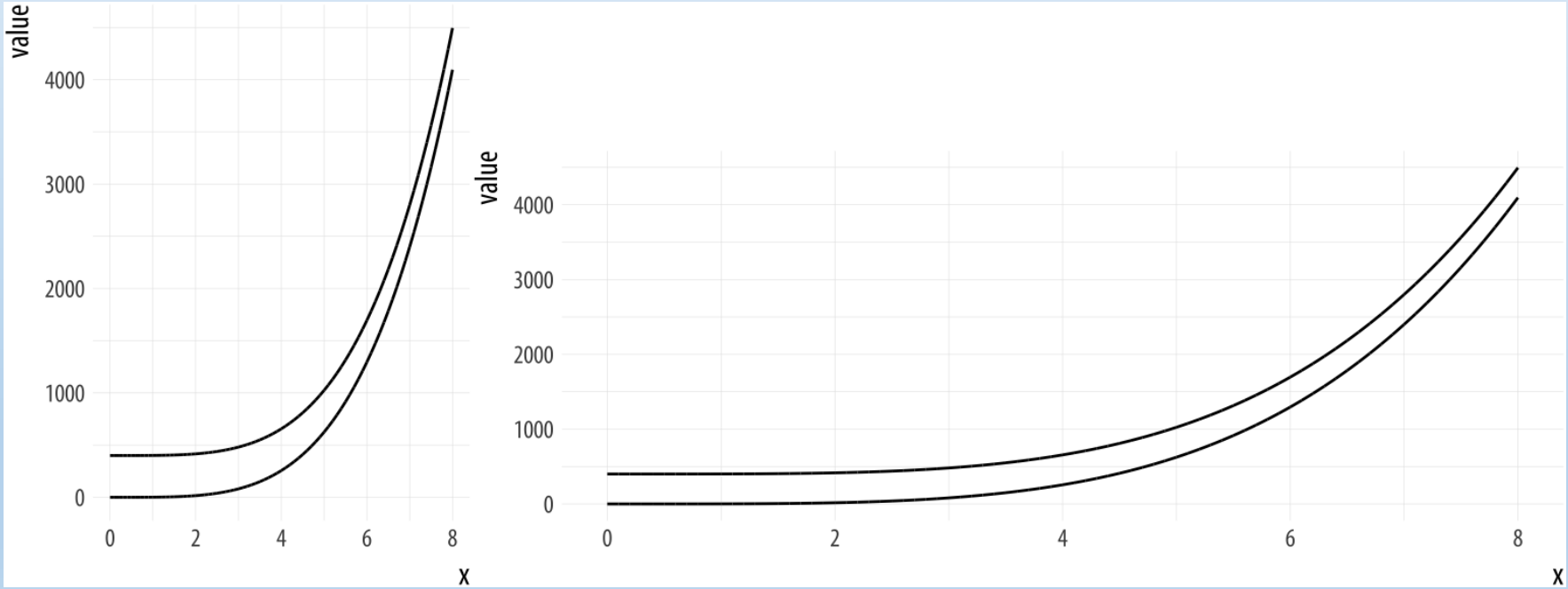
Huge increase in application of top tax rate!



II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading

Dimensions of graphs influence our perception of rate of change..



Ideally: squarish graphs (wider graphs for time)
Same size / axis (if want to compare side by side or top to bottom)

II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading

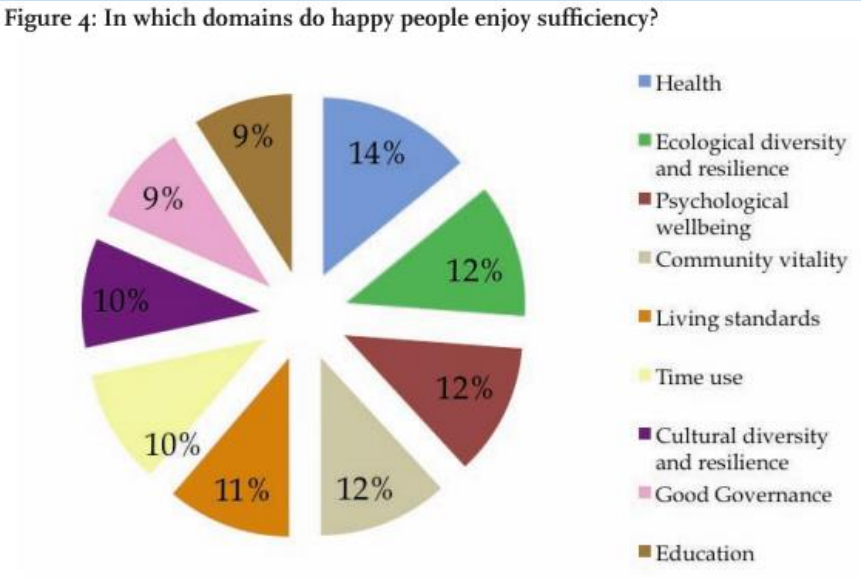
Consider whether colors represent ordered (big to small), diverging, or categorical differences…

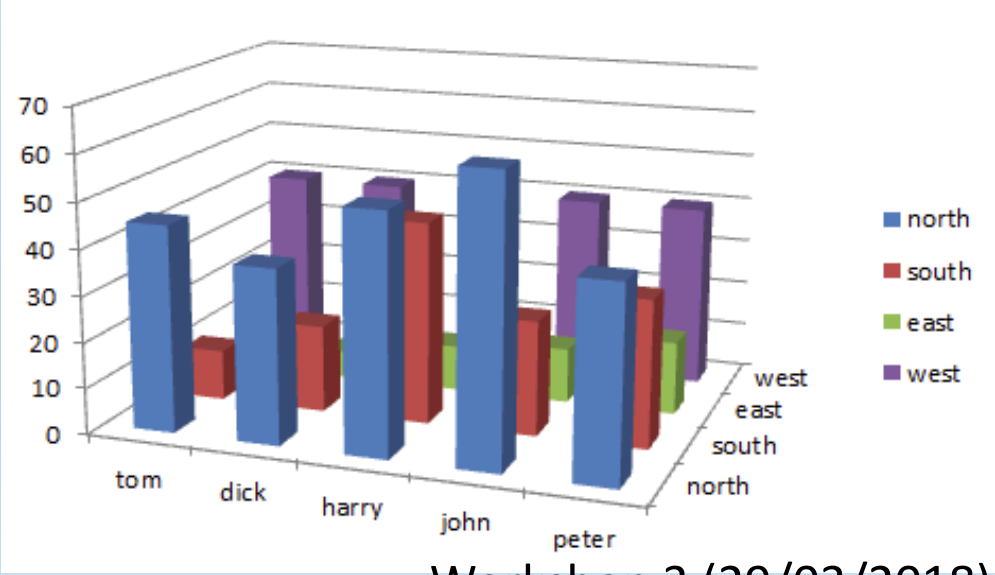
Never use more than 6 colors on a graph (we can’t really distinguish more than that, visually)

II. Graphs: what makes a 'bad' graph?

- Aesthetics: reduce 'chart junk' (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading, confusing

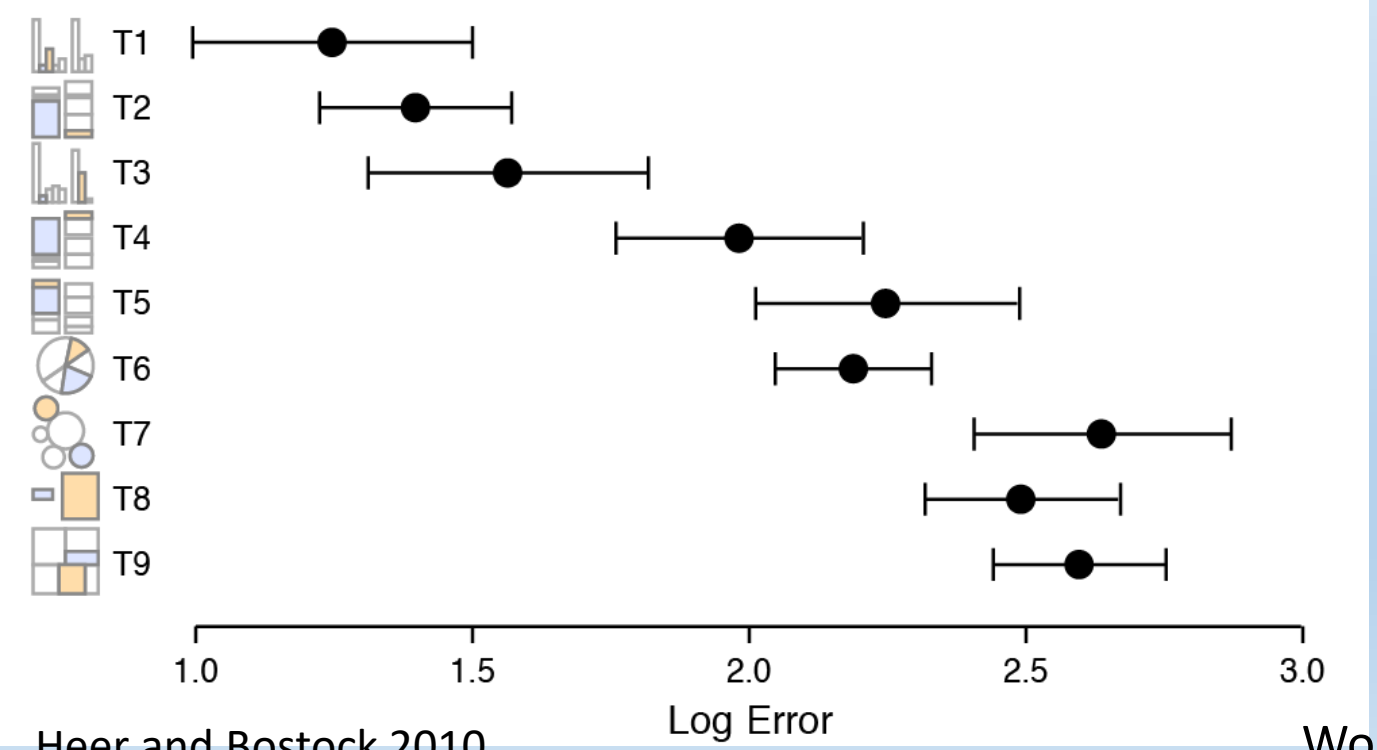
Never use 'blow apart' graphs and 3D bars (hard for us to interpret)





II. Graphs: what makes a 'bad' graph?

- Aesthetics: reduce 'chart junk' (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading, confusing, easy to misunderstand



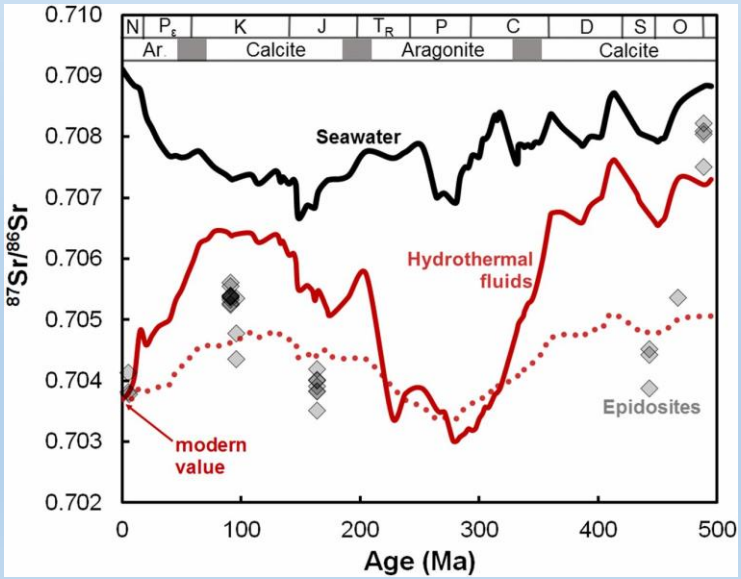
We are better at detecting differences in lengths than angles or area (thus the maligned pie chart)

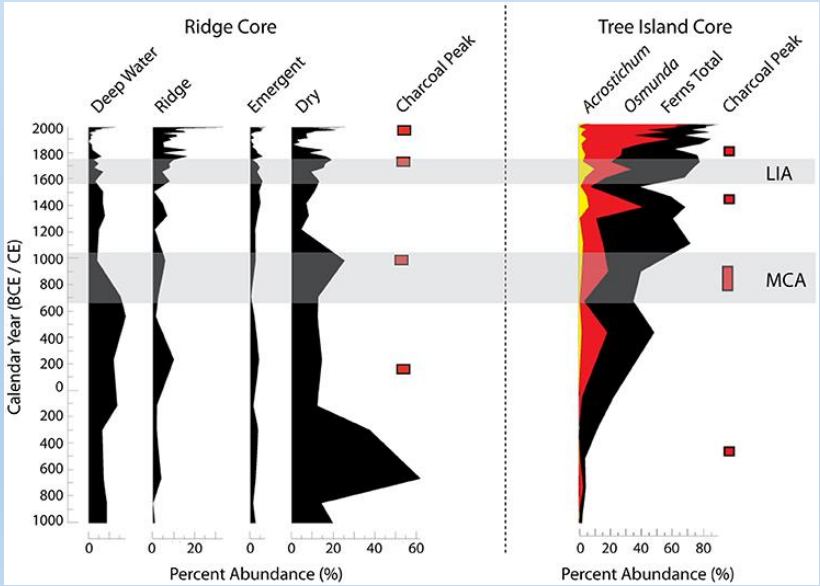
Heer and Bostock 2010

II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading, confusing, easy to misunderstand, defies convention

Which direction should time go? (Western Europeans assume left to right)

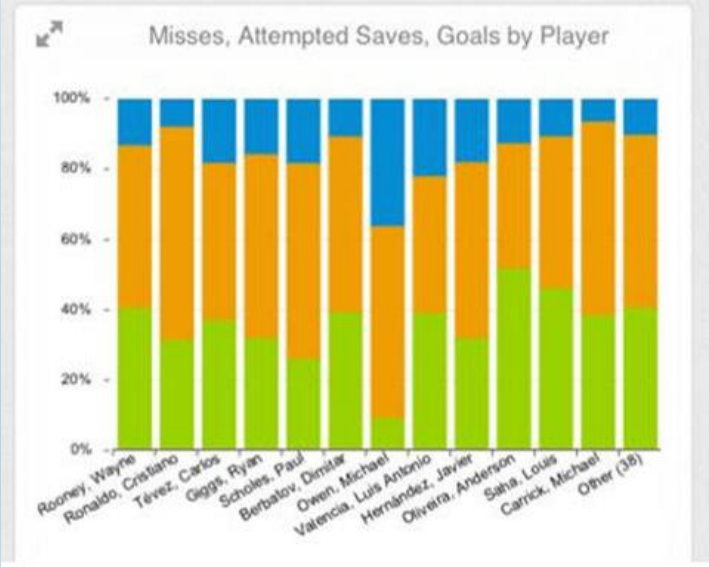


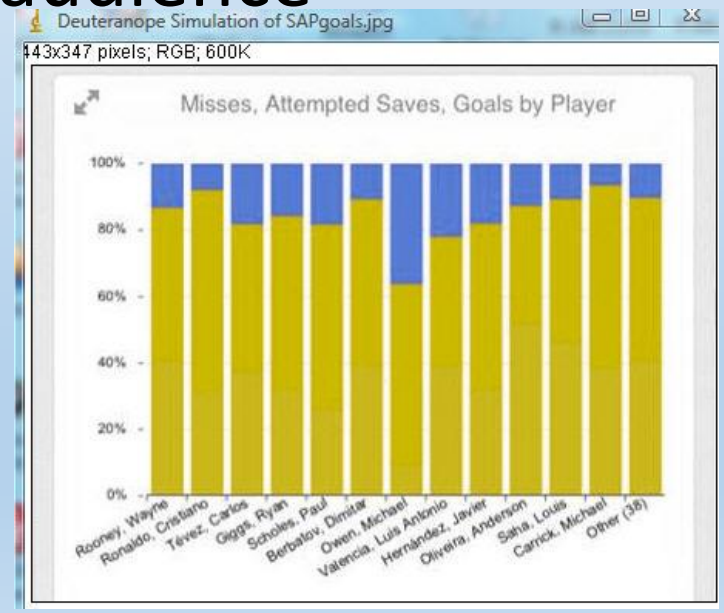


II. Graphs: what makes a ‘bad’ graph?

- Aesthetics: reduce ‘chart junk’ (i.e. unnecessary colors, 3D boxes, etc).
- Substantive: inappropriate data.
- Perception: inadvertently misleading, confusing, easy to misunderstand, defies convention, not accessible for a portion of your audience

Consider that ~4.5% of your audience is colorblind (men more so)





http://www.color-blindness.com/coblis-color-blindness-simulator/

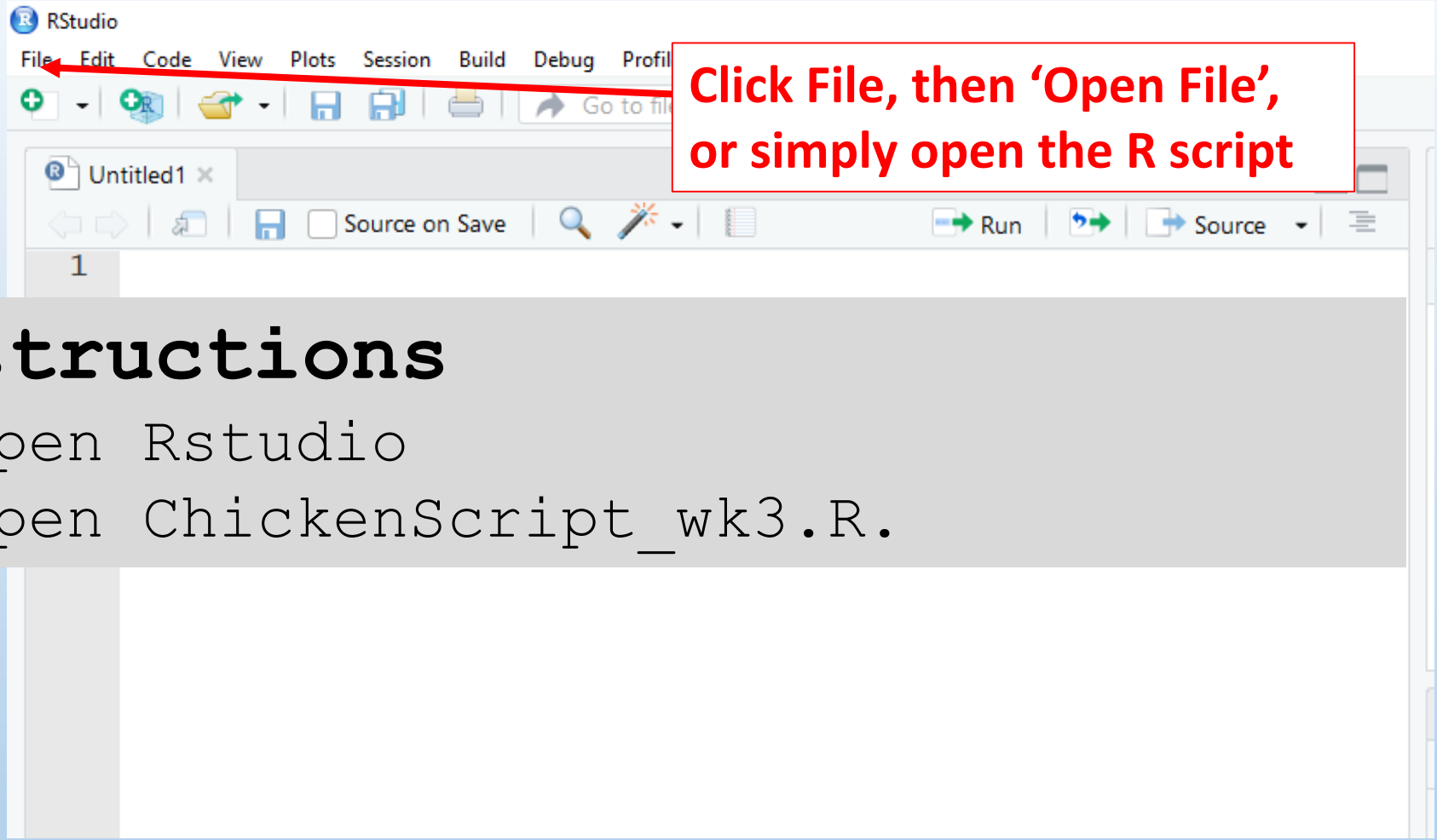
III. Graphs: reminder

```
These are instructions

Do / look / find this
> Type this (but not the >)
```

This is something useful / important

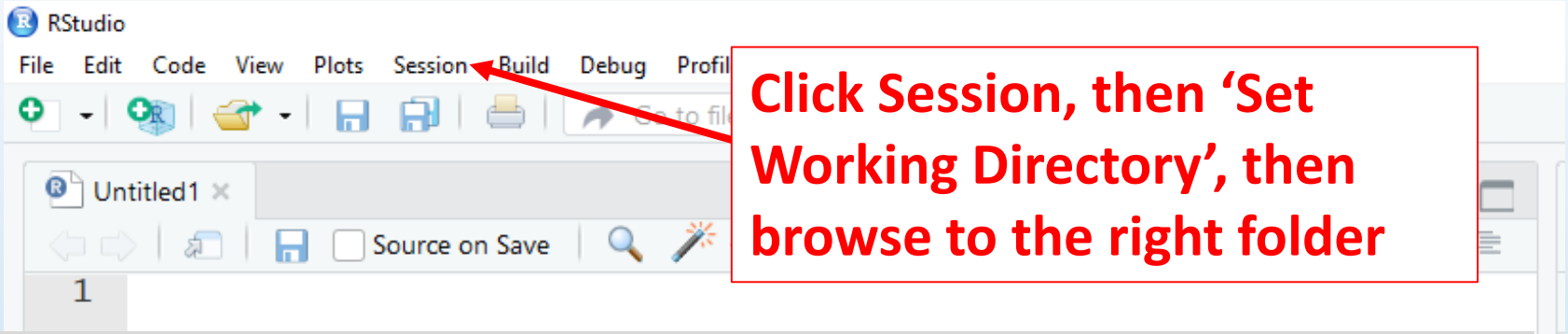
III. Graphs



Instructions

1. Open Rstudio
2. Open ChickenScript_wk3.R.

III. Graphs

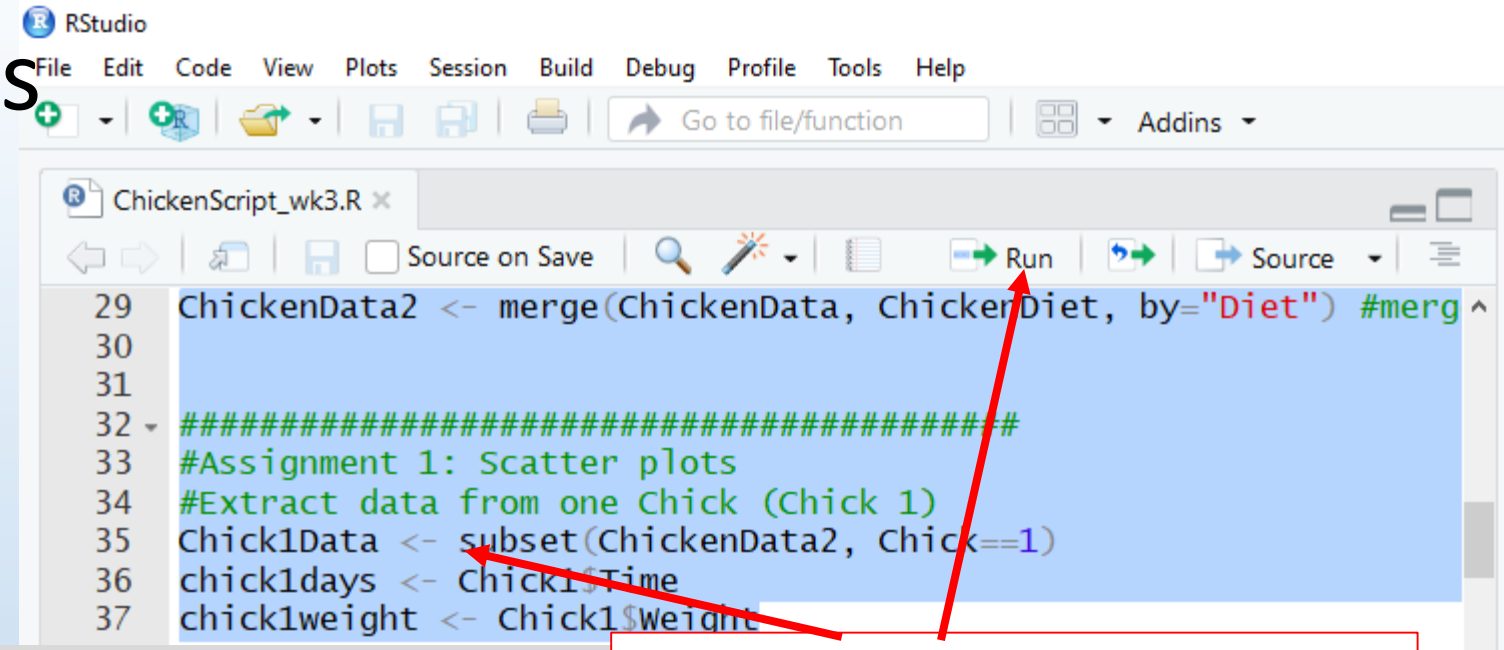


Click Session, then 'Set Working Directory', then browse to the right folder

Instructions

1. Open Rstudio
2. Open ChickenScript_wk3.R.
3. Set your directory to where you have the Chicken Data (Chickens.csv and ChickenDiet.csv)

III. Graphs



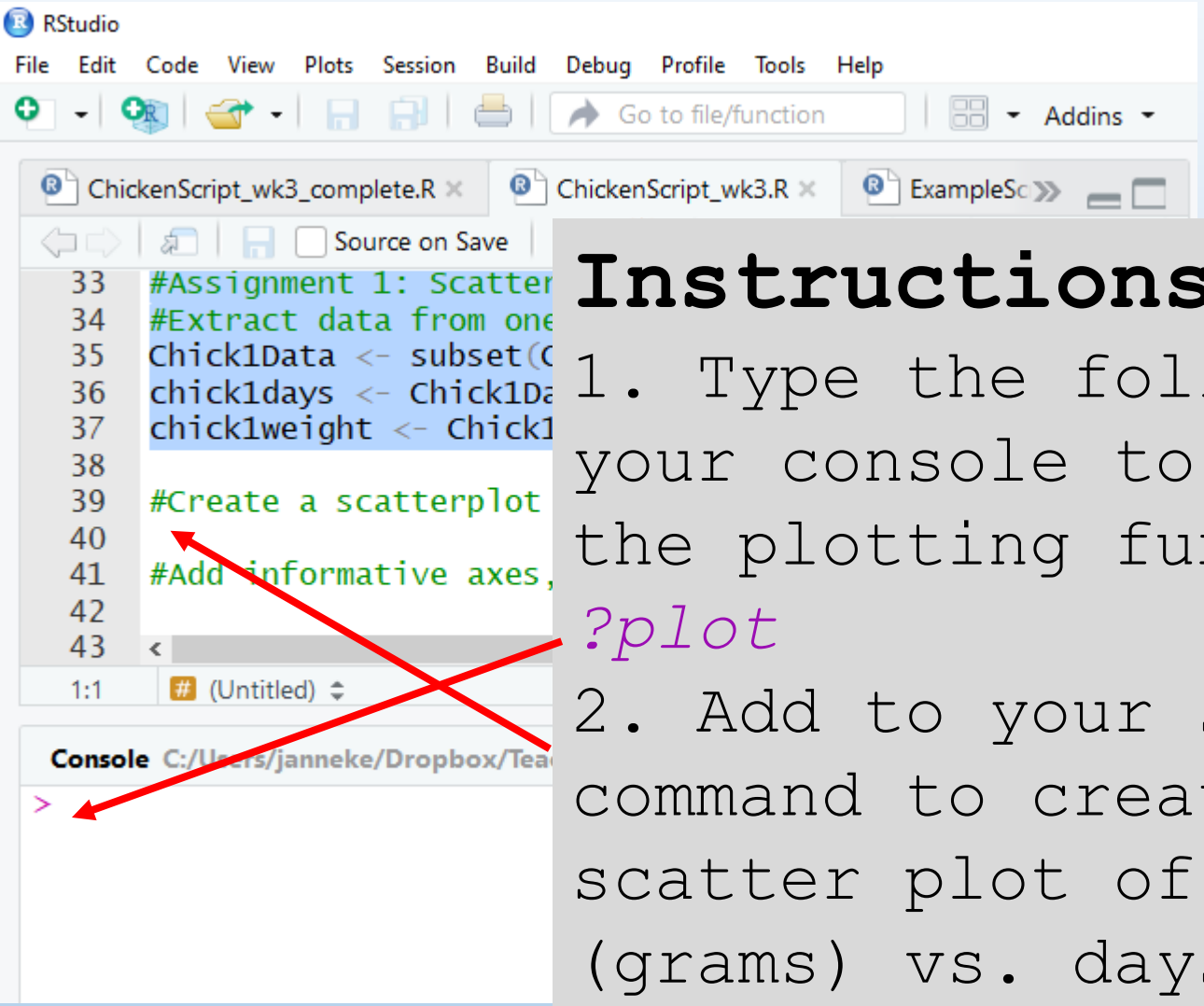
Highlight lines of code up to and including Assignment 1

Click on 'Run'

Instructions

1. Open Rstudio
2. Open ChickenScript_w
3. Set your directory to where you have the Chicken Data (Chickens.csv and ChickenDiet.csv)
4. Run the lines in the script until you get to Assignment 1, scatterplots

III. Graphs: scatterplots



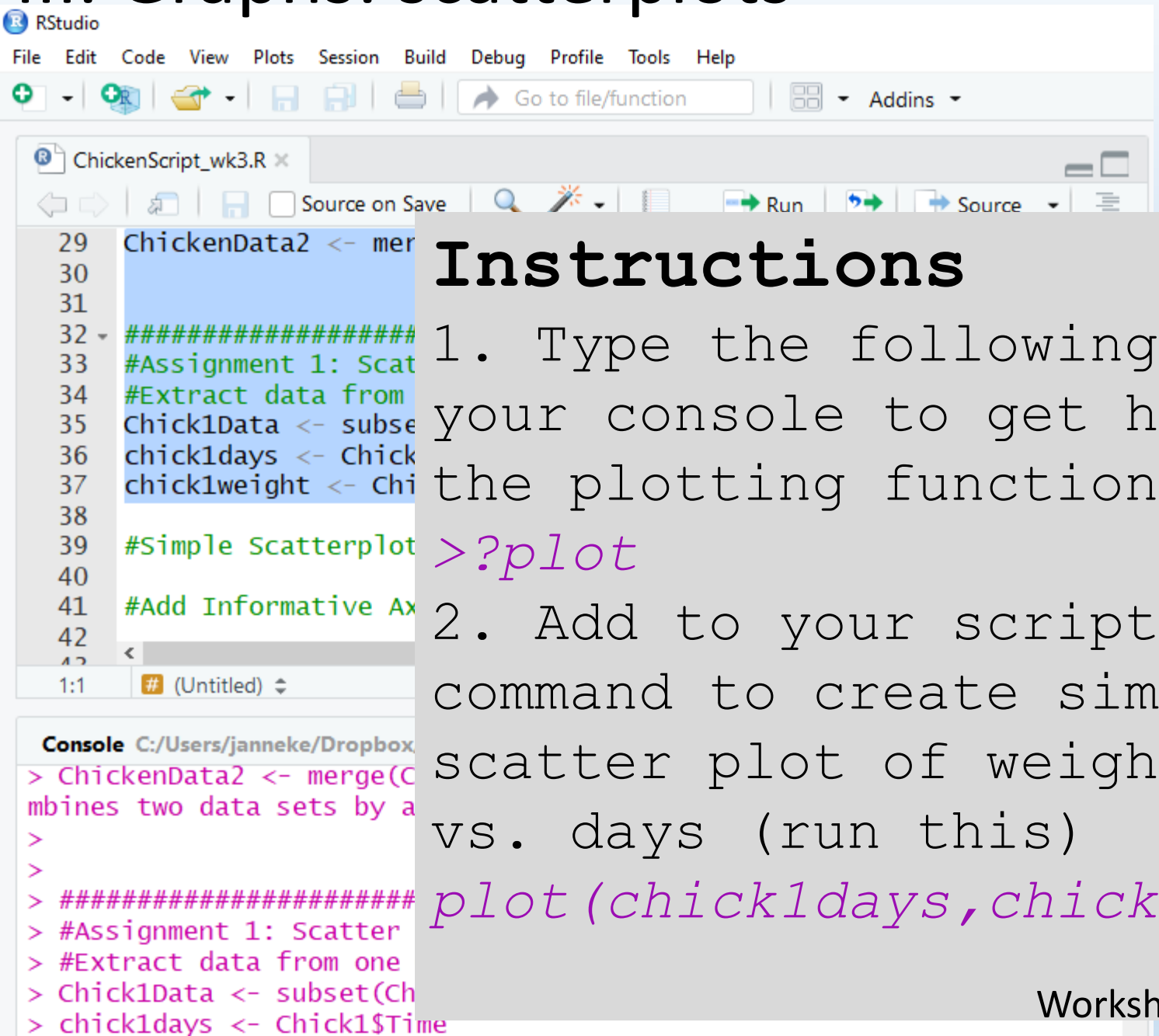
Instructions

1. Type the following into your console to get help on the plotting function

?plot

2. Add to your script a command to create simple scatter plot of weight (grams) vs. days (run this)

III. Graphs: scatterplots



Instructions

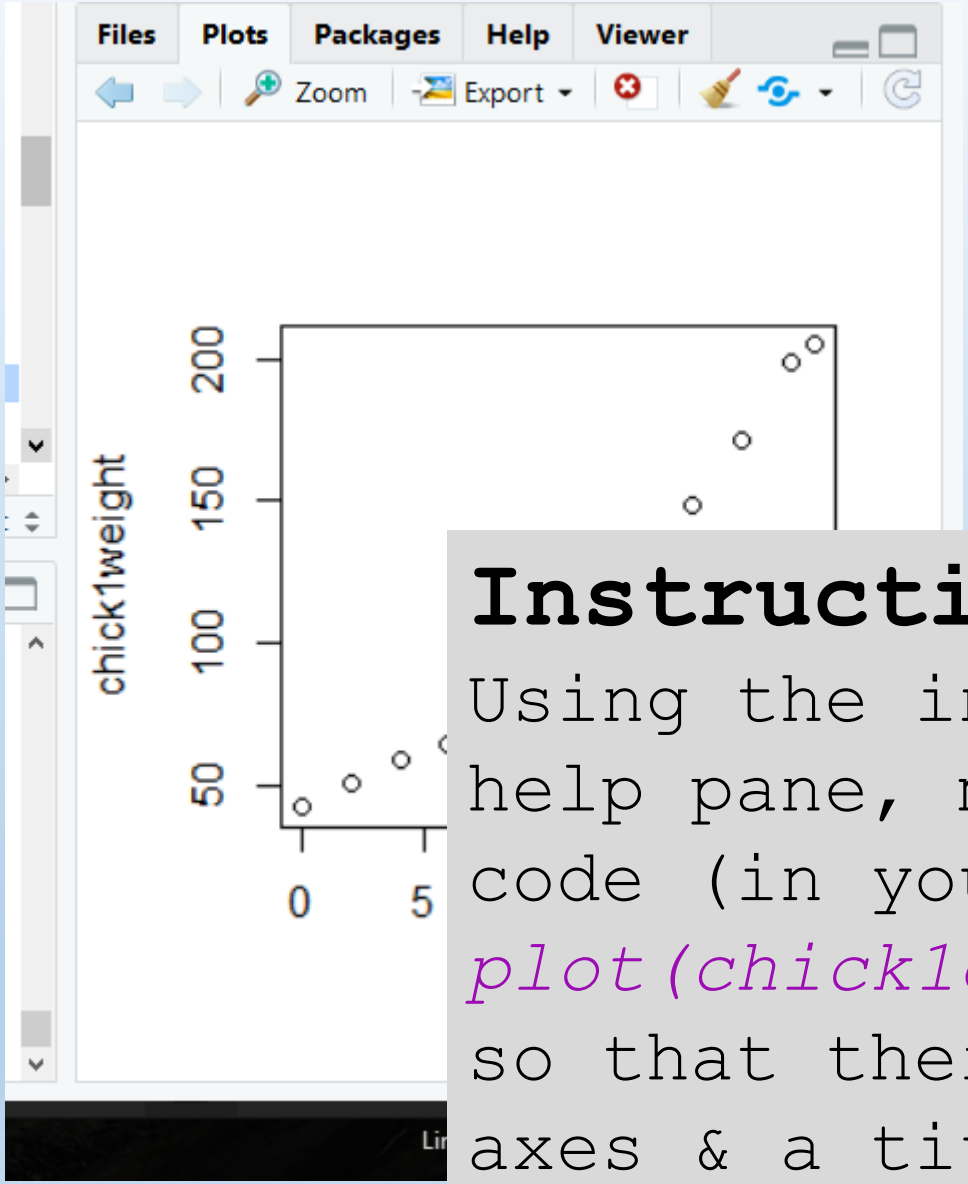
1. Type the following into your console to get help on the plotting function

>?plot

2. Add to your script a command to create simple scatter plot of weight (grams) vs. days (run this)

plot(chick1days,chick1weight)

III. Graphs: scatterplots



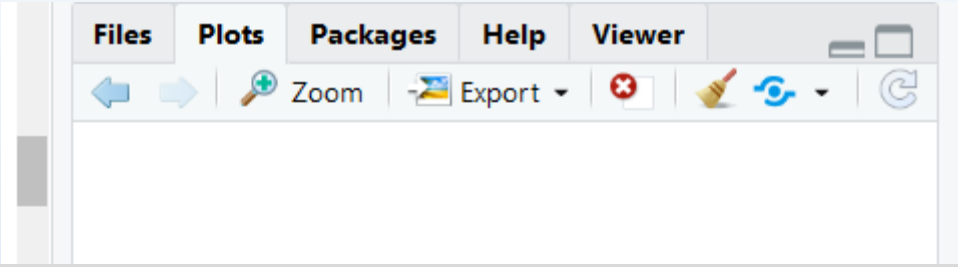
Instructions

Using the information in the help pane, modify this line of code (in your script):

plot(chick1days,chick1weight)

so that there are informative axes & a title.

III. Graphs: scatterplots



Instructions

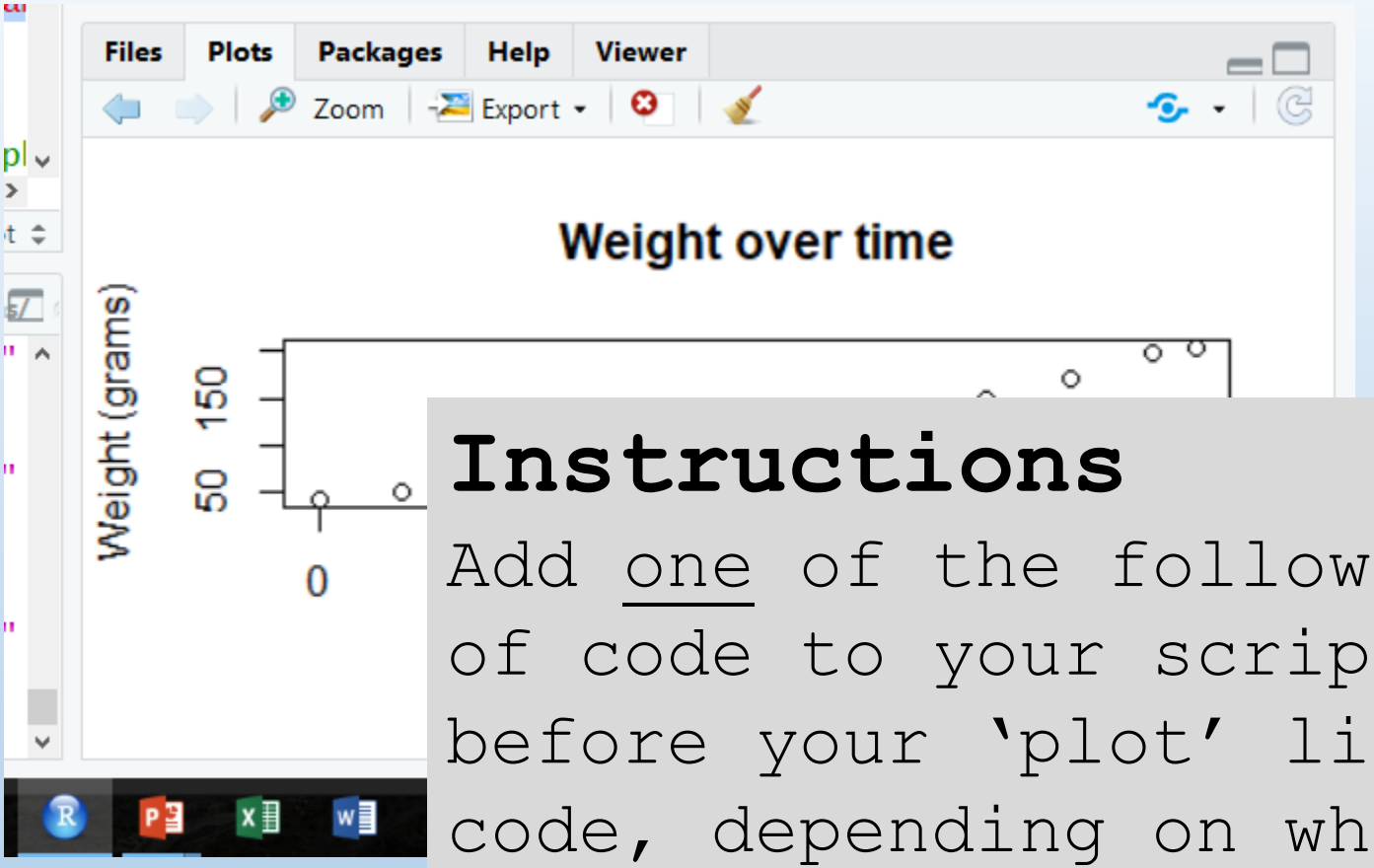
Using the information in the help pane, modify this line of code in your script:

```
plot(chick1days,chick1weight)
```

so that there are informative axes & a title.

```
plot(chick1days, chick1weight, xlab="Time (days)", ylab="Weight (grams)", main="Weight over time")
```

III. Graphs: scatterplots

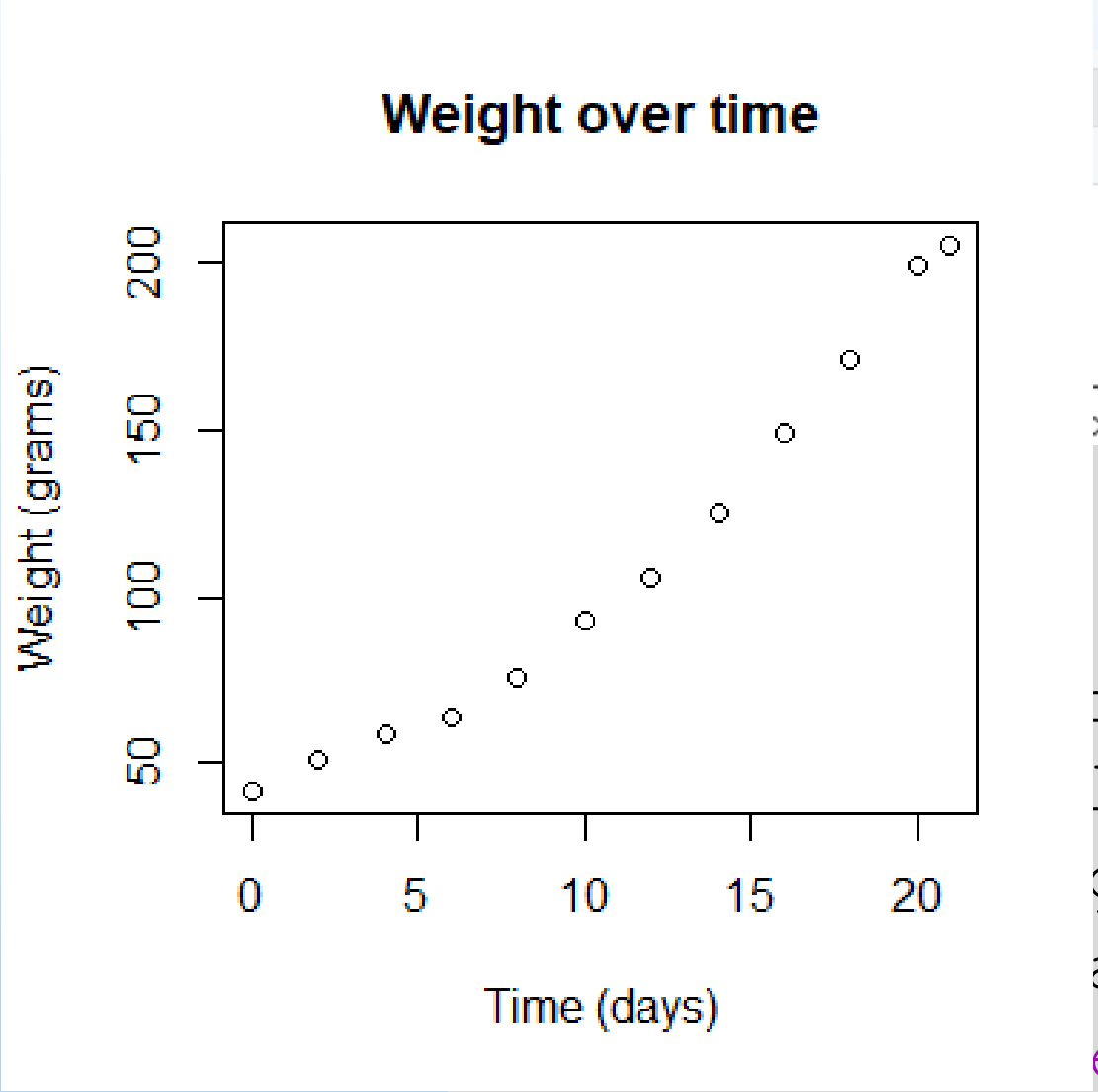


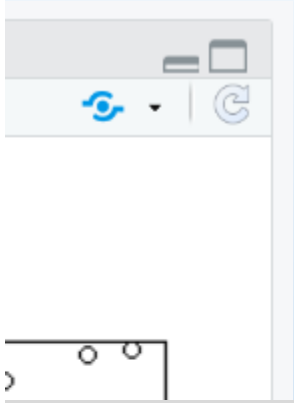
Instructions

Add <u>one</u> of the following lines of code to your script right before your 'plot' line of code, depending on whether you have a PC or Mac:

```
X11(width=5, height=5) #PC
Quartz(width=5, height=5) #Mac
```

III. Graphs: scatterplots





s

following lines
r script right
lot' line of
g on whether you
ac:

```
eight=5) #PC
Quartz(width=5, height=5) #Mac
```

III. Graphs: scatterplots

Instructions

1. Write in your console (to see what kinds of colors you can use):

```
> colors()
```

2. Change your 'plot' command (in your script) so that the points have a color of your choice (you can use pages 3-8 of RColorChart.pdf for exact colors). Remember to use:

```
> ?plot.default
```

if you need help on how to use plot.

III. Graphs: scatterplots

Instructions

1. Write in your console (to see what kinds of colors you can use):

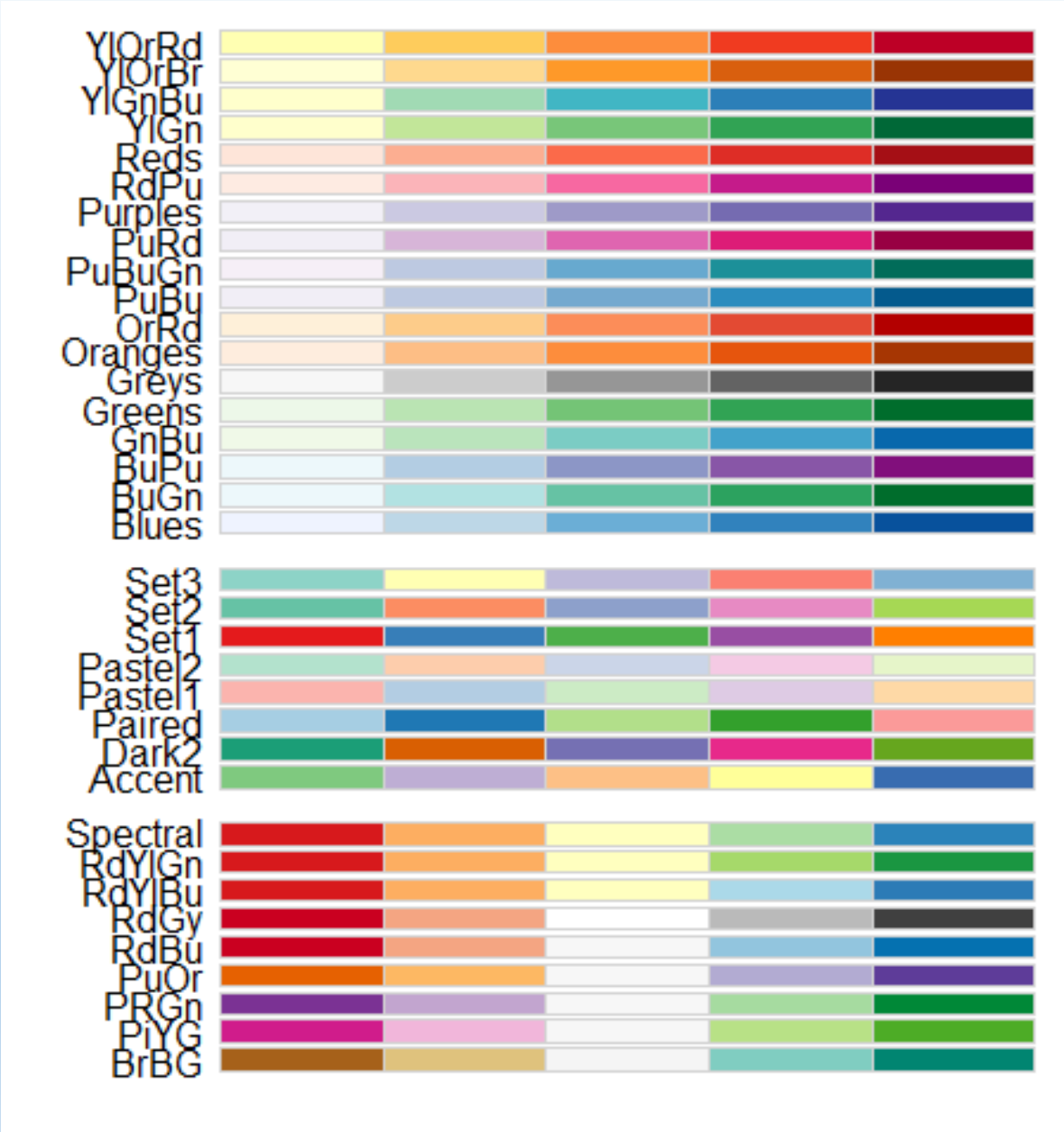
```
> colors()
```

2. Change your 'plot' command (in your script) so that the points have a color of your choice.

```
plot(chick1days, chick1weight,
col="yellow4", xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
```

III. Graphs: color

- RColorBrewer package & interactive viewer
- Color Palette Cheatsheet



III. Graphs: scatterplots

Instructions

1. Write in your console (to see what point shapes are available):

```
> example(points)
```

2. Change your 'plot' command (in your script) so that the points have a shape of your choice (you'll need to look at plot.default, not plot to see how to change your command)

```
> ?plot.default
```

III. Graphs: scatterplots

Instructions

1. Write in your console (to see what point shapes are available):

```
> example(points)
```

2. Change your 'plot' command (in your script) so that the points have a shape of your choice.

```
plot(chick1days, chick1weight,
col="yellow4", pch=17, xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
```

III. Graphs: scatterplots

Instructions

1. Writ
point s
> examp
2. Chan
script) e of
your ch

pch controls the type of symbol, either an integer between 1 and 25, or any single char within ""

1 ○	2 △	3 +	4 ×	5 ◇	6 ▽	7 ⊠	8 ✳
9 ⊕	10 ⊕	11 ✡	12 ⊞	13 ⊗	14 ◪	15 ■	
16 ●	17 ▲	18 ◆	19 ●	20 •	21 ○	22 □	23 ◇
24 △	25 ▽	* *	. ·	X X	a a	? ?	

```
plot(chick1days, chick1weight,
col="yellow4", pch=17, xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
```

III. Graphs: scatterplots

Instructions

1. Write in your console (to see what point shapes are available):

```
> example(points)
```

2. Change your 'plot' command (in your script) so that the points have a shape of your choice.

```
plot(chick1days, chick1weight,
bg="yellow4", pch=21, xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
```

Col for filled points (21-25) specifies border (default black), bg specifies fill

III. Graphs: scatterplots

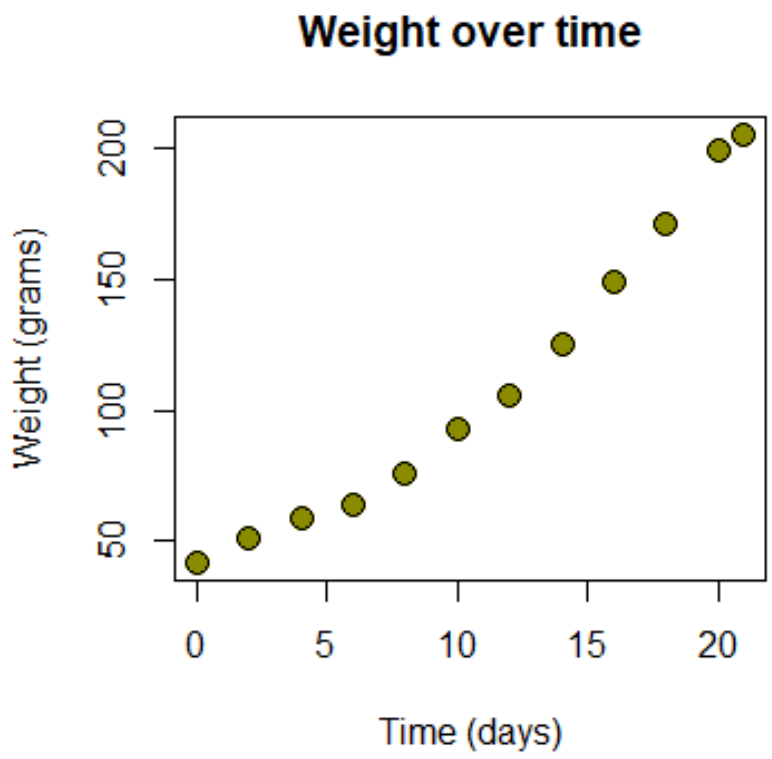
Instructions

1. Write in your consol
point shapes are availa

```
> example(points)
```

2. Change your 'plot' c
script) so that the poi
your choice.

```
plot(chick1days, chick1weight,
bg="yellow4", pch=21, xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
```



III. Graphs: scatterplots

The function abline will add a line to any existing plot. You can specify vertical, horizontal, or lines with a intercept and slope. Check it out!

```
> ?abline
```

Instructions (2 lines of code)

1. Use a linear regression to find the relationship between time and chick weight. Hint: use function lm.
2. Add the best fit line to your plot. Hint: extract slope/intercept using coef, abline command must come after the plot command

III. Graphs: scatterplots

Instructions (2 lines of code)

1. Use a linear regression to find the relationship between time and chick weight. Hint: use function lm.

```
chick1test <- lm(chick1weight~chick1time)
```

2. Add the best fit line to your plot.

```
plot(chick1days, chick1weight,
col="yellow4", pch=17, xlab="Time (days)",
ylab="Weight (grams)", main="Weight over
time")
abline(coef(chick1test))
```

III. Graphs: scatterplots

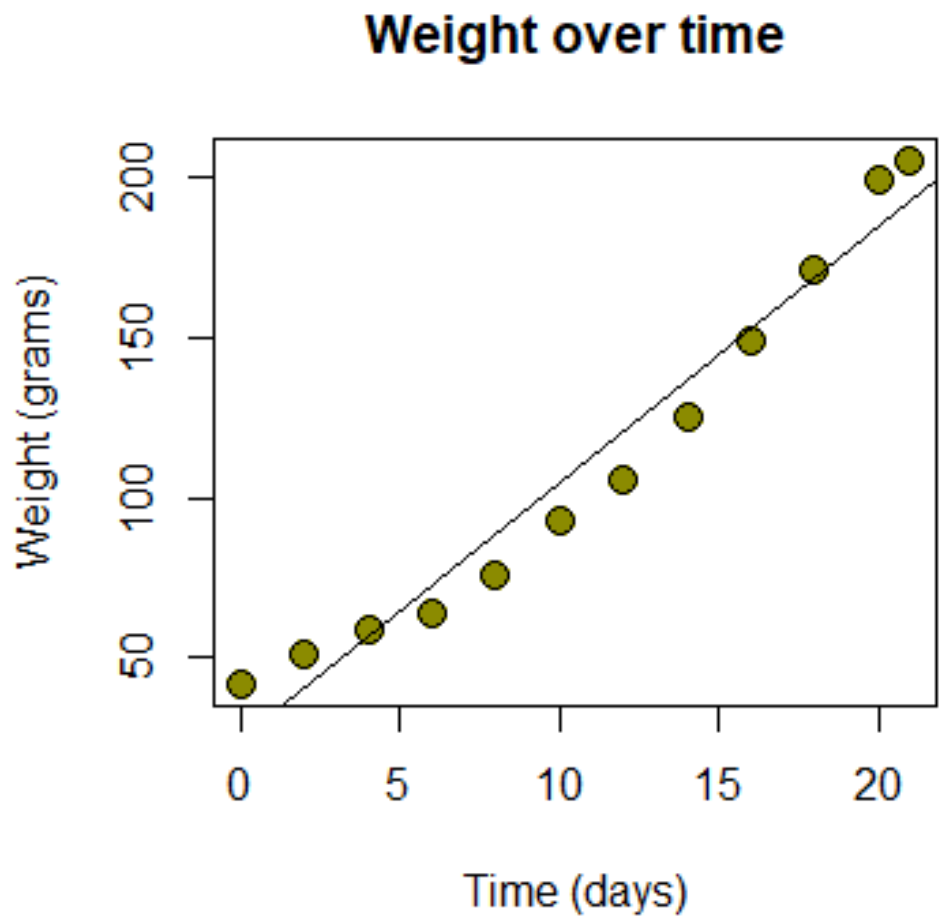
Instructions (2 lines of code)

1. Use a linear regr
   relationship betw
   weight. Hint: use

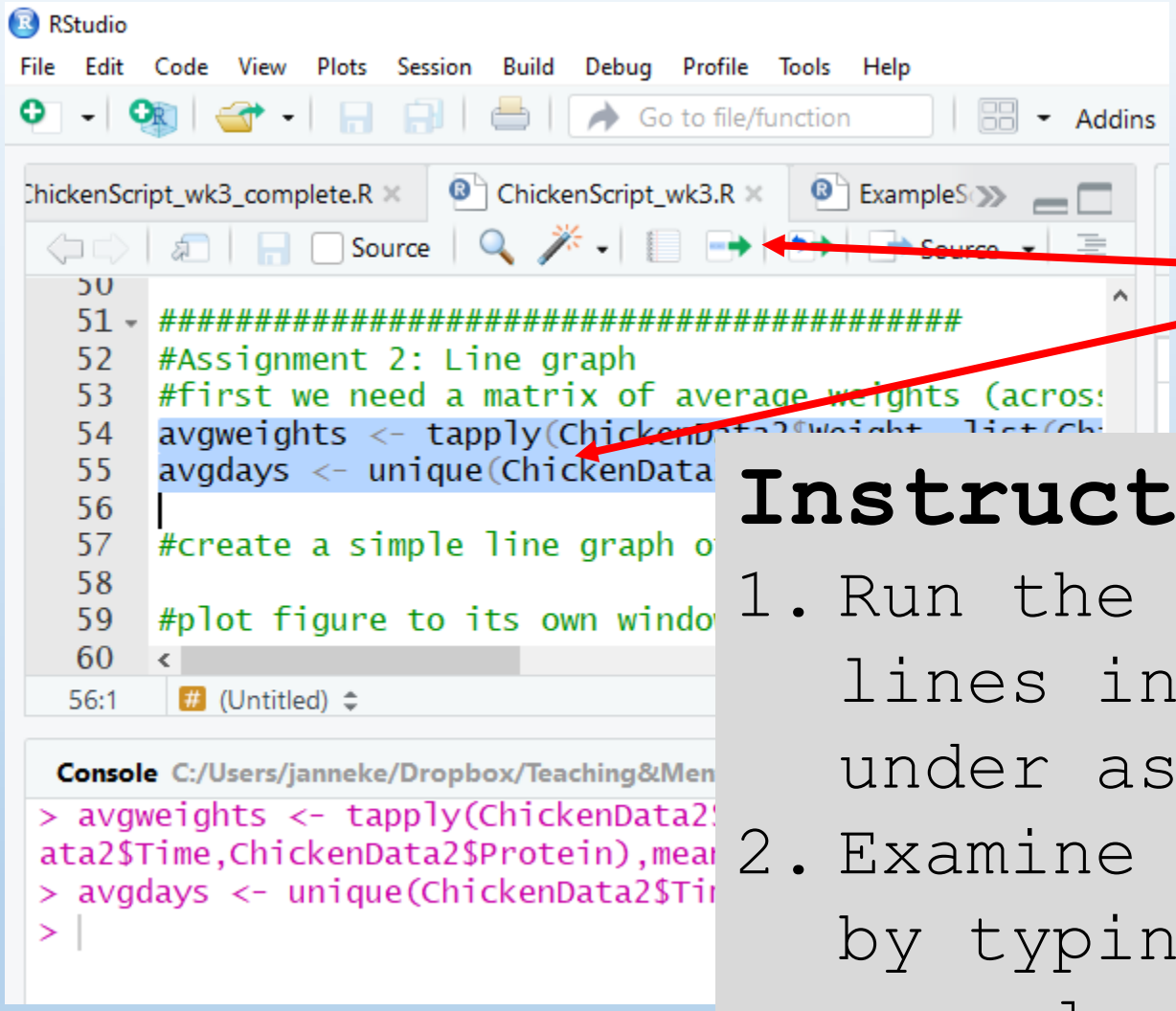
```
chick1test <- lm(ch
```

2. Add the best fit

```
plot(chick1days, ch
col="yellow4", pch=
ylab="Weight (grams
time")
abline(coef(chick1t
```



III. Graphs: line graphs



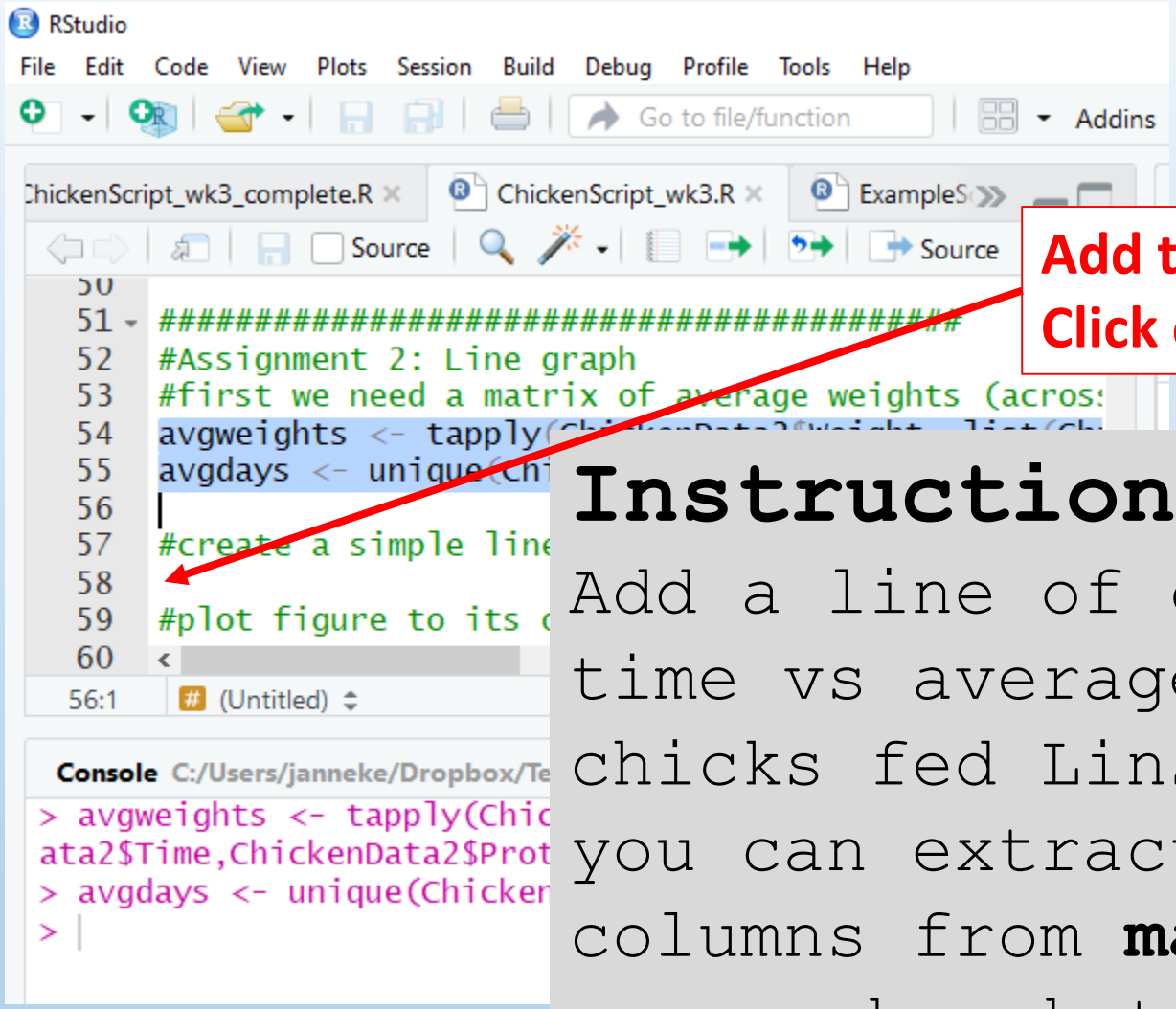
Highlight these lines of code and click on 'Run'

Instructions

1. Run the first few lines in your script under assignment 2
2. Examine averageweights by typing in your console

```
> avgweights
```

III. Graphs: line graphs



Add to script here
Click on 'Run'

Instructions

Add a line of code to plot time vs average weight of chicks fed Linseed. Hint: you can extract rows and columns from **matrices** using square brackets [row#,col#]

III. Graphs: line graphs

Instructions

Add a line of code to plot time average weight of chicks fed Linseed.

```
plot(avgdays,avgweights[,1])
```

However, we wanted a line graph… Check the plot function to see how to do this, and alter your code (hint: you will need to specify the ‘type)

III. Graphs: line graphs

Instructions

Add a line of code to plot time average weight of chicks fed Linseed.

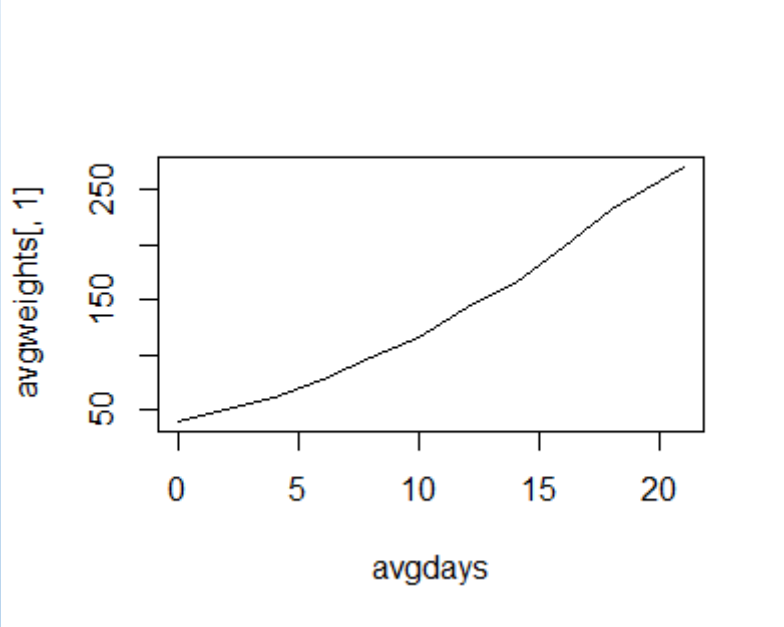
```
plot(avgdays,avgweights[,1])
```

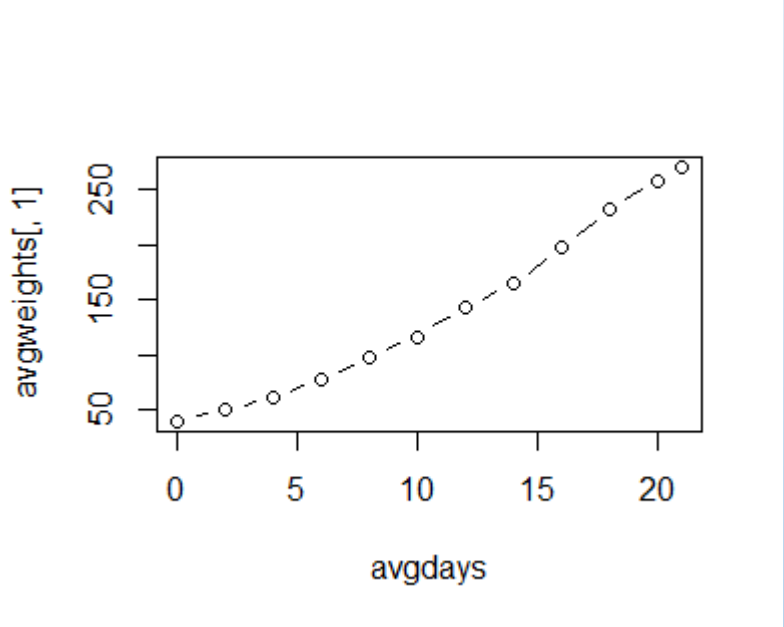
However, we wanted a line graph… Check the plot function to see how to do this, and alter your code.

```
plot(avgdays,avgweights[,1], type="l")
plot(avgdays,avgweights[,1], type="b")
```

III. Graphs: line graphs

```
plot(avgdays,avgweights[,1], type="l")
plot(avgdays,avgweights[,1], type="b")
```





III. Graphs: line graphs

Instructions

Using what we learned while making scatterplots, modify your script to do the following:

1. Add informative axes and a title
2. Draw the figure in its own window, at dimensions you are happy with (using X11 or quartz; PC or Mac)
3. Give the line a color you like.

III. Graphs: line graphs

Instructions

Using what we learned while making scatterplots, modify your script to do the following:

1. Add informative axes and a title
2. Draw the figure in its own window, at dimensions you are happy with.
3. Give the line a color you like.

```
X11(width=6,height=6)
plot(avgdays, avgweights[,1], xlab="Time
 (days)",ylab="Weight (gram)", main =
 “Chicks fed linseed”, type="l",
 col="tan")
```

III. Graphs: line graphs

Instructions

Using what we learned while making scatterplots, modify your script to do the following:

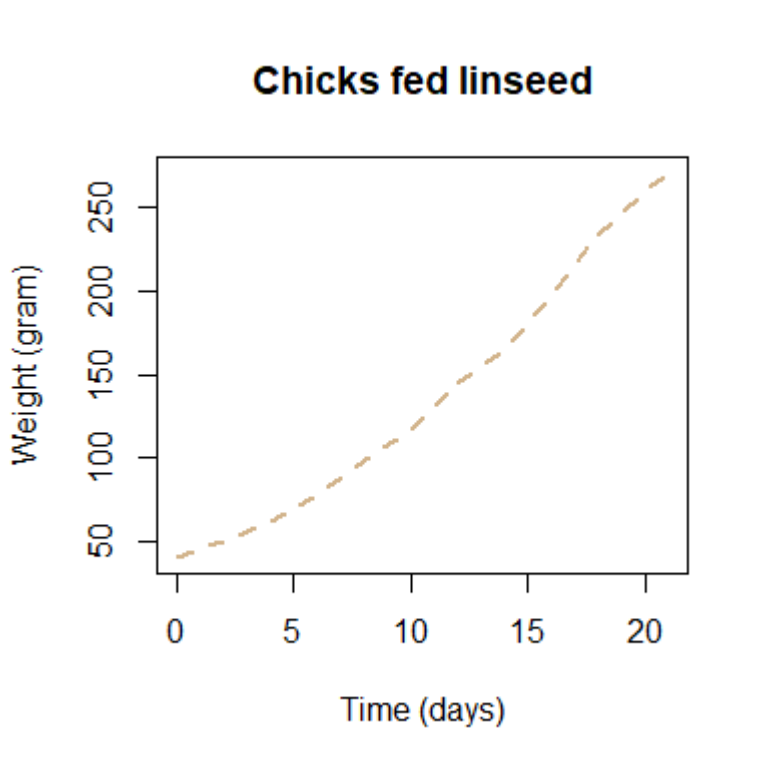
1. Add informative ax
2. Draw the figure in at dimensions you are
3. Give the line a color you like.

Use lwd to control line width, lty to control line type

```
X11(width=6,height=6)
plot(avgdays, avgweights[,1], xlab="Time
  (days)",ylab="Weight (gram)", main =
  "Chicks fed linseed", type="l",
  col="tan", lwd=2, lty=2)
```

III. Graphs: line graphs

Instructions



rned while making
ify your script to do

e a
e i
are
color you like.

Use lwd to control line width, lty to control line type

```
t=6)
weights[,1], xlab="Time
(days)",ylab="Weight (gram)", main =
"Chicks fed linseed", type="l",
col="tan", lwd=2, lty=2)
```

at

8)

III. Graphs: line graphs

The function lines will add a line to an existing plot. Format similar to plot command

```
> ?lines
```

Instructions (3 lines of code)

Add lines representing the average weight of chicks fed meatmeal, soybean and sunflower to your existing plot. Don't forget to choose a color and / or line width for your lines. Hint: will need columns 2, 3, and 4 from avgweight; extract using: [row#,col#]

III. Graphs: line graphs

Instructions (3 lines of code)

Add lines representing the average weight of chicks fed meatmeal, soybean and sunflower to your existing plot.

```
X11(width=4,height=4)
plot(avgdays, avgweights[,1], xlab="Time (days)",
 ylab="Weight (gram)", main="Chicks fed
 different diets", ylim=c(0,300), type="l",
 col="tan", lwd=2)
lines(avgdays,avgweights[,2],col="maroon", lwd=2)
lines(avgdays,avgweights[,3],col="yellowgreen",
 lwd=2)
lines(avgdays,avgweights[,4],col="yellow", lwd=2)
```

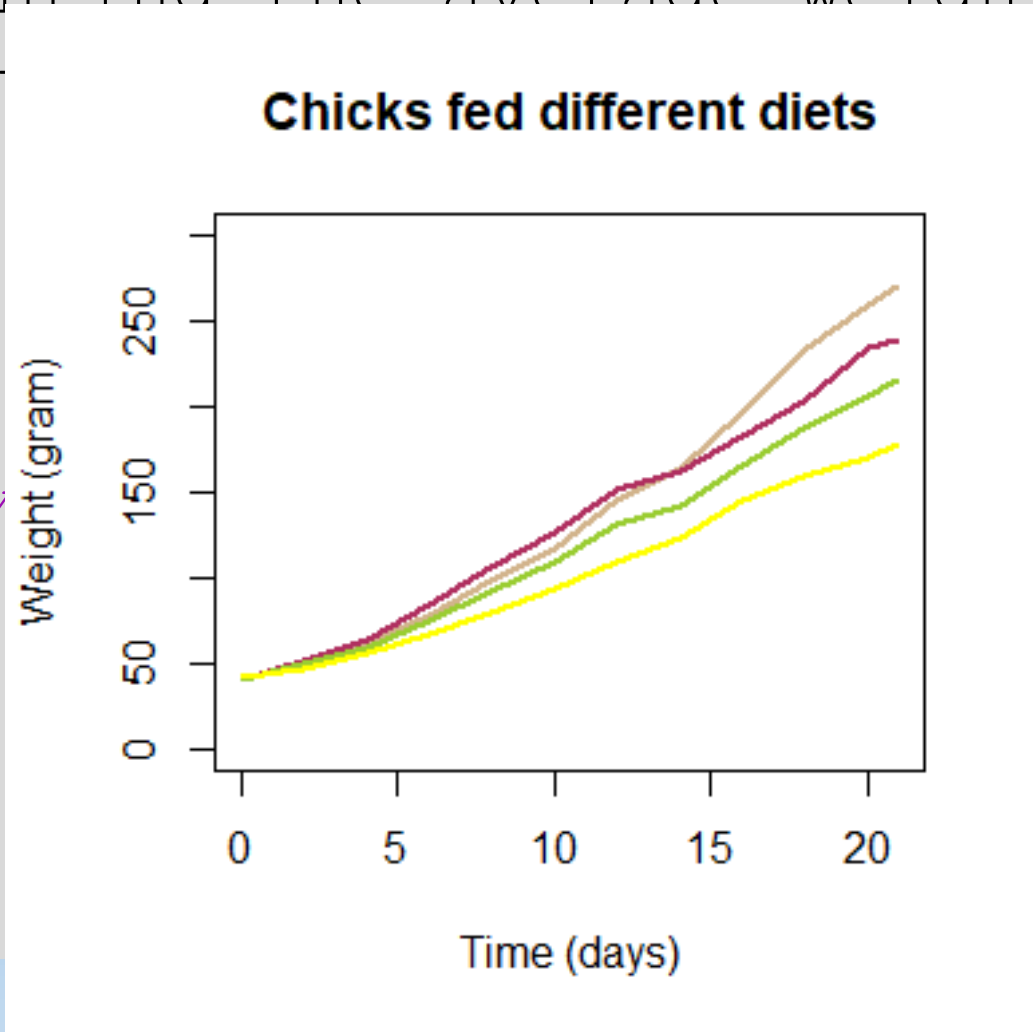
- Note, use xlim and ylim within plotting function to control axes limits

III. Graphs: line graphs

Instructions (3 lines of code)

Add lines representing the average weight of chicks fed mea
sunflower to your

```
X11(width=4,height=4
plot(avgdays, avgwei                                    ",
  ylab="Weight (gram)
  different diets", y
  col="tan", lwd=2)
lines(avgdays,avgwei                                    2)
lines(avgdays,avgwei
  lwd=2)
lines(avgdays,avgwei                                    2)
```



III. Graphs: line graphs

The function legend will add a legend to an existing plot.

```
> ?legend
```

Instruction (1 line of code)

Add a legend (top left of plot). Hints:

- Check 3rd paragraph under "Details" (in help) on how to specify legend location
- Specify vector of food types and colors using c (e.g. *col=c("tan", "maroon", "yellowgreen","yellow)*)
- Specify line width (lwd), or line type, to include lines in legend

III. Graphs: line graphs

Instruction (1 line of code)

Add a legend (top left of plot).

```
legend(x="topleft", legend=c("Linseed",
 "Meatmeal", "Soybean",
 "Sunflower"),lwd=2, col=c("tan",
 "maroon", "yellowgreen", "yellow"),
 cex=0.75)
```

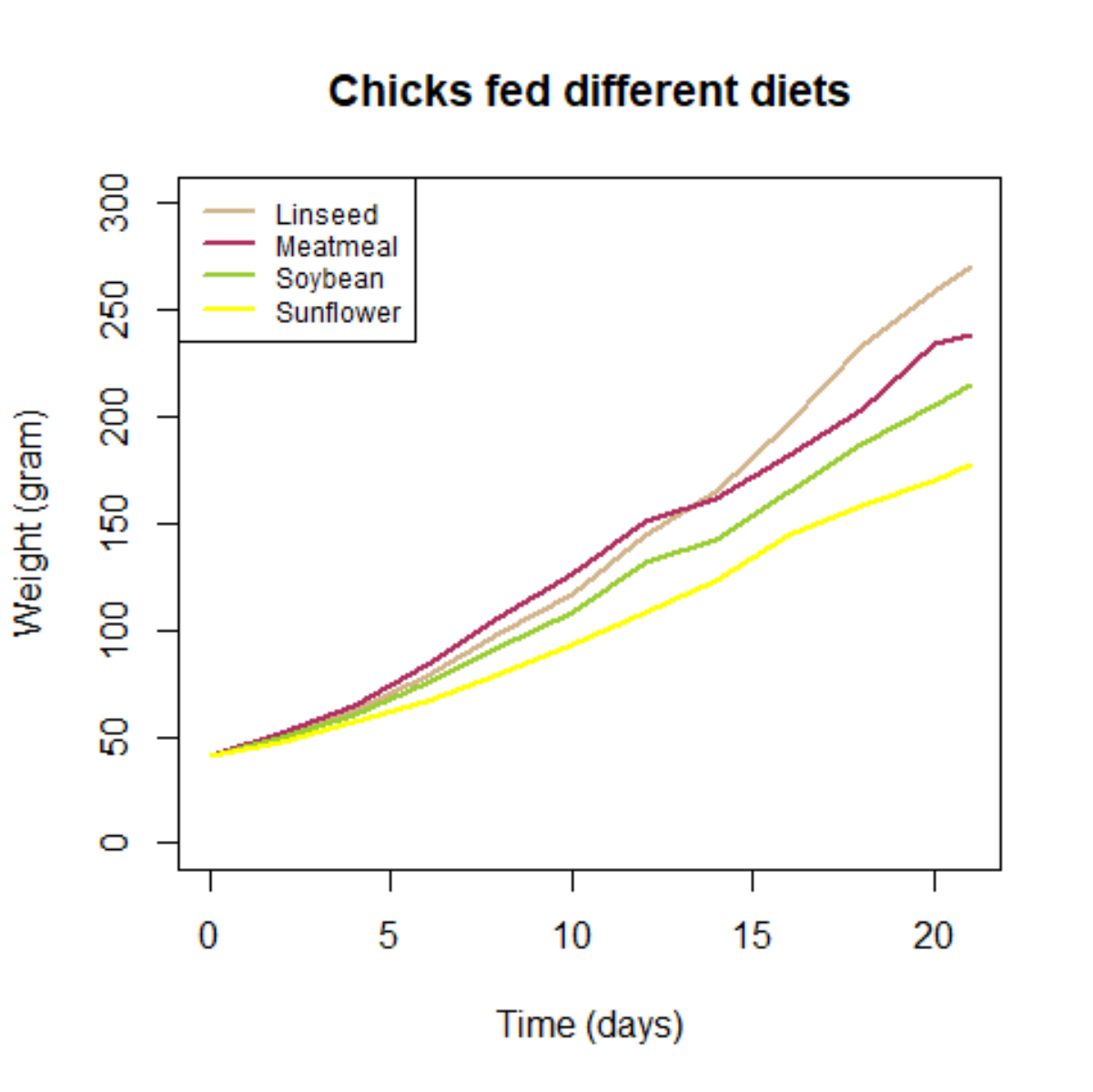
Use cex to control the size of the text in the legend

III. Graphs: line graphs

Instruction

Add a legend (

```
legend(x="topl
  "Meatmeal", "
  "Sunflower"),
  "maroon", "ye
  cex=0.75)
```



III. Graphs: boxplots

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

ChickenScript_wk3_complete.R | ChickenScript_wk3.R | ExampleScript_Graphs.R | ChickenScript

Source on Save Run Source

```
86
87
88  ##########################################
89  #Assignment 3: Boxplot
90  #Lets first subset the data into final chicken weights
91  ChickenDataFinal <- subset(ChickenData2, Time==21) #weight at 21 days
92  finalfood <- ChickenDataFinal$Protein
93  finalweight <- ChickenDataFinal$Weight
94
95  #Simple Boxplot
96  plot(finalfood,finalweight)
97
98  #plot figure to its own window
99  #Add Informative Axes, and a Title
100 #Pick some colors for the bars
101 X11(height=5,width=5)
102
```

94:1 (Untitled)

Highlight these lines of code and click on 'Run'

Console C:/Users/janneke/Dropbox/Teaching&Mentor

```
> ##################################
> #Assignment 3: Boxplot
> #Lets first subset the data into fi
> ChickenDataFinal <- subset(ChickenD
> finalfood <- ChickenDataFinal$Protein
> finalweight <- ChickenDataFinal$Weight
```

Instructions

1. Run the first few lines in your script under assignment 3

III. Graphs: boxplots

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

ChickenScript_wk3_complete.R | ChickenScript_wk3.R | ExampleScript

Source on Save Run

```
67
68  ##########################################
69  #Assignment 2: Boxplot
70  #Lets first subset the data into final chicken weights
71  ChickenDataFinal <- subset(ChickenData2, Time==21) #weigl
72  finalfood <- ChickenDataFinal$Protein
73  finalweight <- ChickenDataFinal$Weight
74
75  #Simple Boxplot
76
77
```

68:1 (Untitled) R Script

Add to script here
Click on 'Run'

Console C:/Users/janneke/Dropbox/Teaching&Mentorir

```
> ##########################################
> #Assignment 2: Boxplot
> #Lets first subset the data into fin
> ChickenDataFinal <- subset(ChickenDa
21 days
> finalfood <- ChickenDataFinal$Protei
> finalweight <- ChickenDataFinal$Weigl
>
```

Instructions

Add a line of code to plot finalweight vs. final food

III. Graphs: boxplots

Instructions

Add a line of code to plot finalweight vs. final food.

```
plot(finalfood,finalweight
```

Because food is considered a 'factor' (i.e. categorical explanatory variable), the plot command automatically creates a boxplot

III. Graphs: boxplots

Instructions

Using what we learned while making scatterplots and line graphs, modify your script to do the following:

1. Add informative axes and a title
2. Draw the figure in its own window, at dimensions you are happy with (using X11 or quartz; PC or Mac)
3. Give the bars colors (hint, use the ‘c’ command to create a vector of 4 colors, don’t forget the “; e.g. “red”)

III. Graphs: boxplots

Instructions

Using what we learned while making a scatterplot, modify your script to do the following:

1. Add informative axes and a title
2. Draw the figure in its own window
3. Give the bars colors

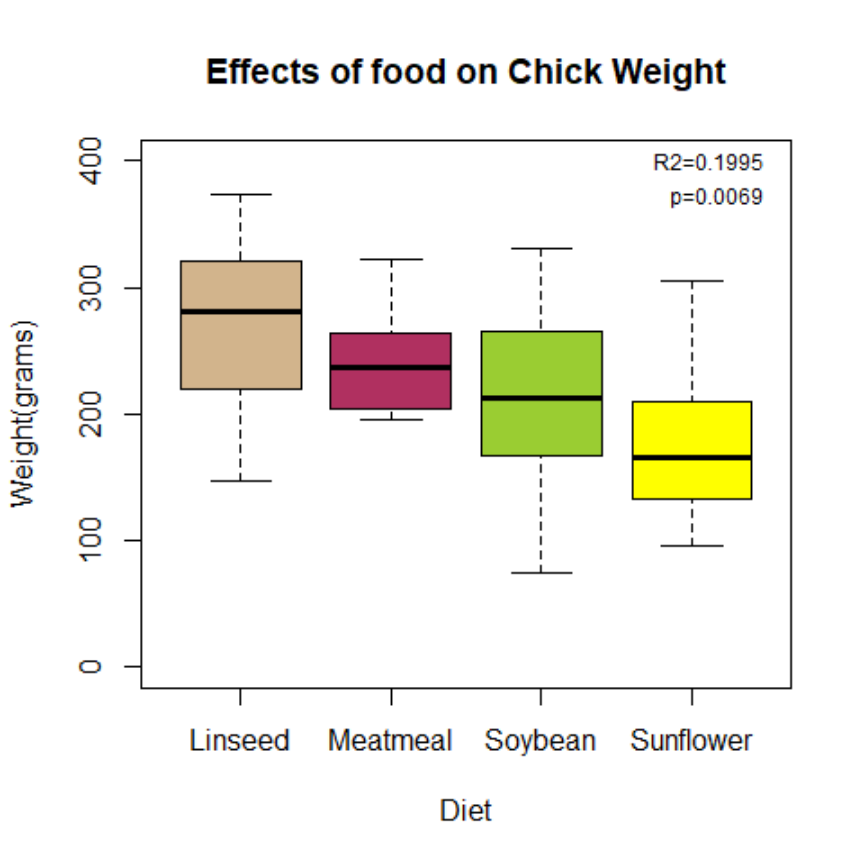
```
X11(height=5,width=5)

plot(finalfood, finalweight, xlab="Diet",
ylab="Weight(grams)", main="Effects of
food on Chick Weight", col=c("tan",
"maroon", "yellowgreen", "yellow"))
```

III. Graphs: boxplots

```
X11(height=5,width=5)

plot(finalfood,
finalweight,
xlab="Diet",
ylab="Weight(grams)",
main="Effects of food
on Chick Weight",
col=c("tan", "maroon",
"yellowgreen",
"yellow"))
```



III. Graphs: boxplots

Instructions

We should probably make sure the y axis starts at 0. Modify your code as follows:

```
plot(finalfood, finalweight, xlab="Diet",
ylab="Weight(grams)", main="Effects of
food on Chick Weight", ylim=c(0,400),
col=c("tan", "maroon", "yellowgreen",
"yellow"))
```

- Note, use xlim and ylim within plotting function to control axes limits

III. Graphs: boxplots

The function mtext will add text in the margin of a plot. Check it out!

```
> ?mtext
```

Instructions (2 lines of code)

1. Run a linear model to find the relationship between food and chick weight (use function lm, and summary to examine the output).
2. Add the (adjusted) r^2 and p-value in the upper right margin of the plot using the function mtext. Hint: play around with ‘side=‘, ‘line=‘, and ‘adj=‘ within the command mtext

III. Graphs: boxplots

Instructions (2 lines of code)

1. Run a linear model to find the relations
   weight (u

```
finaltest <
summary(fin
```

Console C:/Users/janneke/Dropbox/Teaching&Mentoring/NZ_visit/R_StatsWorkshops/WK3_Gra

```
> summary(finaltest)

Call:
lm(formula = finalweight ~ finalfood)

Residuals:
     Min       1Q   Median       3Q      Max 
-140.700  -39.700   -1.556   37.250  127.250 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)          270.30      20.23  13.360  < 2e-16 ***
finalfoodMeatmeal    -31.74      29.40  -1.080  0.28653    
finalfoodSoybean     -55.60      28.61  -1.943  0.05889 .  
finalfoodSunflower   -92.55      25.79  -3.588  0.00088 ***
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 63.98 on 41 degrees of freedom
Multiple R-squared:  0.2541,	Adjusted R-squared:  0.1995 
F-statistic: 4.655 on 3 and 41 DF,  p-value: 0.006858
```

III. Graphs: boxplots

Instructions (2 lines of code)

1. Run a linear model to find the relationship between food and chick weight (use function lm).

```
finaltest <- lm(finalweight~finalfood)
summary(finaltest)
```

2. Add the (adjusted) r^2 and p-value in the upper right margin of the plot.

```
mtext("R2=0.1995",side=3, line=-1,
  adj=0.95, cex=0.75)
mtext("p=0.0069", side=3,line=-2,
  adj=0.95, cex=0.75)
```

III. Graphs: boxplots

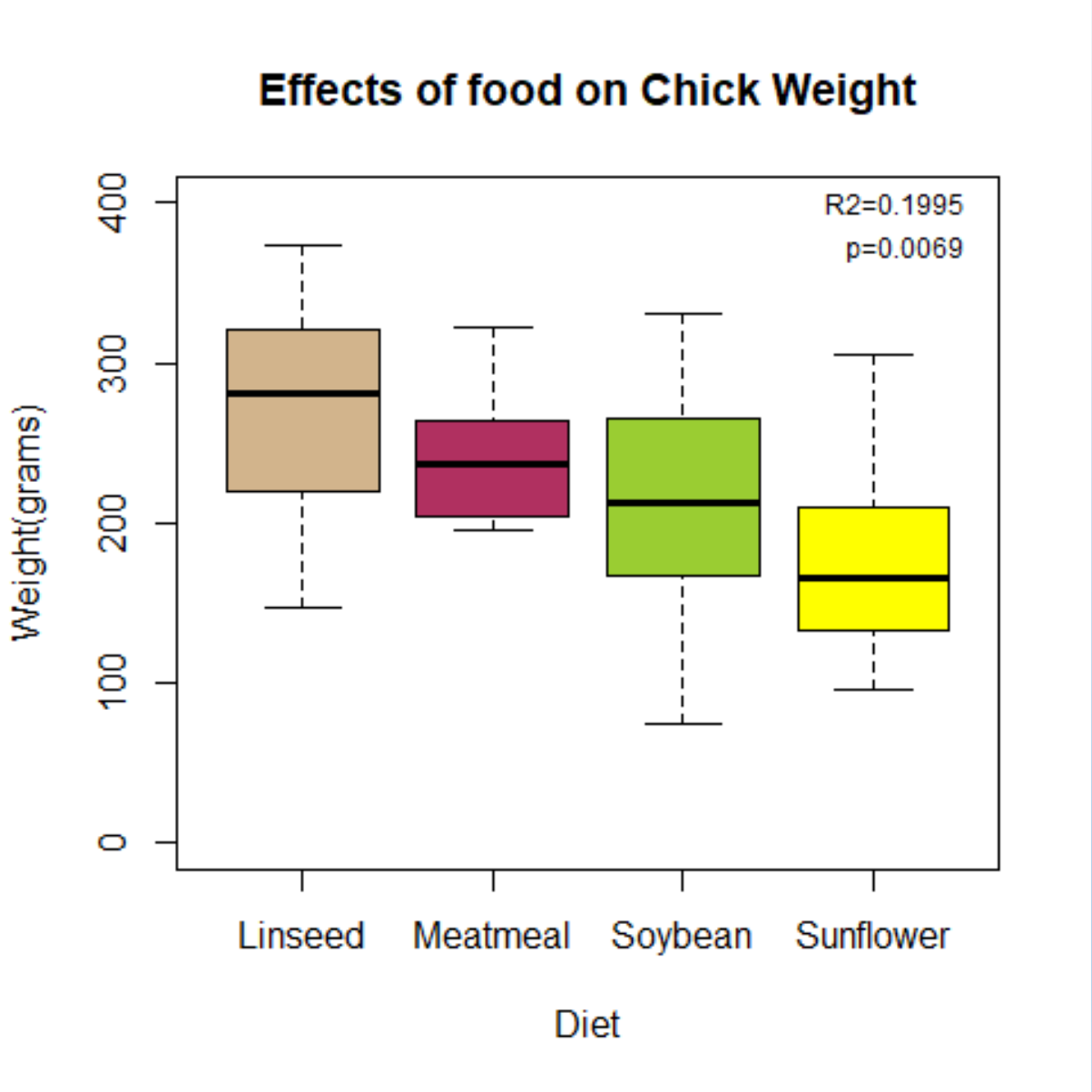
Instructions

1. Run a linear
   relationship
   weight (use

```
finaltest <- lm
summary(finalte
```

2. Add the (adj
   the upper ri

```
mtext("R2=0.199
  adj=0.95, cex=
mtext("p=0.0069
  adj=0.95, cex=
```



III. Graphs: boxplots

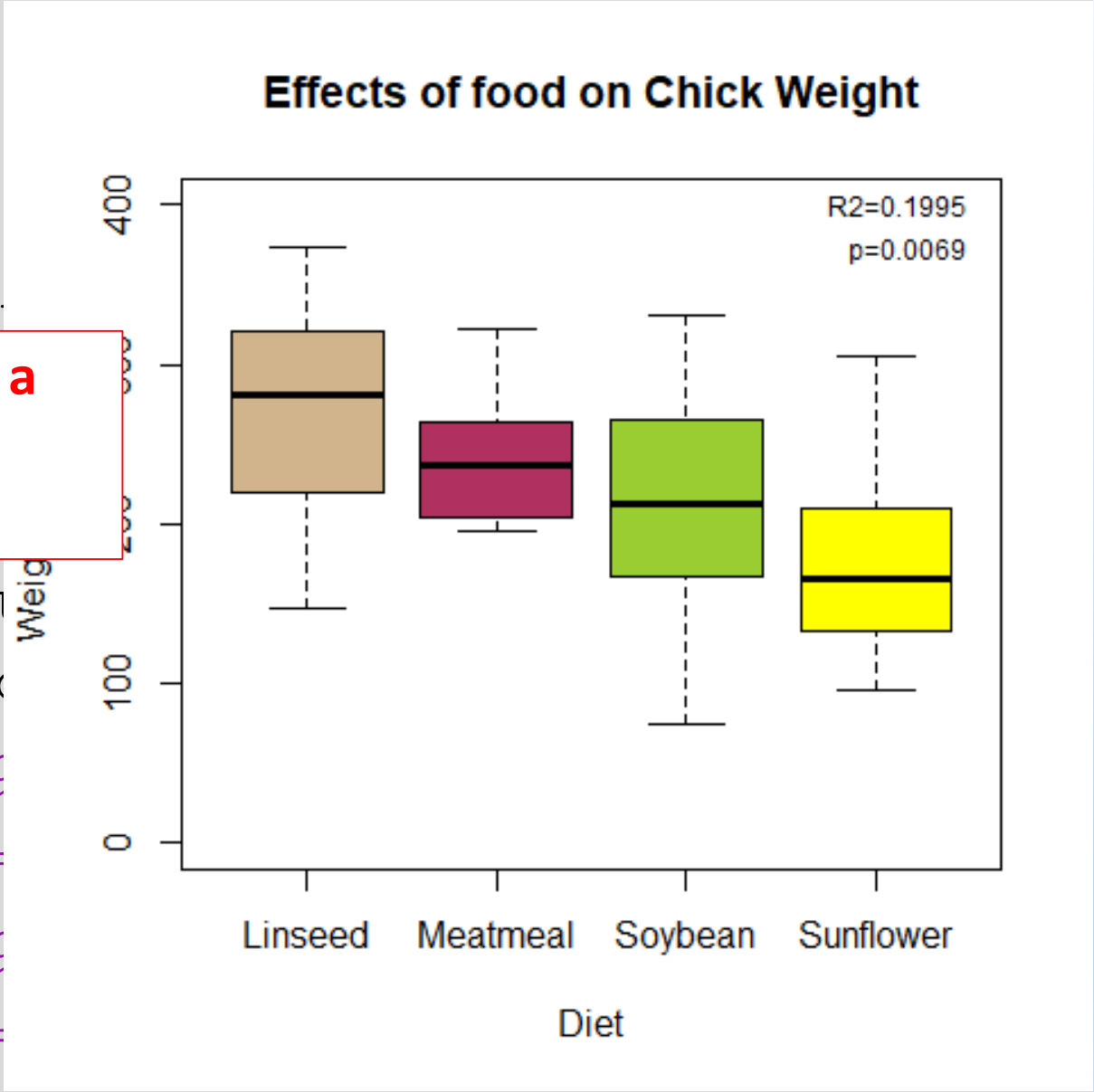
Instructions

1. Run a linear
   relationship
   weight (use

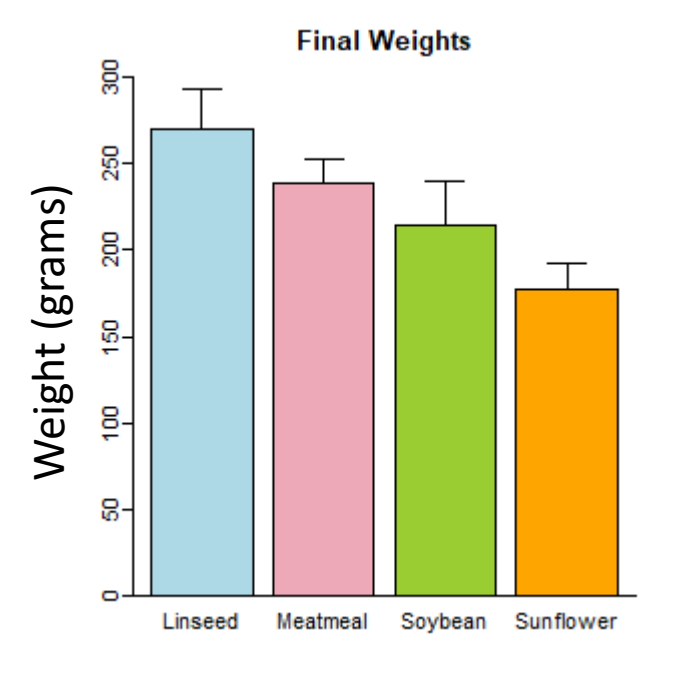
Set adj to not quite 1 for a buffer between the plot edge and text'

2. Add the (adj
   the upper ri

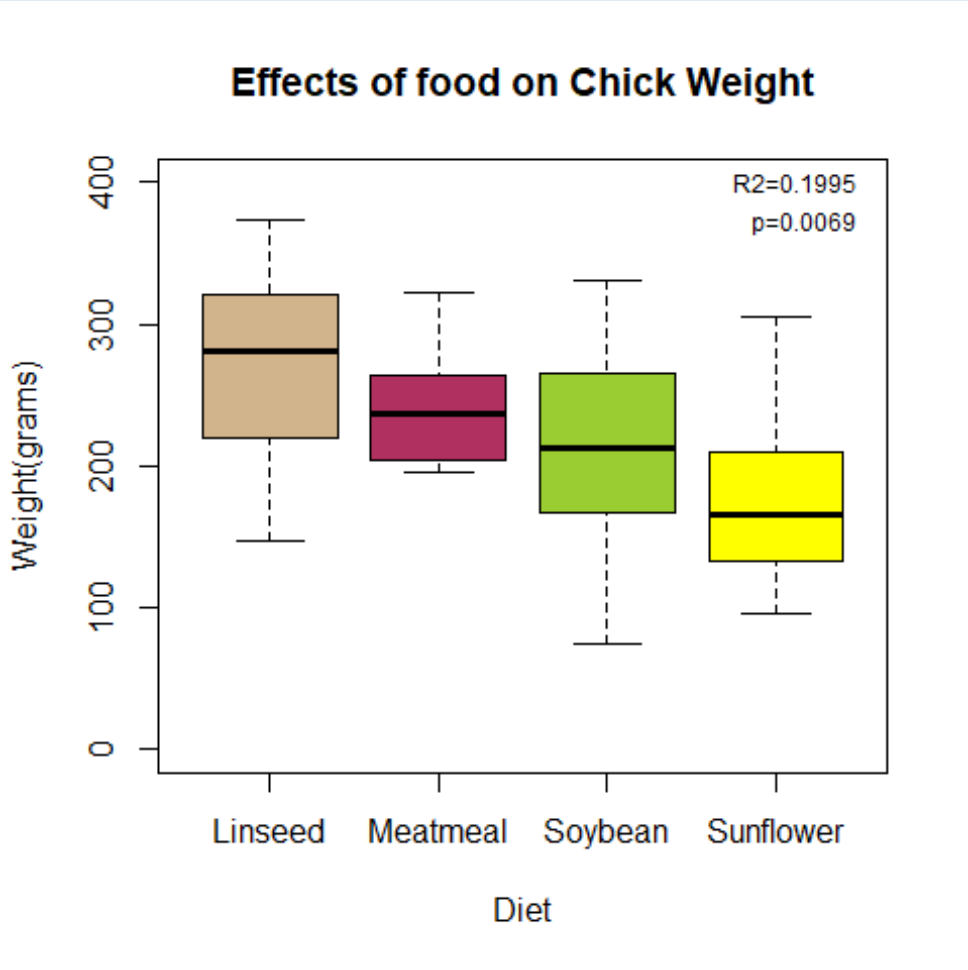
```
mtext("R2=0.199
  adj=0.95, cex=
mtext("p=0.0069
  adj=0.95, cex=
```



III. Graphs: boxplots vs. barplots



- Use barplot, and arrows to add confidence intervals

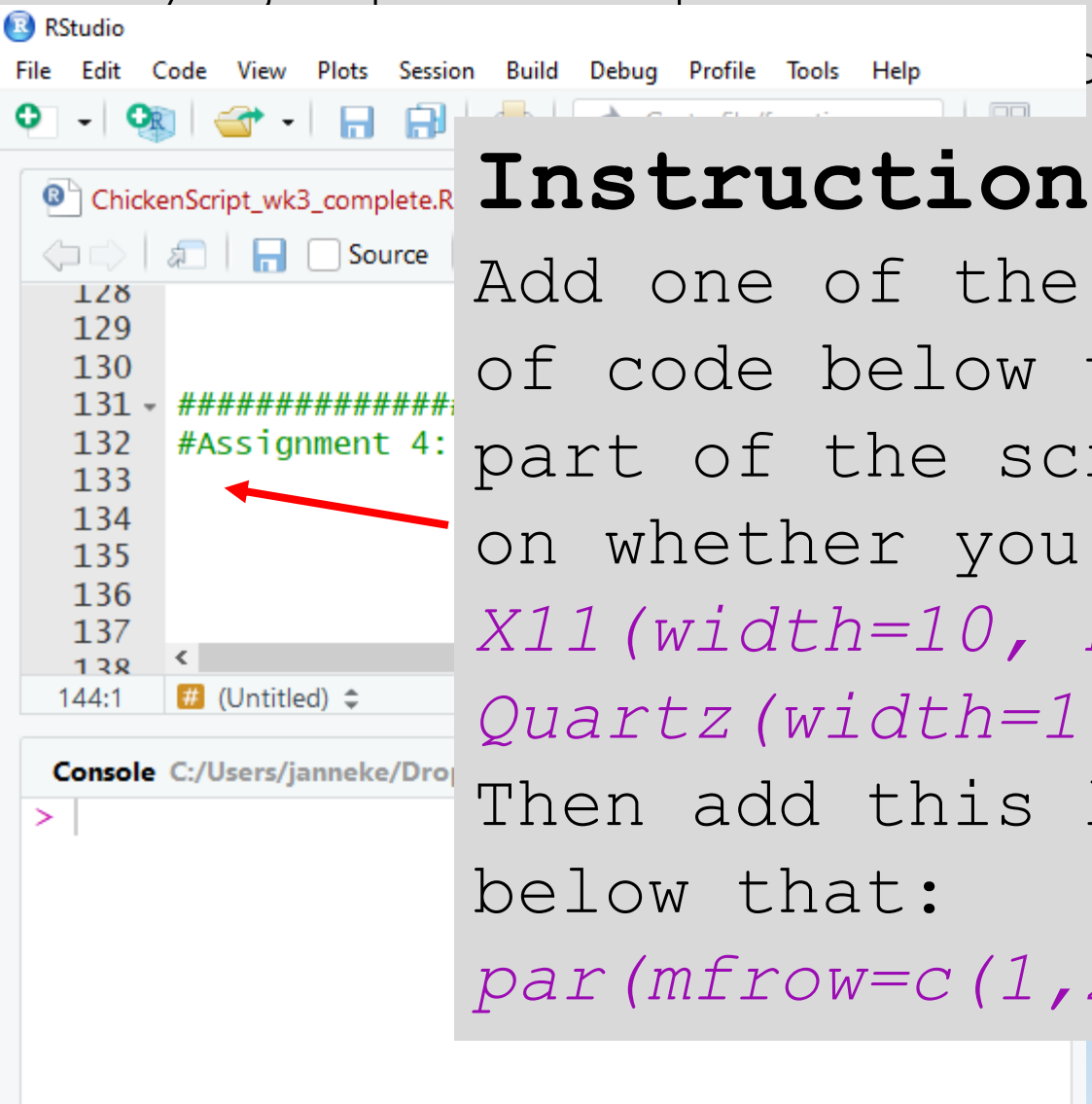


III. Graphs: multi-panel graphs

```
The function par can be used to control
many graphical parameters (WARNING - can
reset default graphical parameters). We
will use it to create a multi-panel figure
> ?par
```

III. Graphs: multi-panel graphs

The function par can be used to control many graphical parameters (WARNING – can parameters). We



Instructions

Add one of the following lines of code below the Assignment 4 part of the script, depending on whether you have a PC/Mac:

```
X11(width=10, height=5) #PC
Quartz(width=10, height=5) #Mac
```

Then add this line of code below that:

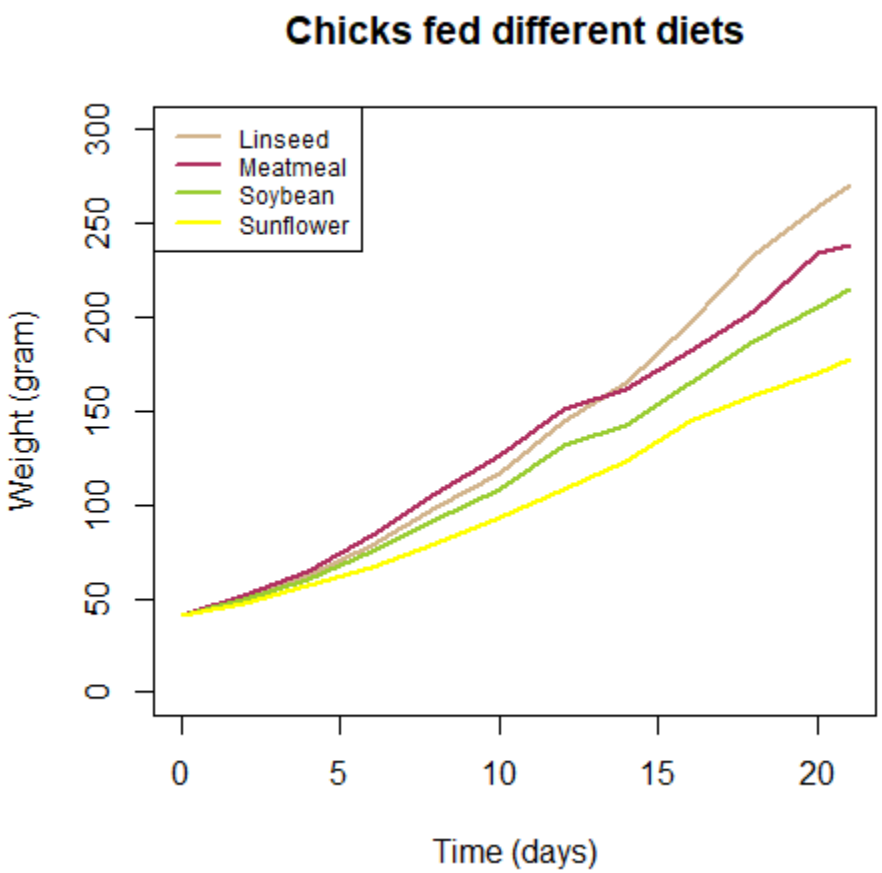
```
par(mfrow=c(1,2))
```

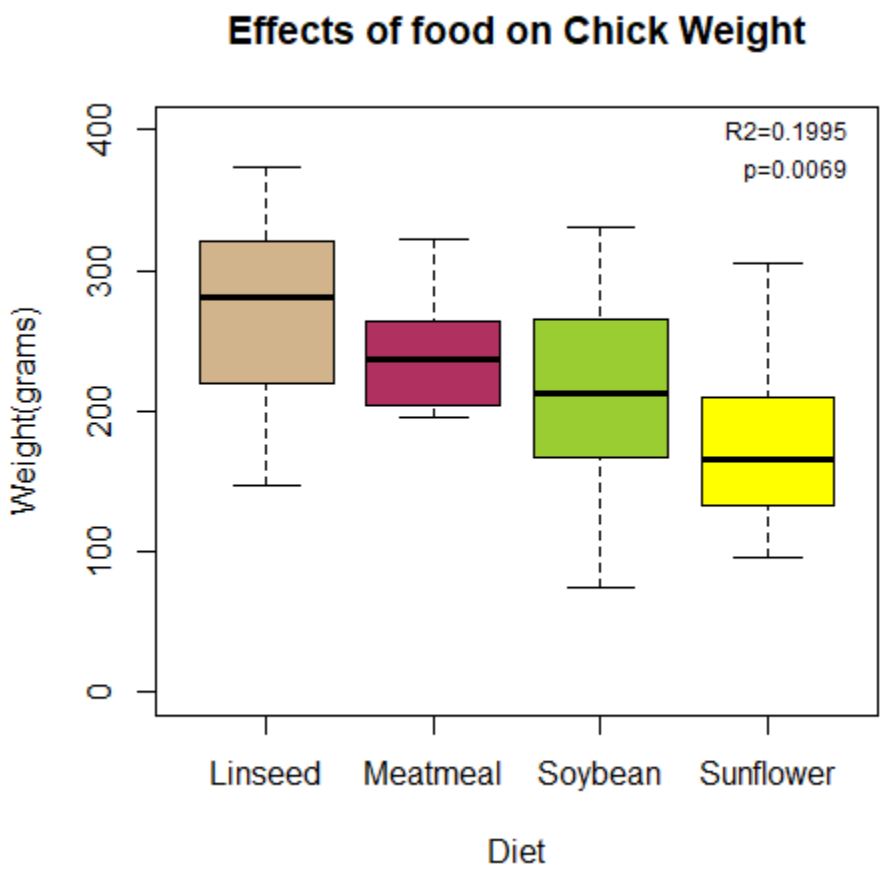
III. Graphs: multi-panel plots

Instructions

Cut and paste the blocks of code you used to create the line graph (assignment 2) and the box plot (assignment 3) below the *X11 / quartz* and *par* lines of code you just wrote (exclude the *X11 / quartz* code you wrote in those blocks!). Then run all the lines of code starting with *X11 / quartz*.

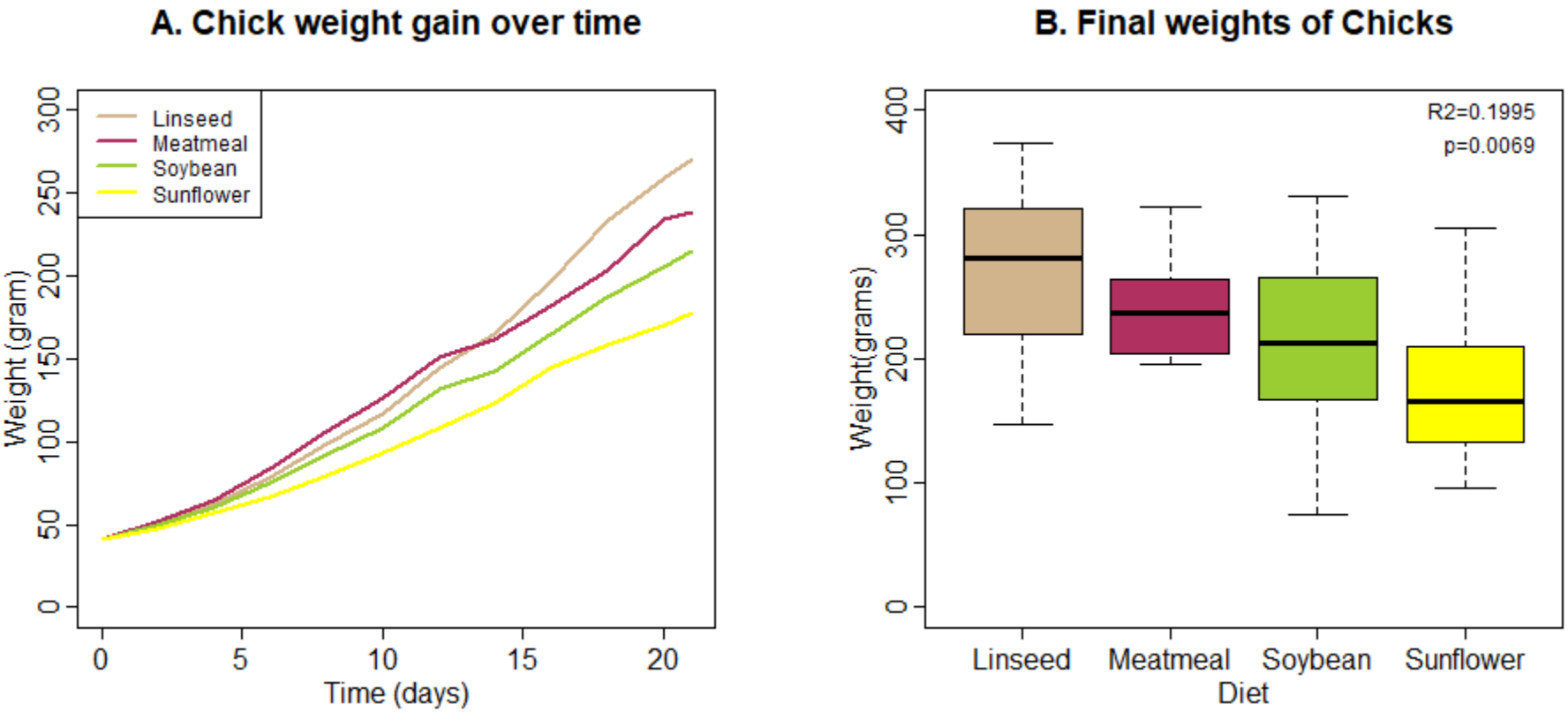
III. Graphs: multi-panel plots





- Can add A and B (as in Figure 1A, 1B), using mtext (or add to title). Also better titles!

III. Graphs: multi-panel plots

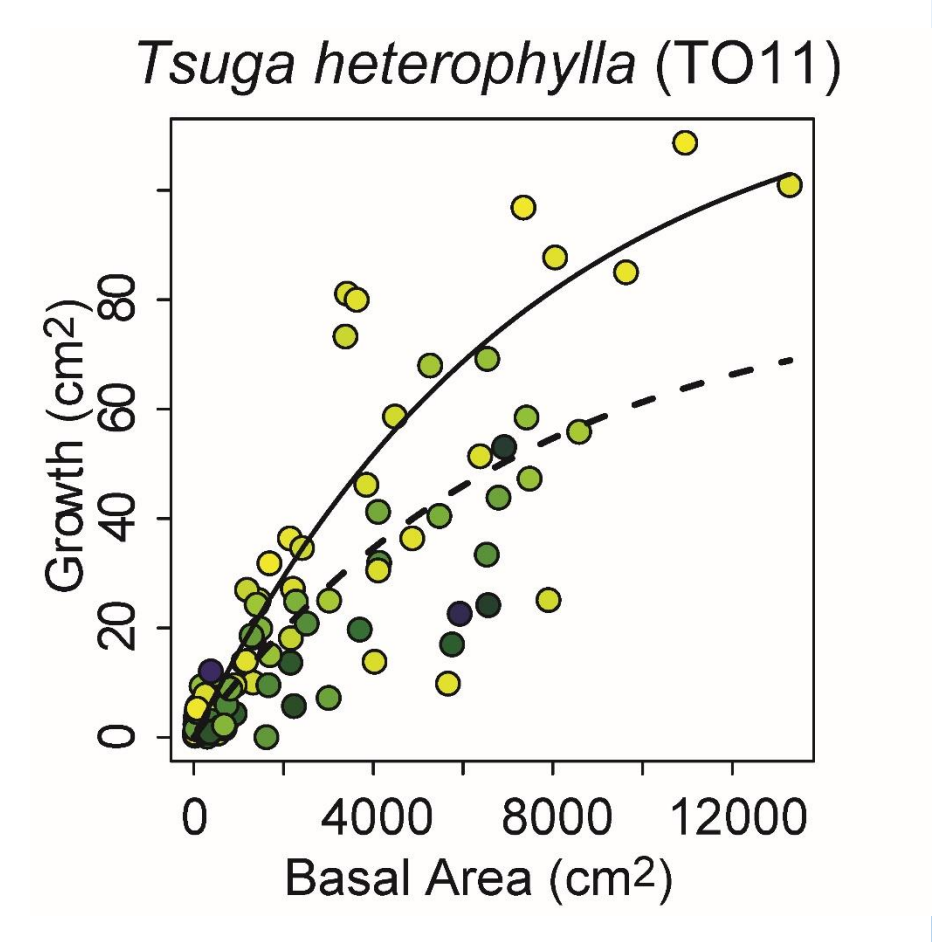


I like to use par for tighter spacing on placement of axes labels / numbers and shorter tick marks:

```
par(mfrow=c(1,2),mgp=c(1.4,0.4,0),tck=-0.02)
```

III. Graphs: R has many more capabilities!

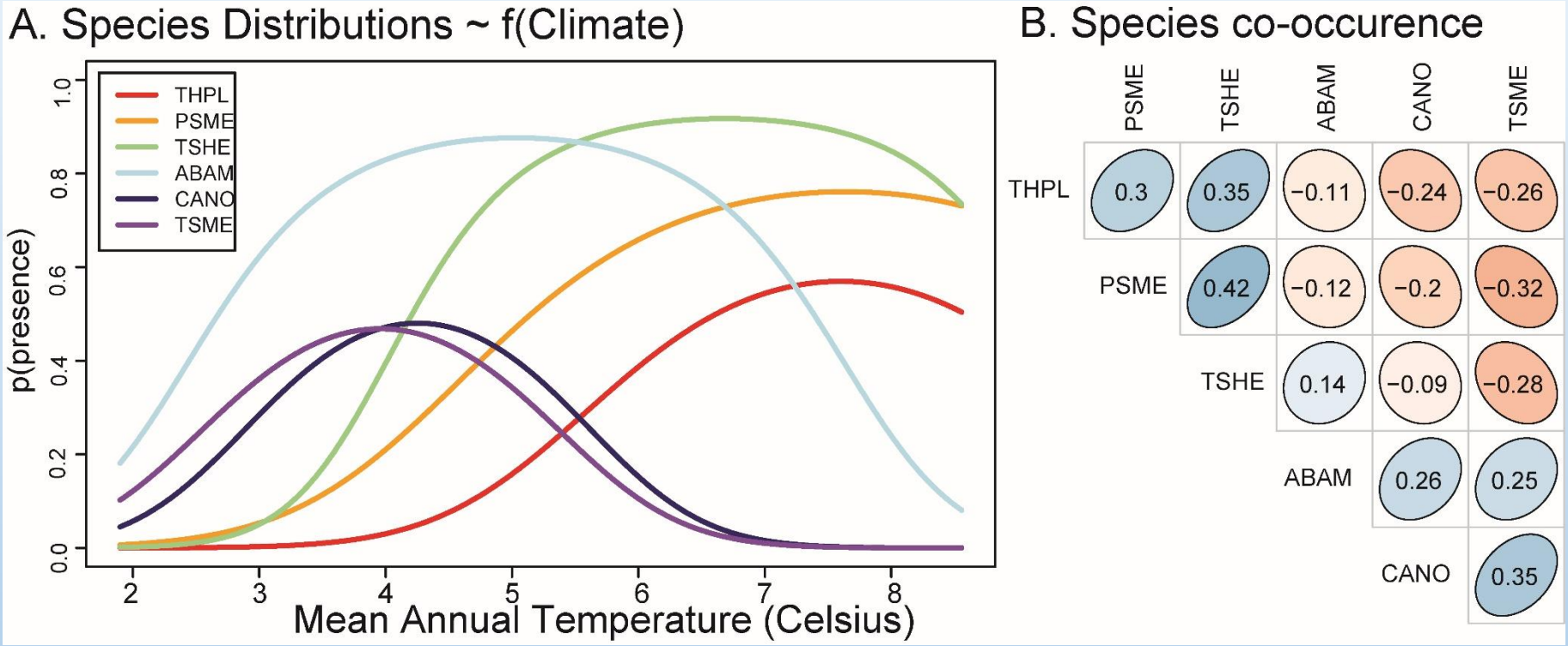
Color ramps for other information



Anderegg 2014

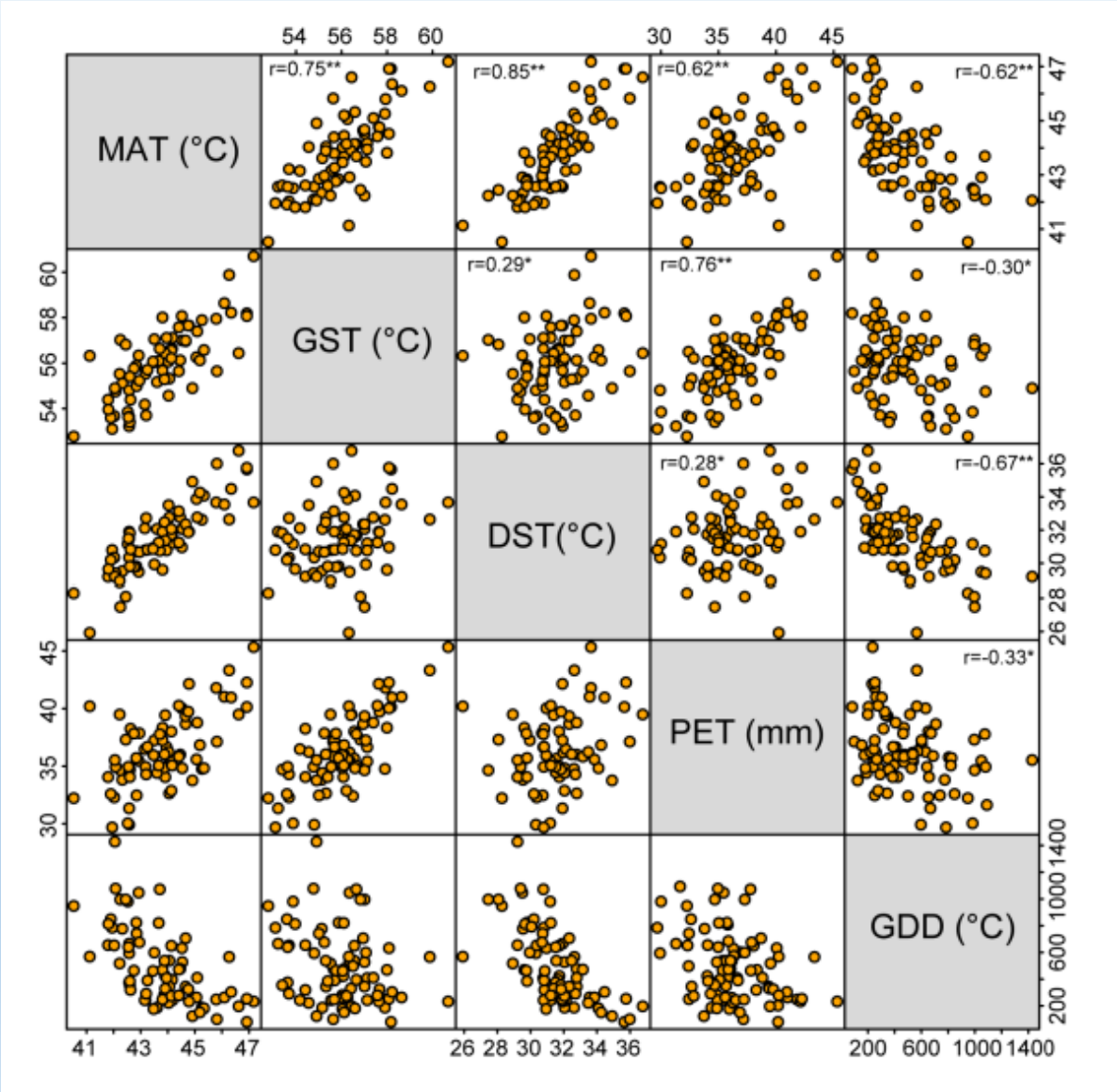
III. Graphs: R has many more capabilities!

Correlations



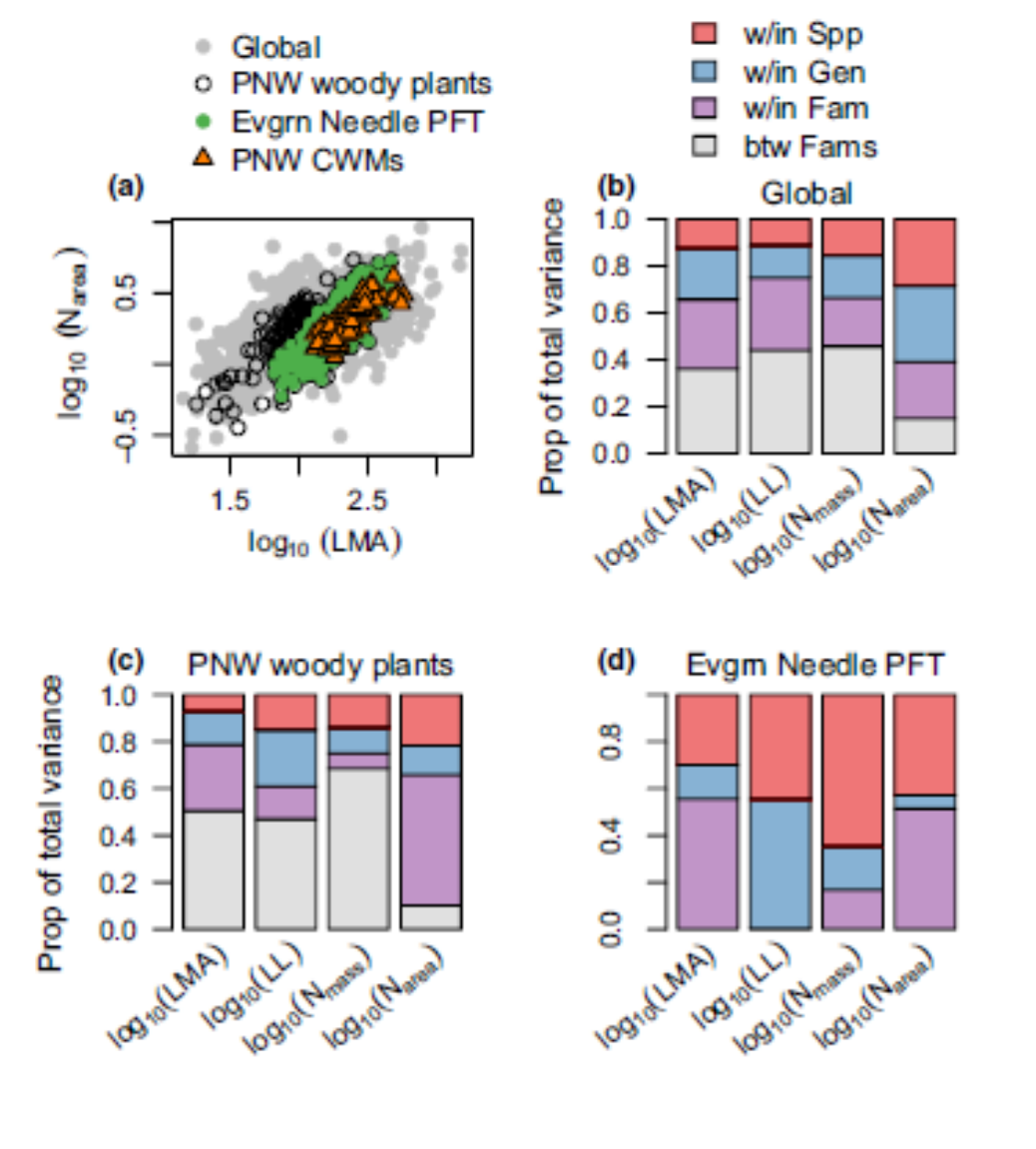
III. Graphs: R has many more capabilities!

More correlations (check function ‘pairs’)



III. Graphs: R has many more capabilities!

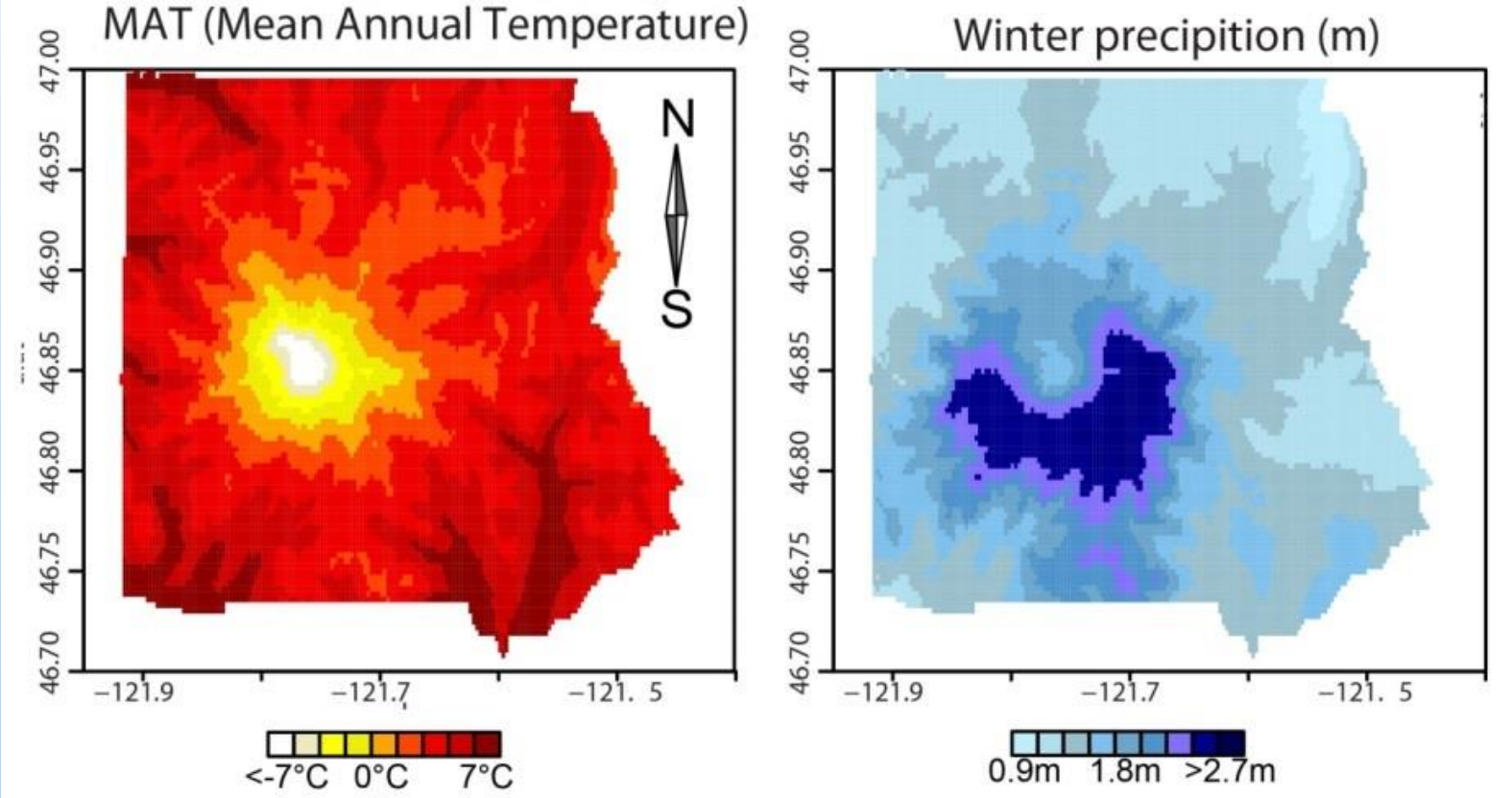
Multi-panel colored graphs



op 3 (29/03/2018)

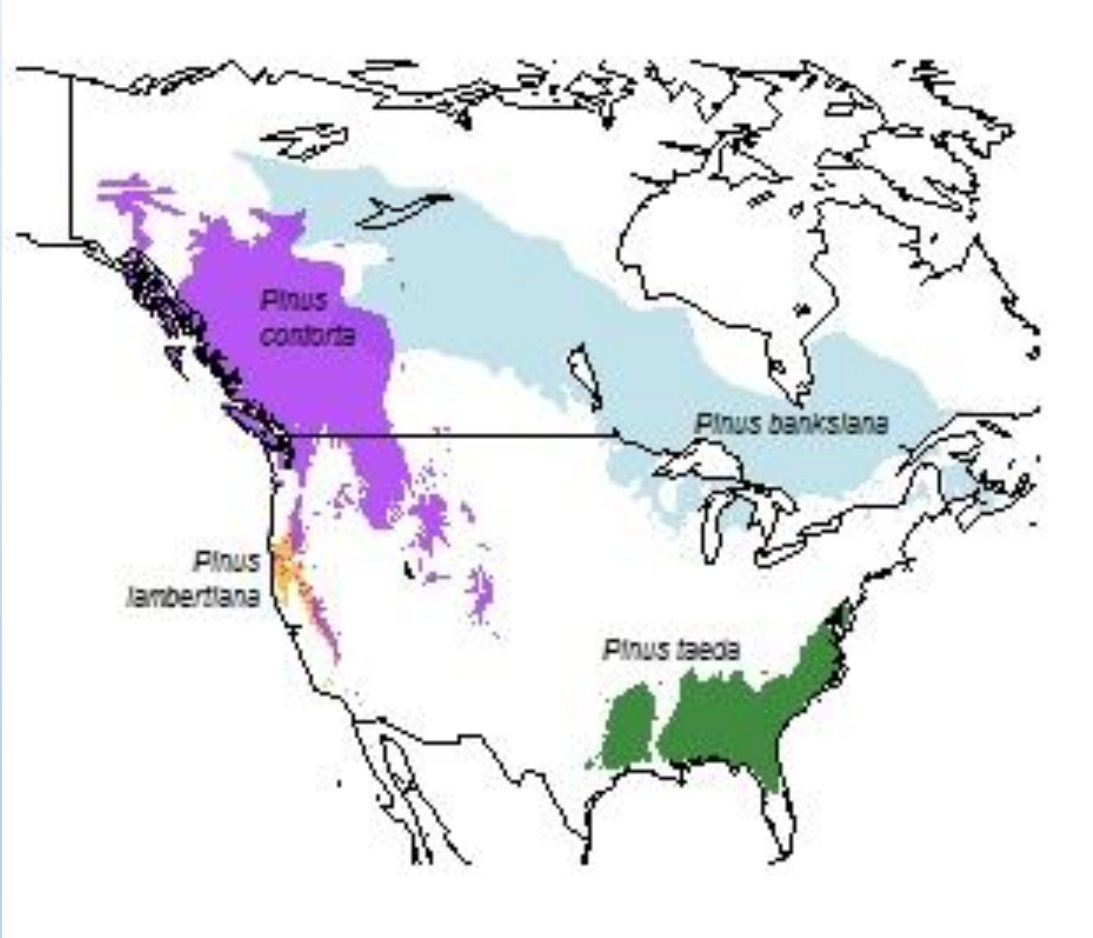
III. Graphs: R has many more capabilities!

Maps / shading

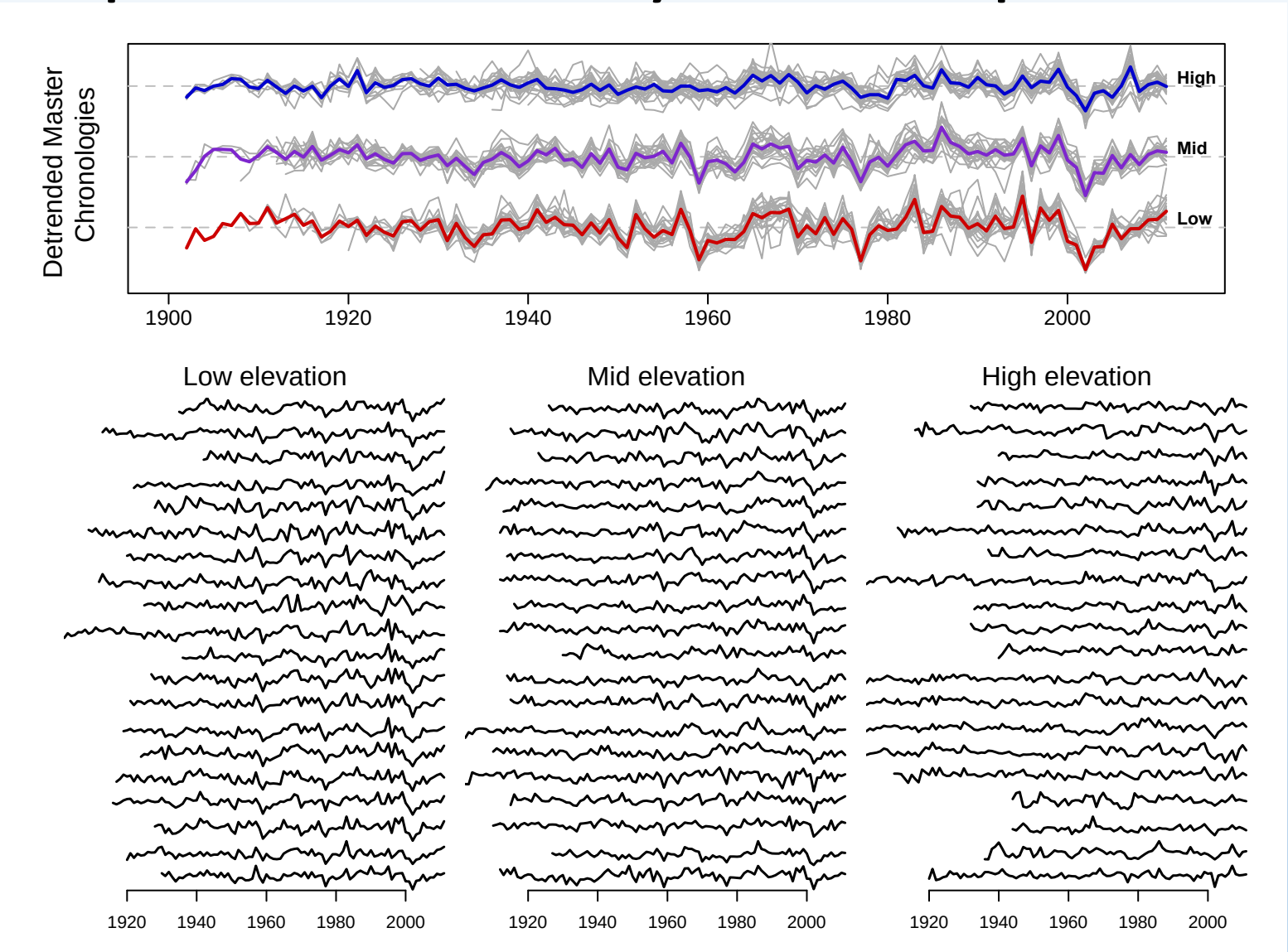


III. Graphs: R has many more capabilities!

Shape files and maps



III. Graphs: R has many more capabilities!



Anderegg 2014

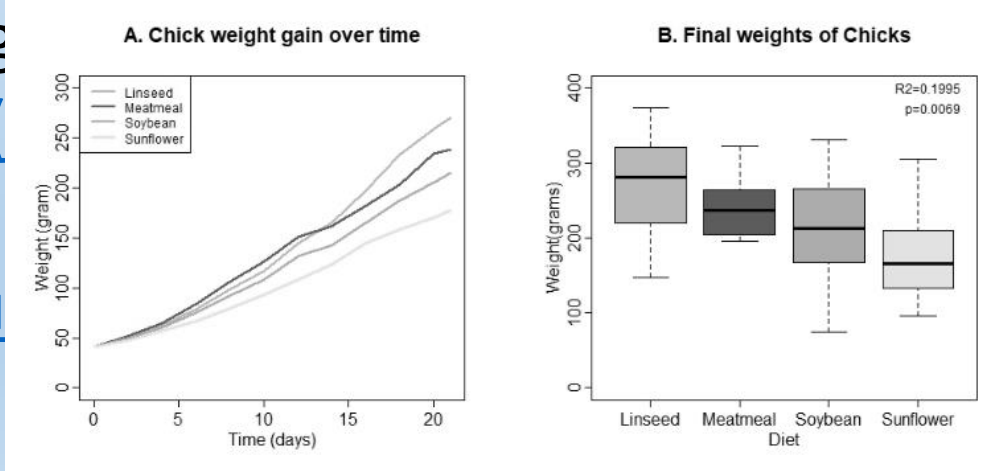
Retweeted by Tufte!

III. Graphs: Want more practice?

1. Work on your own data (I can consult)
2. Work with nutnet data (NutnetInstructions.pdf)

IV. Additional Resources

- In press data visualization book by Kiran Healy (Princeton Press – see here)
- R chart gallery – see here
- For more practice, try this course on how to create plots in R from DataCamp
- Many resources online to create colorblind friendly graphs, including this one (upload a figure and see what it would look like to those with different colorblindness)
- Wickhams graphics packag
  graphics for data analysis (
- Jeffrey Heer's website
- Classic book by Tufte (Visu

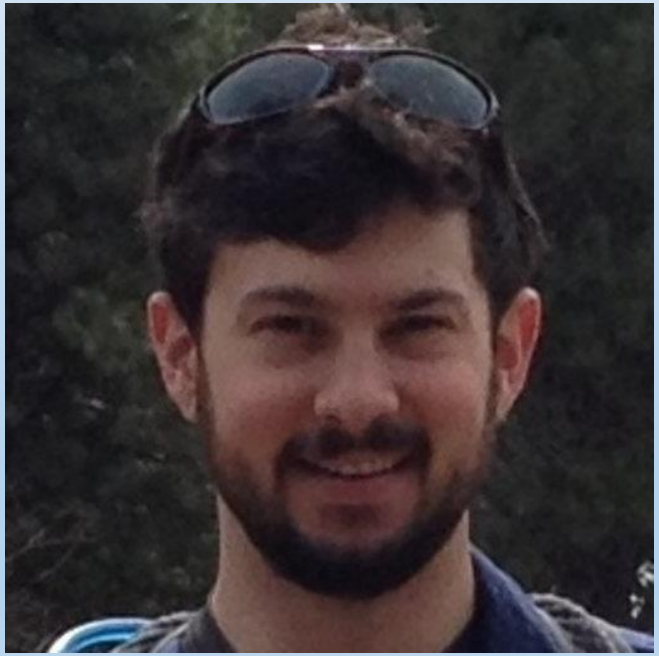


Any others? Please send

Acknowledgments

Plus all pirated resources
(hopefully mostly cited)

Trevor Branch
UW SAFS – R course (SAFS 552, 553)



Clay Wright
UW Biology, R course (SAFS 552, 553) & Other

Thank you for coming!

A favor: Please respond to survey when I send it out!