An introduction to ggplot

Federico Lopez

October 23, 2015

Contents

Install the necessary packages

```
install.packages("RCurl")
install.packages("ggplot2")
install.packages("gridExtra")
install.packages("ggmap")
install.packages("maps")
install.packages("rdryad")
install.packages("rgbif")
install.packages("plyr")
install.packages("rWBclimate")
install.packages("cati")
```

```
# Load libraries
library(RCurl) # Download URLs
library(ggplot2) # Create plots
library(gridExtra)
library(ggmap) # Create maps
```

```r
library(maps) # For map data
library(rdryad) # Retrieve data sets from dryad
library(rgbif) # Retrieve occurence data from gbif
library(plyr) # Split data frames and apply functions
library(rWBclimate) # Retrieve World Bank climate data
library(cati) # For morphological data from Darwin's finches
```

1 Getting started with qplot

ggplot2 is a powerful plotting system for R. ggplot2 documentation is available at docs.ggplot2.org. `qplot` is the basic plotting function in the `ggplot2` package. `qplot` is similar to the base R function `plot`; however, you cannot pass any type of R object to `qplot`.
Basic usage: `qplot(x, y, data...)`, where data is the data frame to use and `...` means other aesthetics passed for each layer.

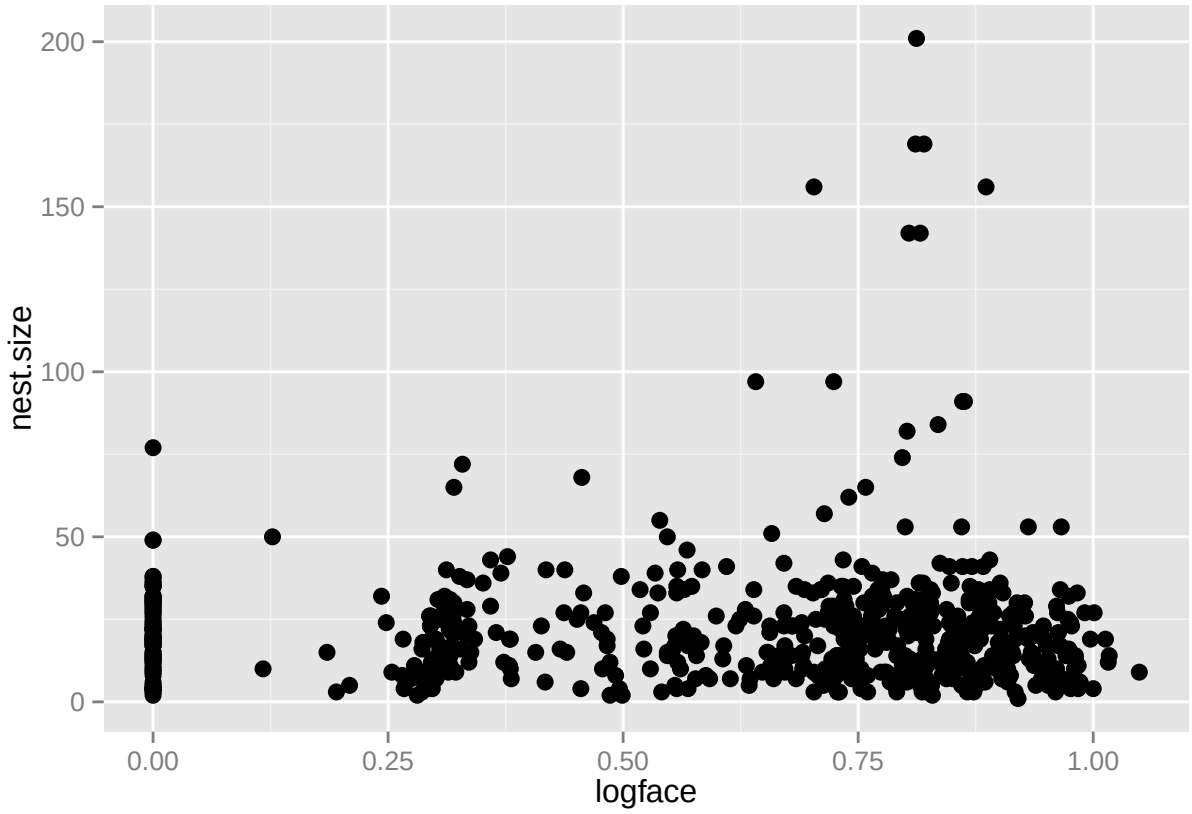
1.1 Using qplot to create scatterplots, histograms and boxplots

1.1.1 Scatterplots

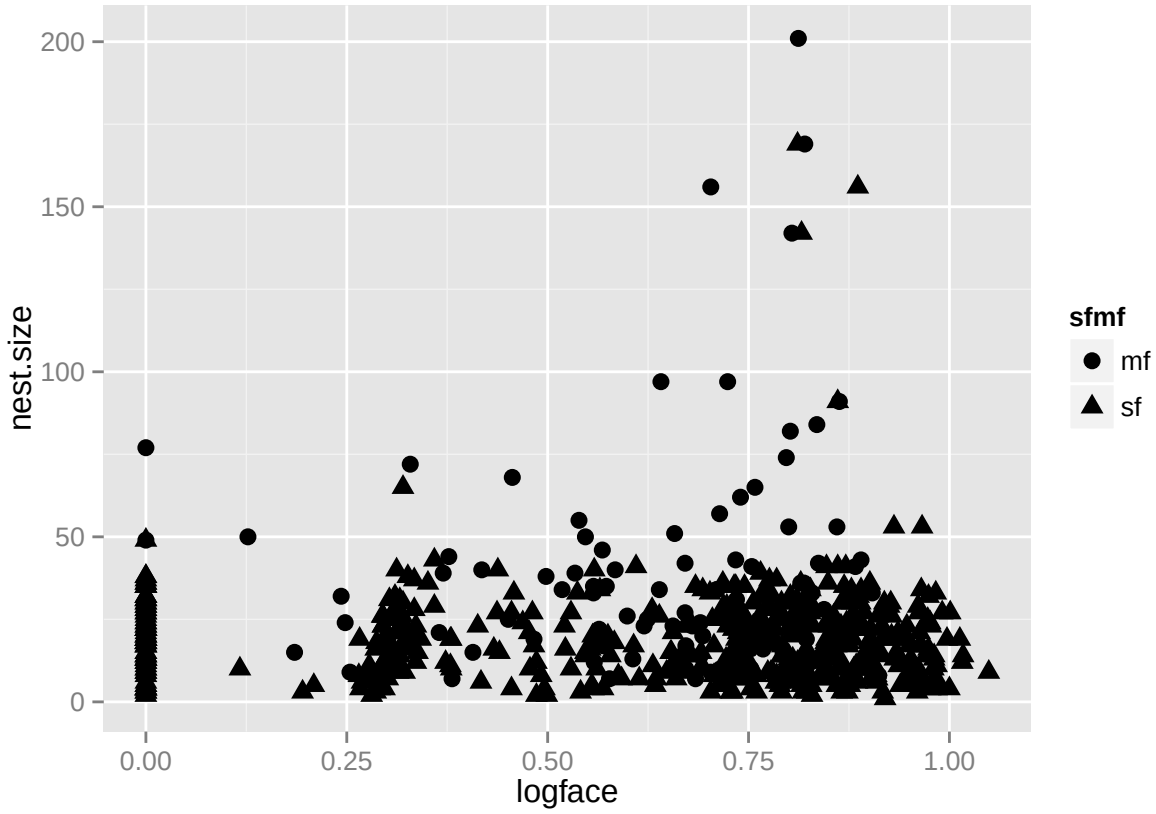
```r
# Retrieve a published dataset
# Data source: http://onlinelibrary.wiley.com/doi/10.1111/evo.12793/abstract
# http://datadryad.org/resource/doi:10.5061/dryad.0398p
# Copy the file's raw GitHub URL
polistesdat.url <- "https://raw.githubusercontent.com/flopezo/atd/master/Tibbetts_et_al_2015_data.csv"
polistesdat.url <- getURL(polistesdat.url)
polistes.data <- read.csv(textConnection(polistesdat.url))
# Show the variables included in the dataset
summary(polistes.data)
```

```
##       year      day.collected       weight        sfmf
##  Min.   :2011   Min.   :-6.00   Min.   :0.05100   mf:106
##  1st Qu.:2011   1st Qu.: 1.00   1st Qu.:0.08400   sf:505
##  Median :2011   Median :25.00   Median :0.09600
##  Mean   :2011   Mean   :20.96   Mean   :0.09692
##  3rd Qu.:2012   3rd Qu.:31.00   3rd Qu.:0.10900
##  Max.   :2012   Max.   :53.00   Max.   :0.15300
##    nest.size         logface
##  Min.   :  1.00   Min.   :0.0000
##  1st Qu.: 10.00   1st Qu.:0.3315
##  Median : 19.00   Median :0.7270
##  Mean   : 22.28   Mean   :0.6014
##  3rd Qu.: 27.00   3rd Qu.:0.8470
##  Max.   :201.00   Max.   :1.0490
```

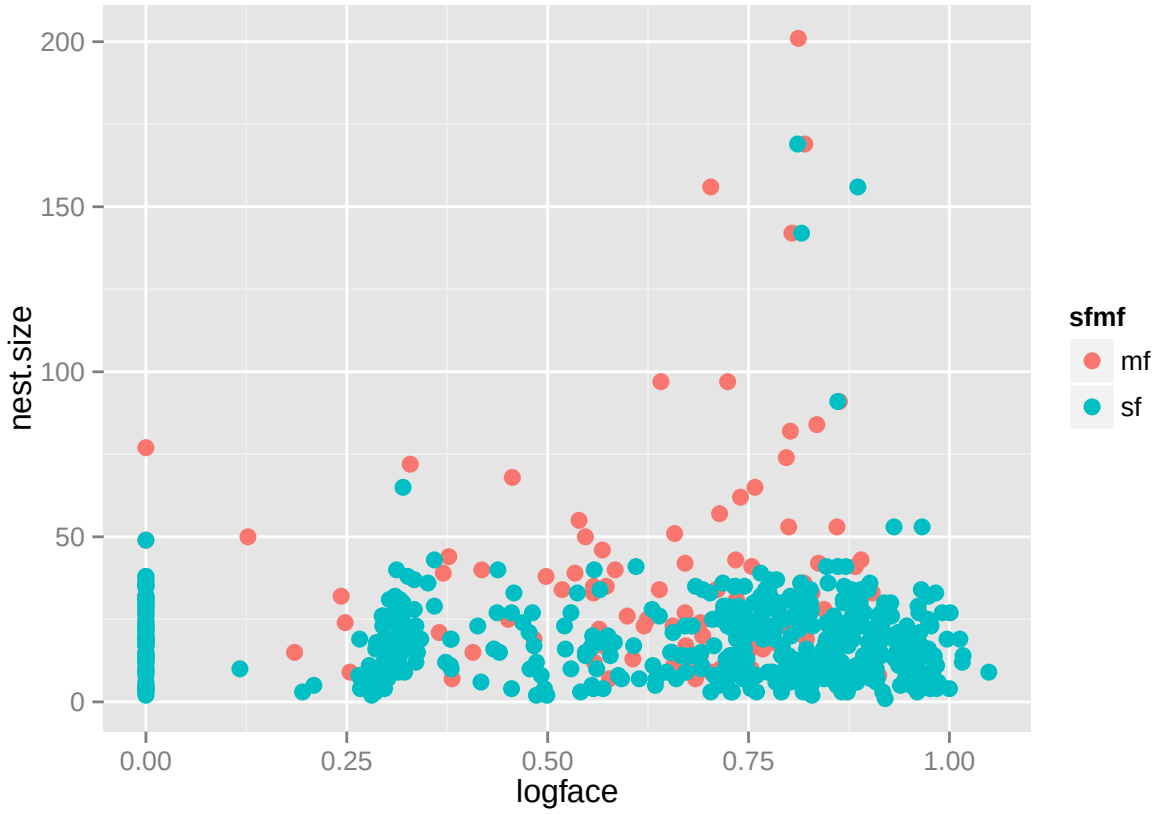
```r
# Create a scatter diagram
qplot(logface, nest.size, data=polistes.data) + geom_point(size=3)
```




```
qplot(logface, nest.size, data=polistes.data, shape=sfmf) + geom_point(size=3)
```




```
# Add colors based on a categorical variable
qplot(logface, nest.size, data=polistes.data, colour=sfmf) + geom_point(size=3)
```

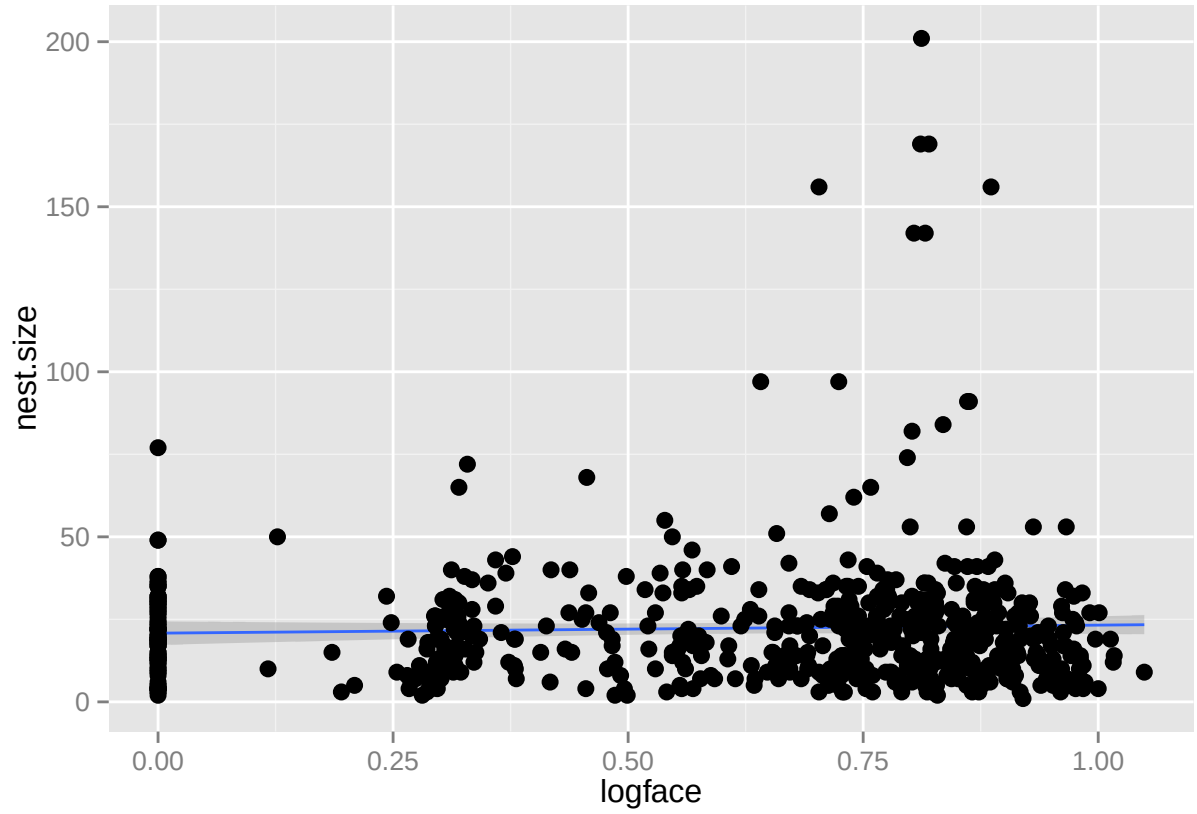


In `ggplot`, geoms, short for geometric objects, describe the type of plot you will produce. For example, `geom_point` is used for scatterplots. `geom_point` is the default in `qplot` if x and y are specified. If only x is specified, `qplot` defaults to `geom_histogram`. In the scatter diagrams above, the `size` argument in `geom_point` controls the size of data points.

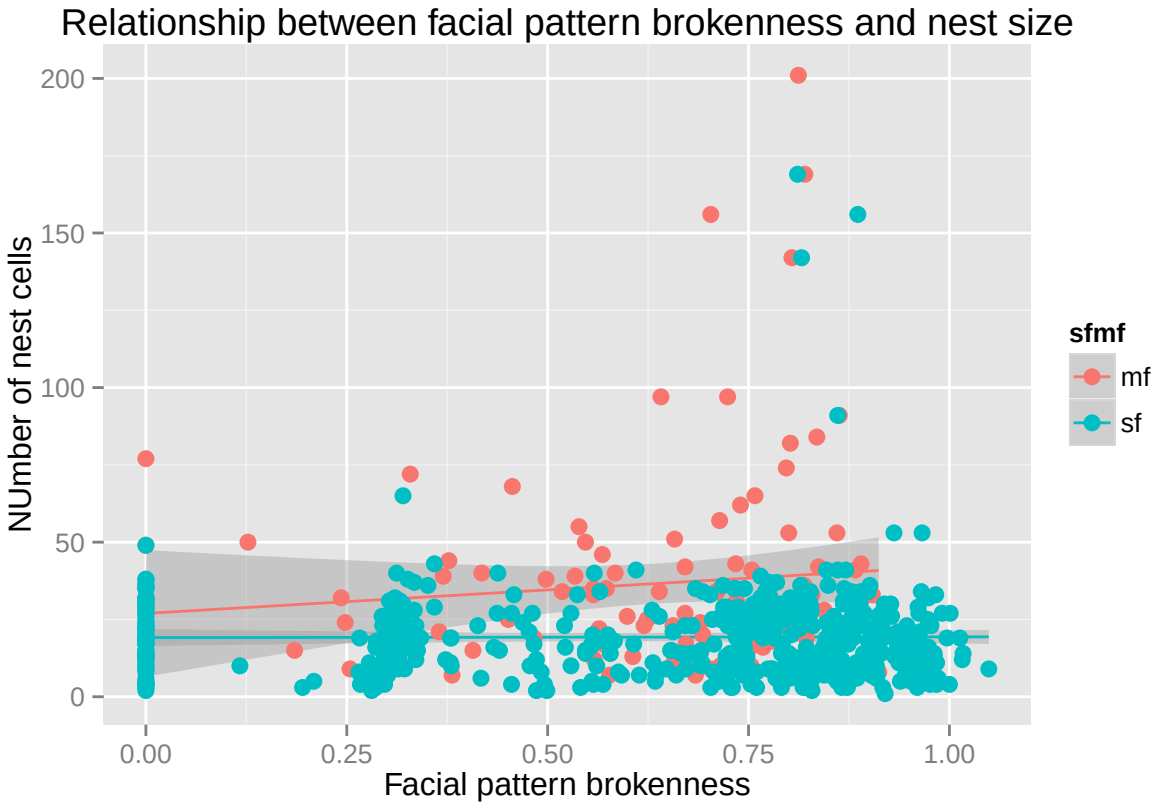
1.1.2 Fit a linear model

Add a smoothed line and fit linear models. `lm` is used to fit linear models and carry out regressions. A typical `lm` model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` specifies a linear predictor for response.

```
# Fit a linear model (by default includes 95% confidence region)
qplot(logface, nest.size, data=polistes.data, geom=c("point", "smooth"),
  method="lm") + geom_point(size=3)
```




```
# Create separate regressions for each factor and add labels
qplot(logface, nest.size, data=polistes.data, geom=c("point", "smooth"),
  method="lm", colour=sfmf,
  main="Relationship between facial pattern brokenness and nest size",
  xlab="Facial pattern brokenness",
  ylab="NUmber of nest cells") + geom_point(size=3)
```



1.1.3 Boxplots and jittered points

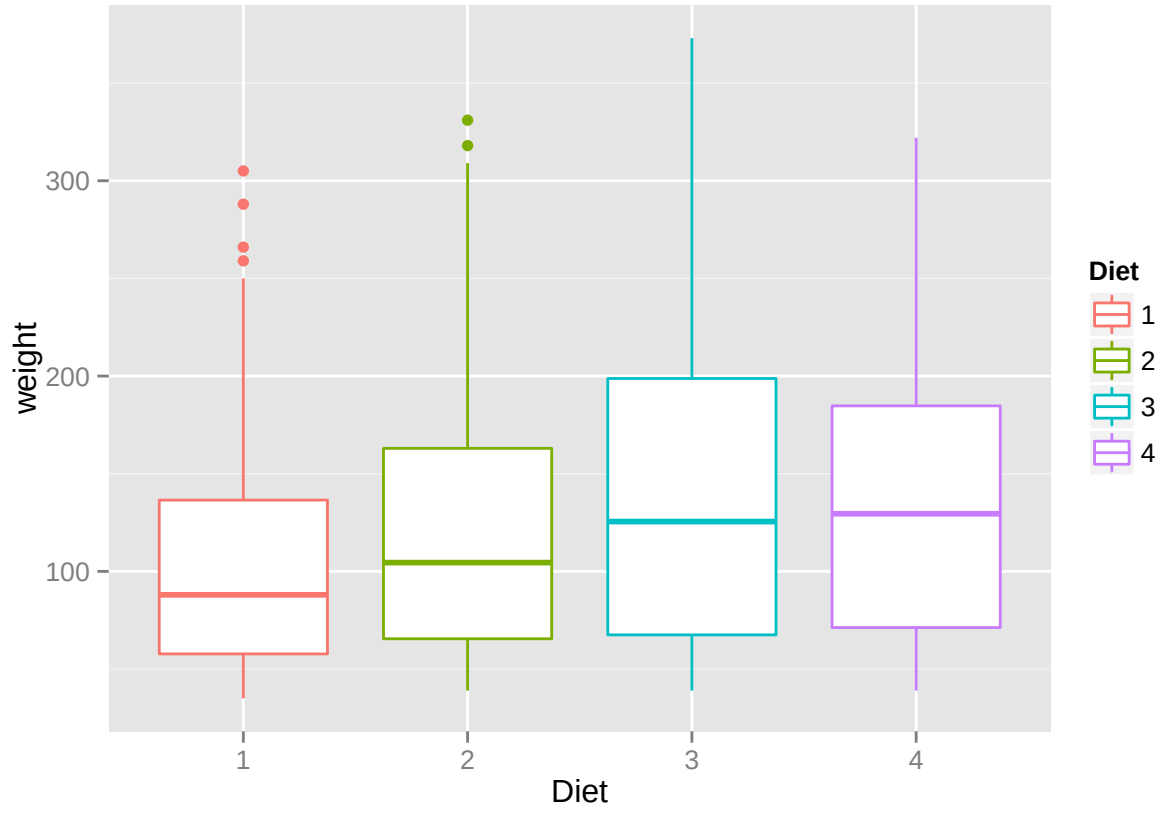
```
# Load the ChickWeight dataset from the base R packages
# Results from an experiment on the effect of diet on early growth of chicks
cw <- ChickWeight
summary(cw)
```

```
##      weight           Time           Chick     Diet
##  Min.   : 35.0   Min.   : 0.00   13     : 12   1:220
##  1st Qu.: 63.0   1st Qu.: 4.00   9      : 12   2:120
##  Median :103.0   Median :10.00   20     : 12   3:120
##  Mean   :121.8   Mean   :10.72   10     : 12   4:118
##  3rd Qu.:163.8   3rd Qu.:16.00   17     : 12
##  Max.   :373.0   Max.   :21.00   19     : 12
##                                  (Other):506
```

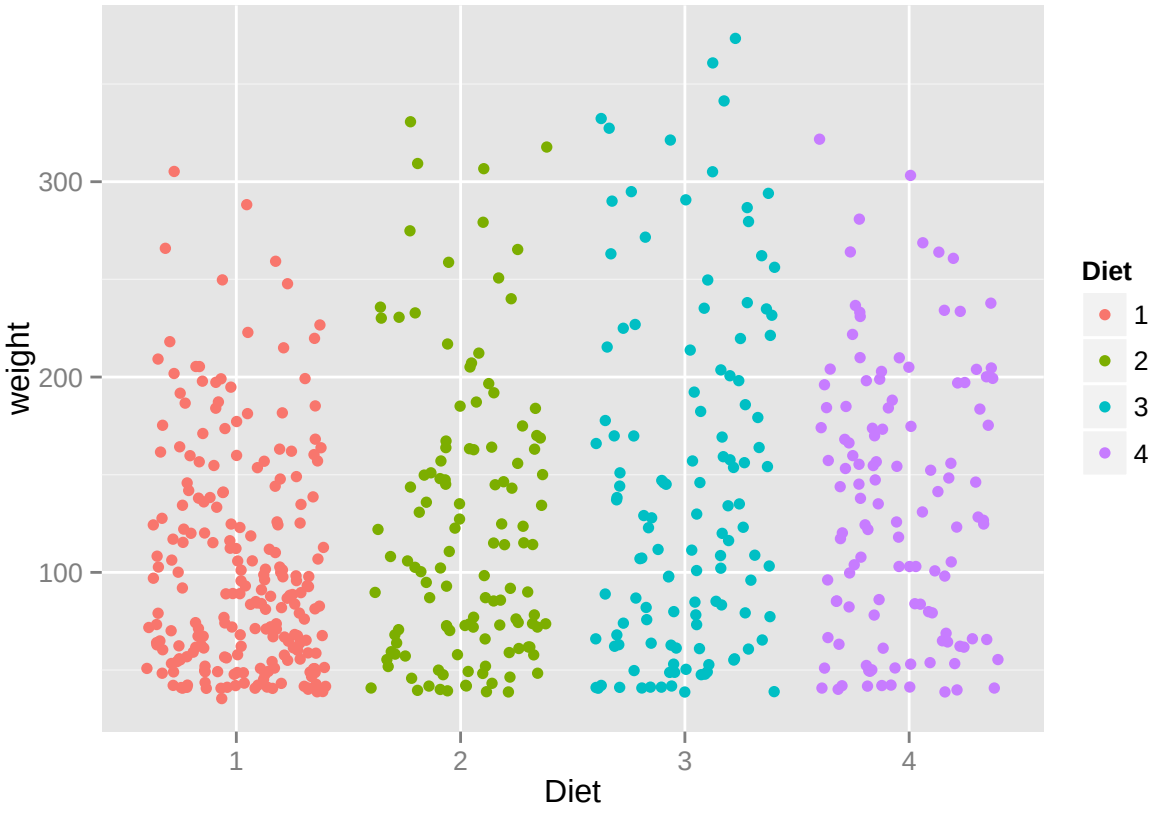
```
tapply(cw$weight, cw$Diet, FUN=mean)
```

```
##        1        2        3        4
## 102.6455 122.6167 142.9500 135.2627
```

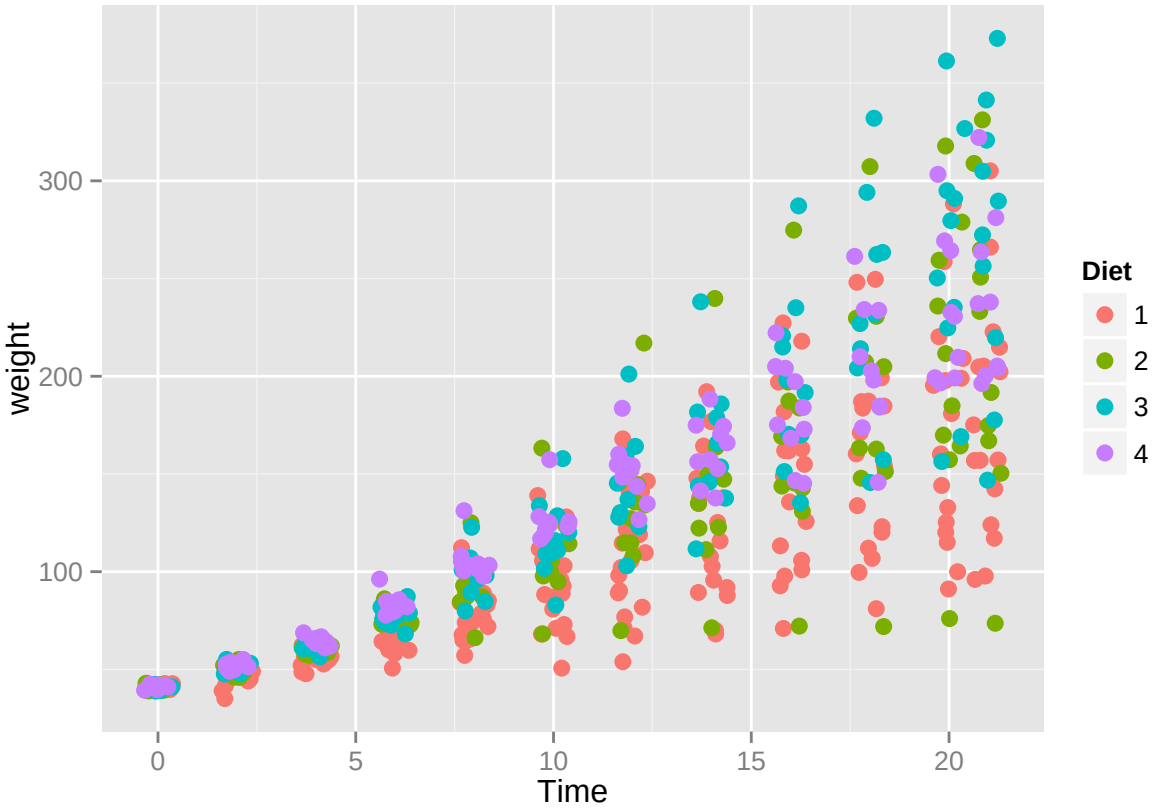
```
qplot(Diet, weight, data=cw, geom="boxplot", colour=Diet)
```



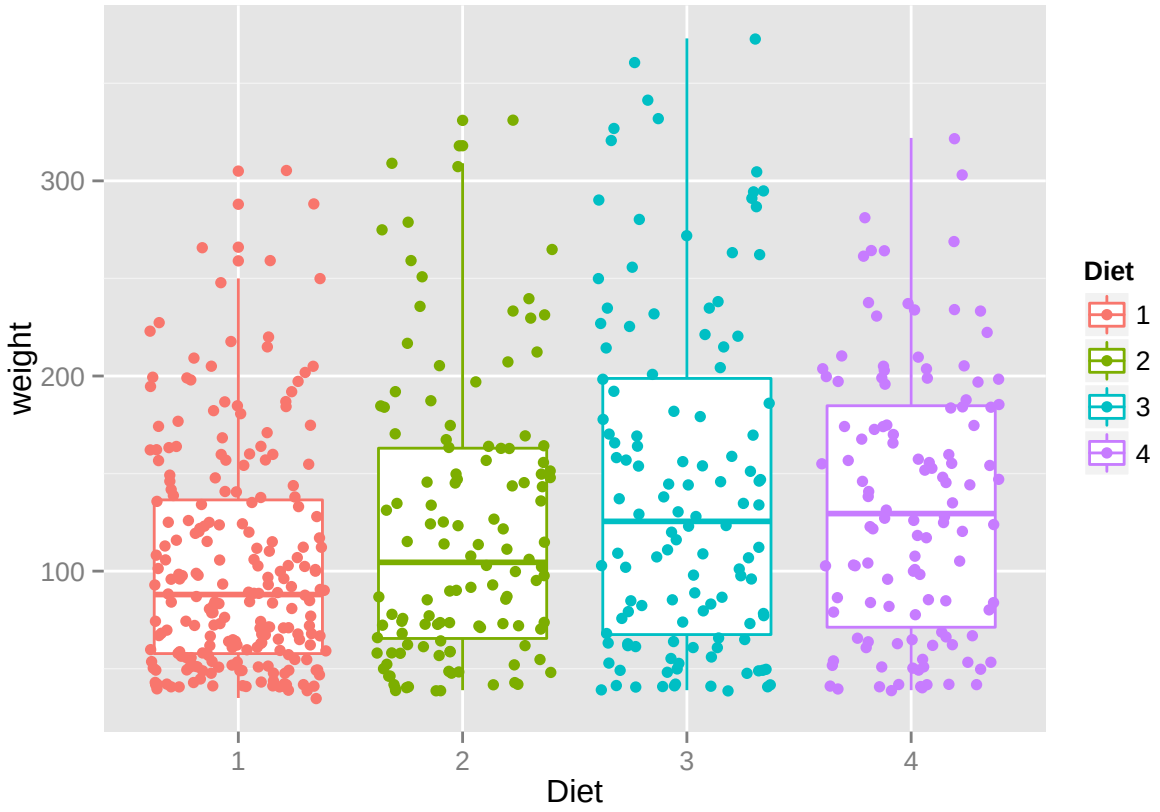

```
qplot(Diet, weight, data=cw, geom="jitter", colour=Diet)
```




```
# Use I() to manually set the aesthetics, e.g., colour = I("red") or size = I(3)
qplot(Time, weight, data=cw, geom="jitter", colour=Diet, size=I(3))
```



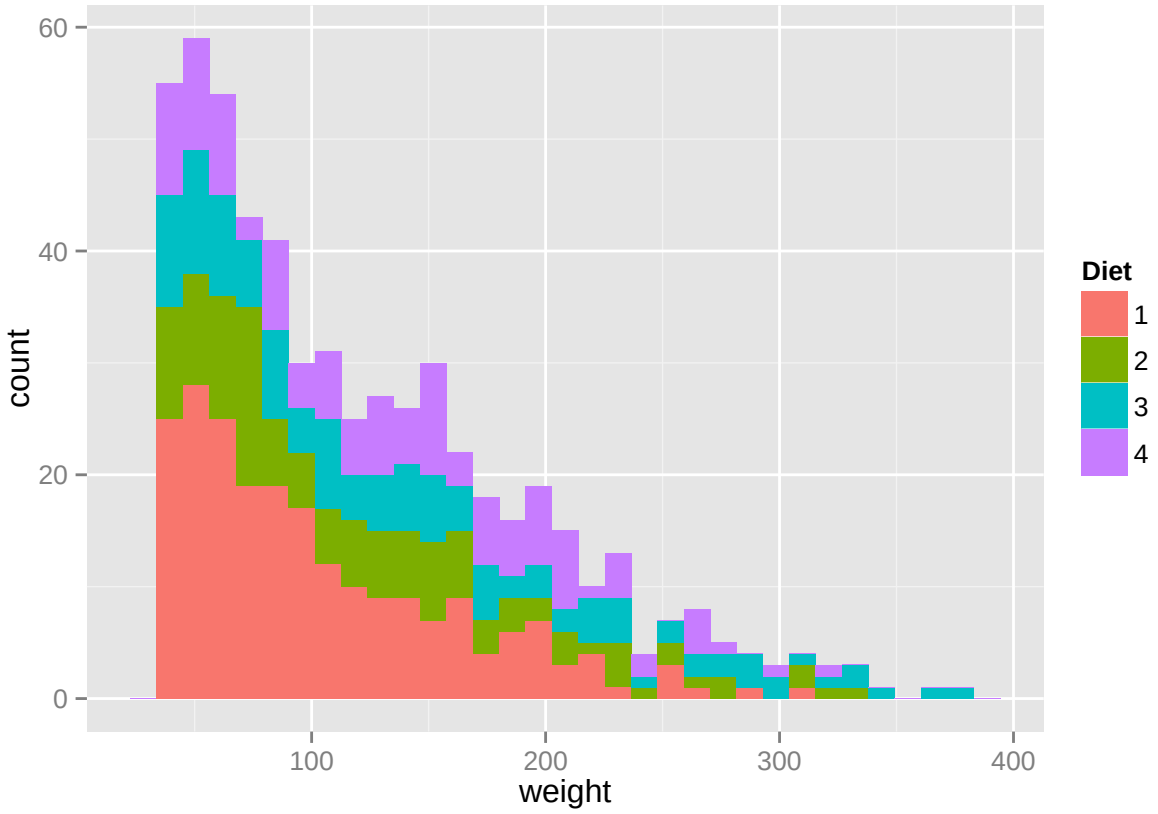

```
qplot(Diet, weight, data=cw, geom=c("boxplot","jitter"), colour=Diet)
```



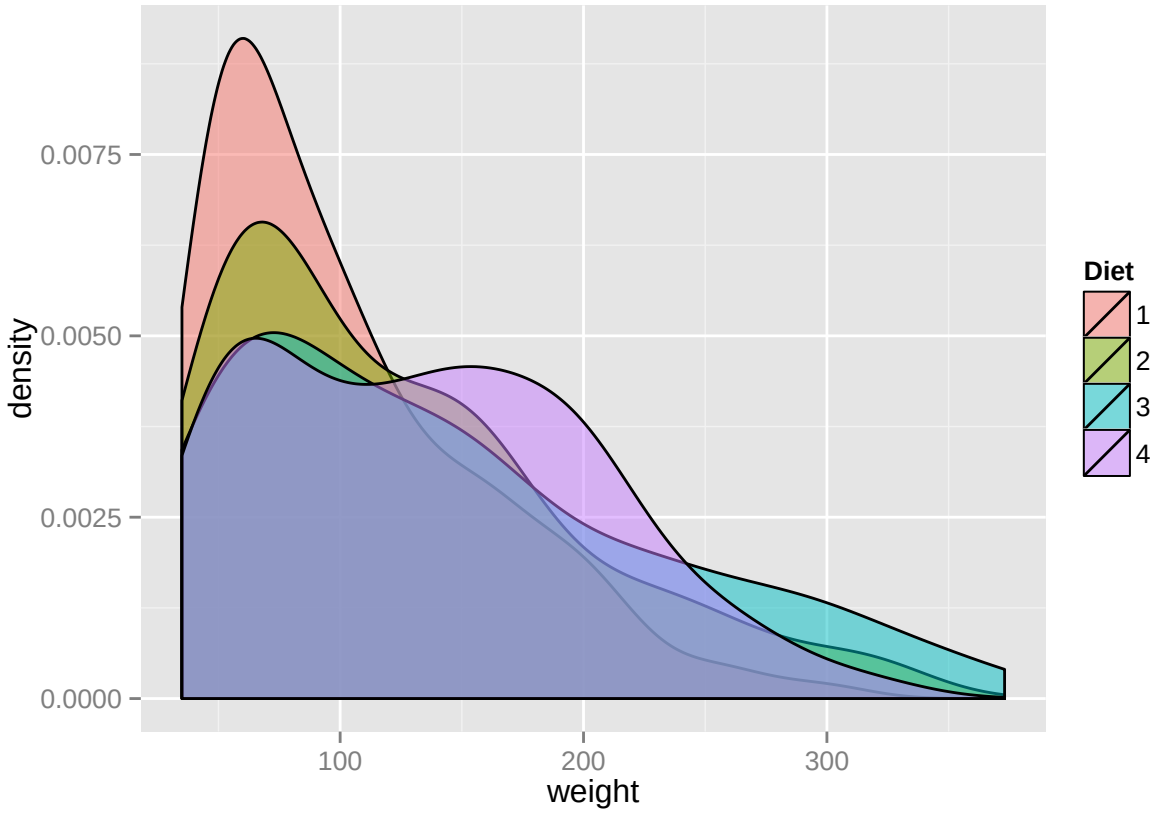
1.1.4 Histograms and density plots

```
qplot(weight, data=cw, geom="histogram", fill=Diet)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
qplot(weight, data=cw, geom="density", fill=Diet, alpha=I(0.5))
```

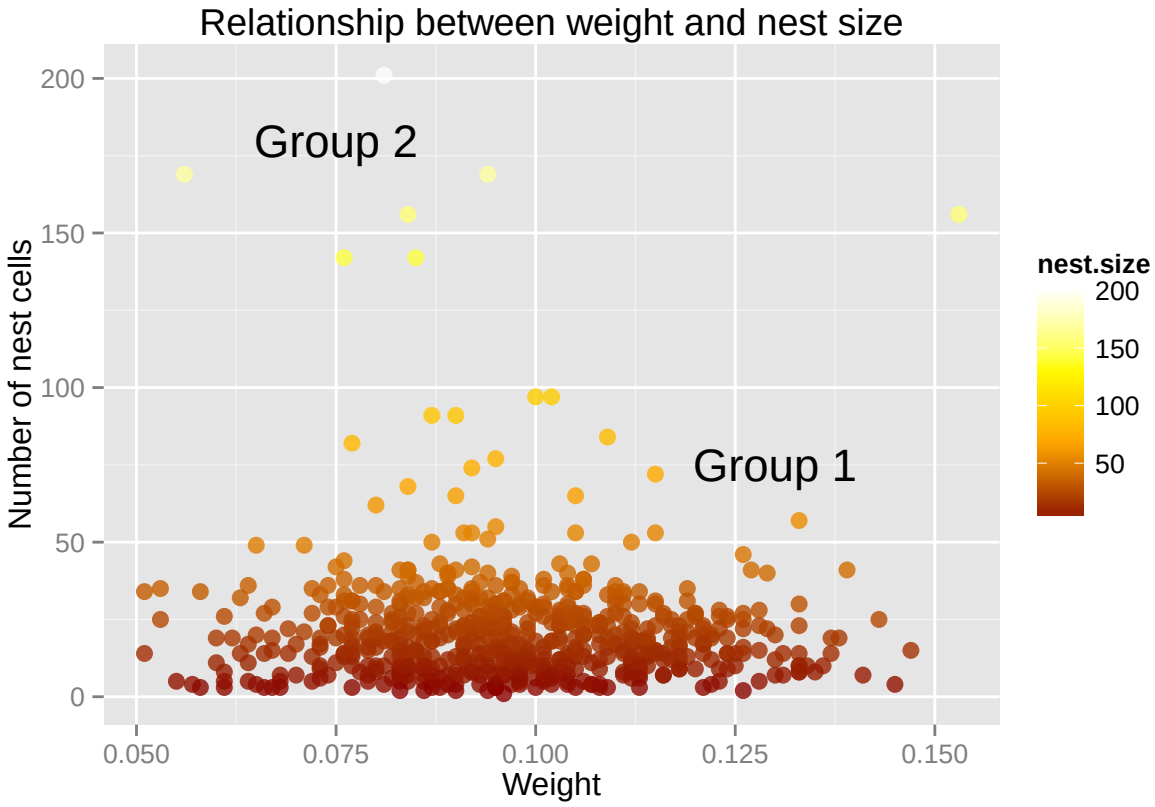


2 ggplot

`qplot` does not show the power of `ggplot`. `ggplot2` functions can be chained with "+" signs. All the options that can be chained are available at docs.ggplot2.org. Let us remake the previous graphs using a few of the wide variety of options available in `ggplot`.

2.1 Scatterplots

```
ggplot(polistes.data, aes(x=weight, y=nest.size, colour=nest.size)) +
  geom_point(size=3, alpha=0.8) +
  # Add main and axis titles
  labs(title="Relationship between weight and nest size",
    x="Weight", y="Number of nest cells") +
  # Add text to graph
  annotate("text", x=0.130, y=75, label="Group 1", size=6) +
  annotate("text", x=0.075, y=180, label="Group 2", size=6) +
  # Use a manually defined palette
  scale_colour_gradientn(colours=c("darkred", "orange", "yellow", "white"))
```

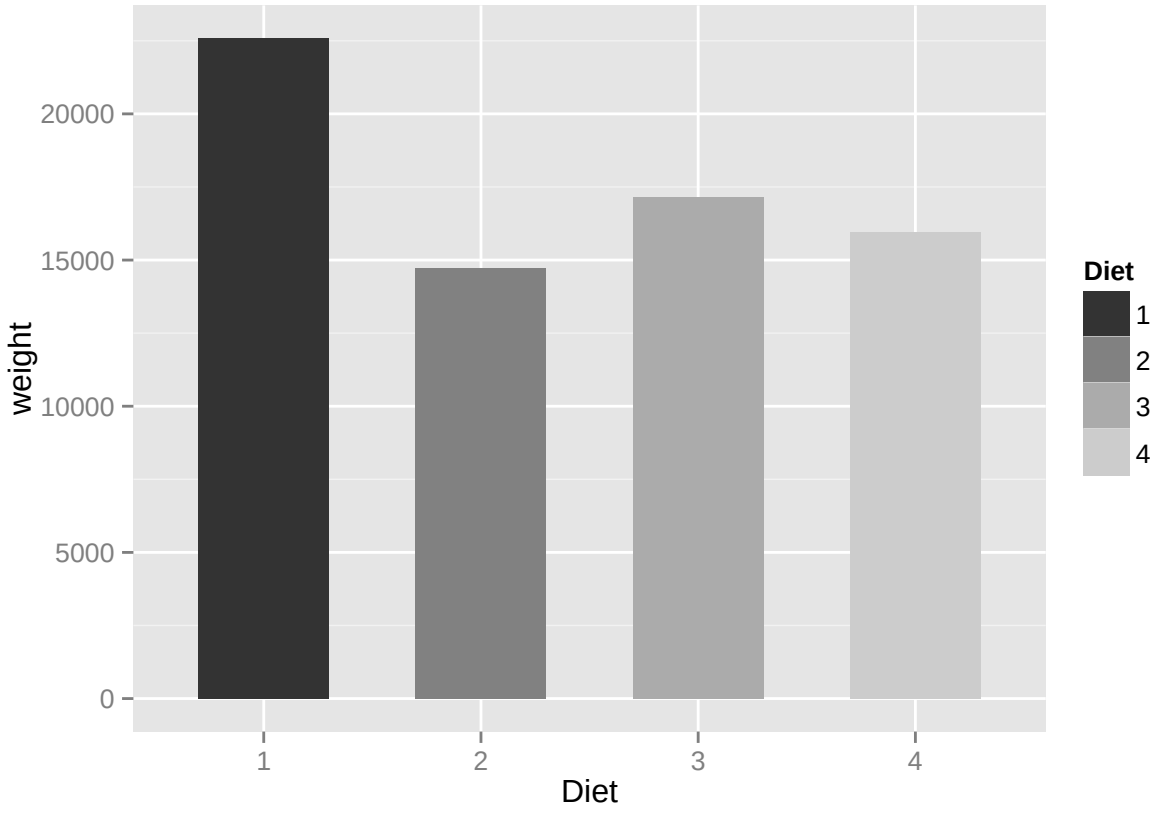


2.2 Boxplots

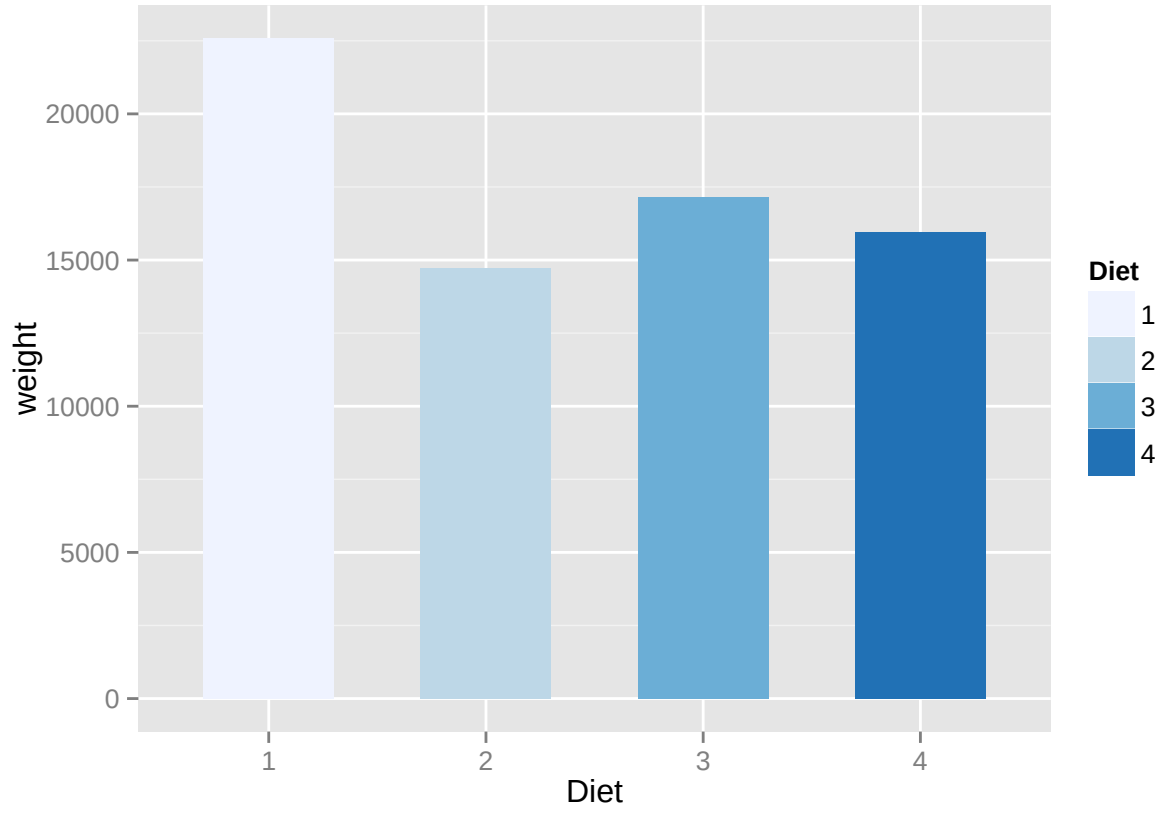
```
# The redundant legend can be removed with '+ guides(fill=FALSE)'
ggplot(cw, aes(x=Diet, y=weight, colour=Diet)) + geom_boxplot() +
  stat_summary(fun.y="mean", geom="point", shape=5, size=4) +
  labs(title="Relationship between diet and weight",
    x="Diet", y="Weight") +
  theme(plot.title=element_text(size=rel(1.5))) + # Title twice the base font size
  theme(axis.title.x=element_text(size=rel(1.2))) +
  theme(axis.title.y=element_text(size=rel(1.2))) +
  scale_fill_brewer(palette="Pastel1")
```

2.3 Bargraphs

```
# Use stat="identity" if you want the heights of the bars to represent
# values in the data
ggplot(cw, aes(x=Diet, y=weight, fill=Diet)) +
  geom_bar(stat="identity", width=0.6) +
  scale_fill_grey()
```



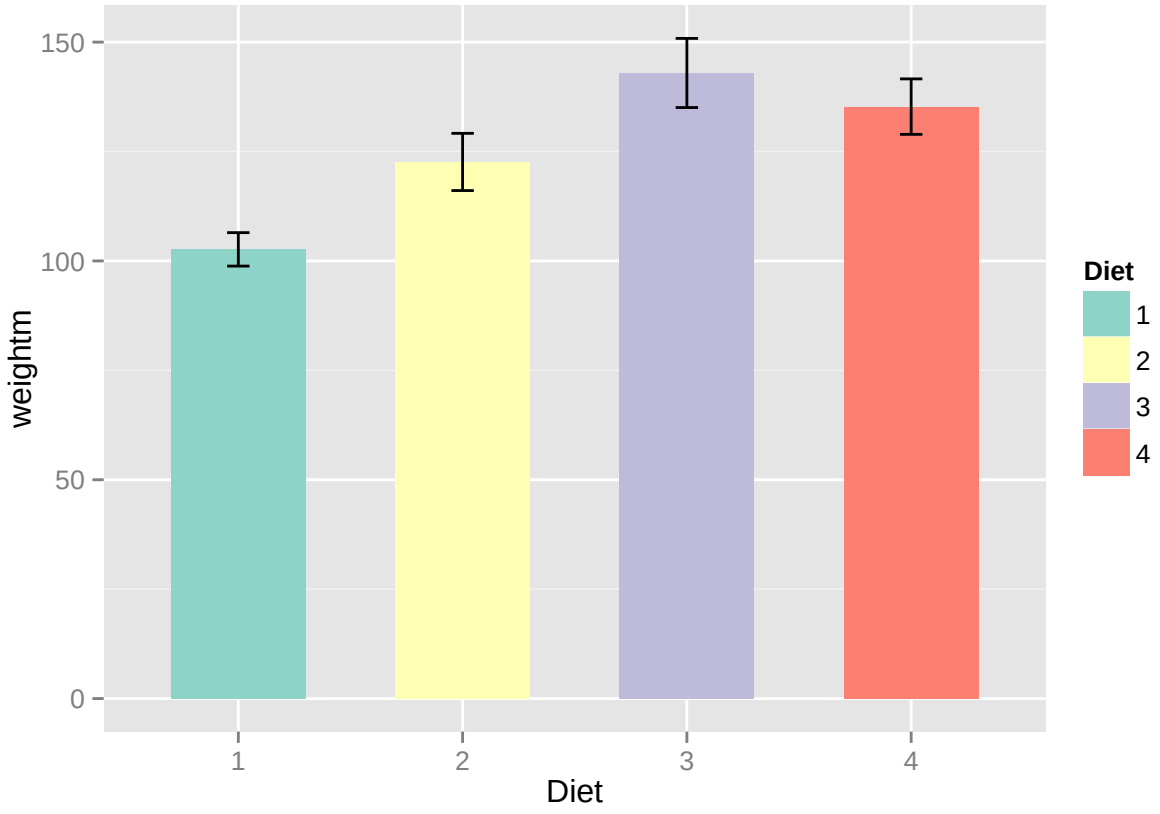

```
ggplot(cw, aes(x=Diet, y=weight, fill=Diet)) +
  geom_bar(stat="identity", width=0.6) +
  scale_fill_brewer(palette="Blues")
```



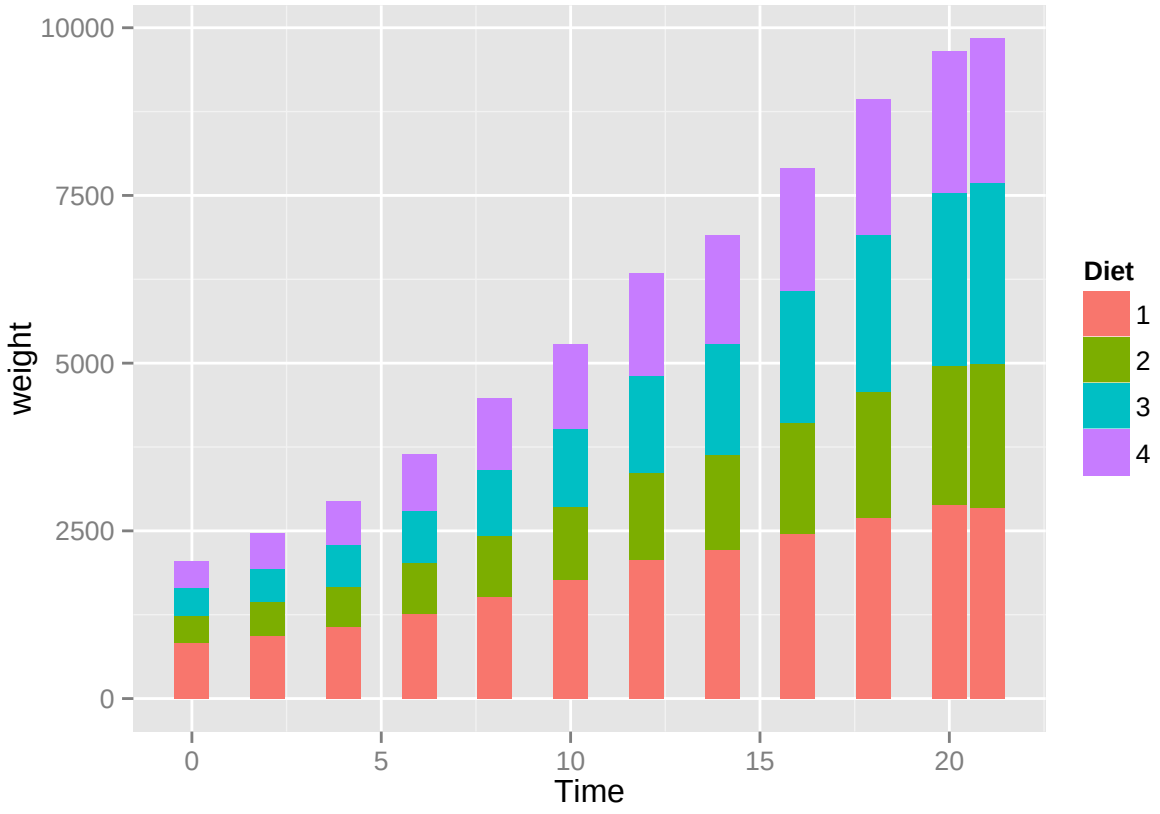

```
# Add error bars
# Split data frame, apply function, and return results in a data frame
library(plyr)
cwse <- ddply(cw, "Diet", summarise,
  weightm=mean(weight, na.rm=TRUE),
  sd=sd(weight, na.rm=TRUE),
  n=sum(!is.na(weight)),
  se=sd/sqrt(n))
cwse
```

```
##   Diet  weightm       sd   n       se
## 1    1 102.6455 56.65655 220 3.819784
## 2    2 122.6167 71.60749 120 6.536840
## 3    3 142.9500 86.54176 120 7.900146
## 4    4 135.2627 68.82871 118 6.336197
```

```
cwse$Diet <- as.factor(cwse$Diet)
ggplot(cwse, aes(x=Diet, y=weightm, fill=Diet)) +
  geom_bar(stat="identity", width=0.6) +
  geom_errorbar(aes(ymin=weightm-se, ymax=weightm+se), width=0.1) +
  scale_fill_brewer(palette="Set3")
```

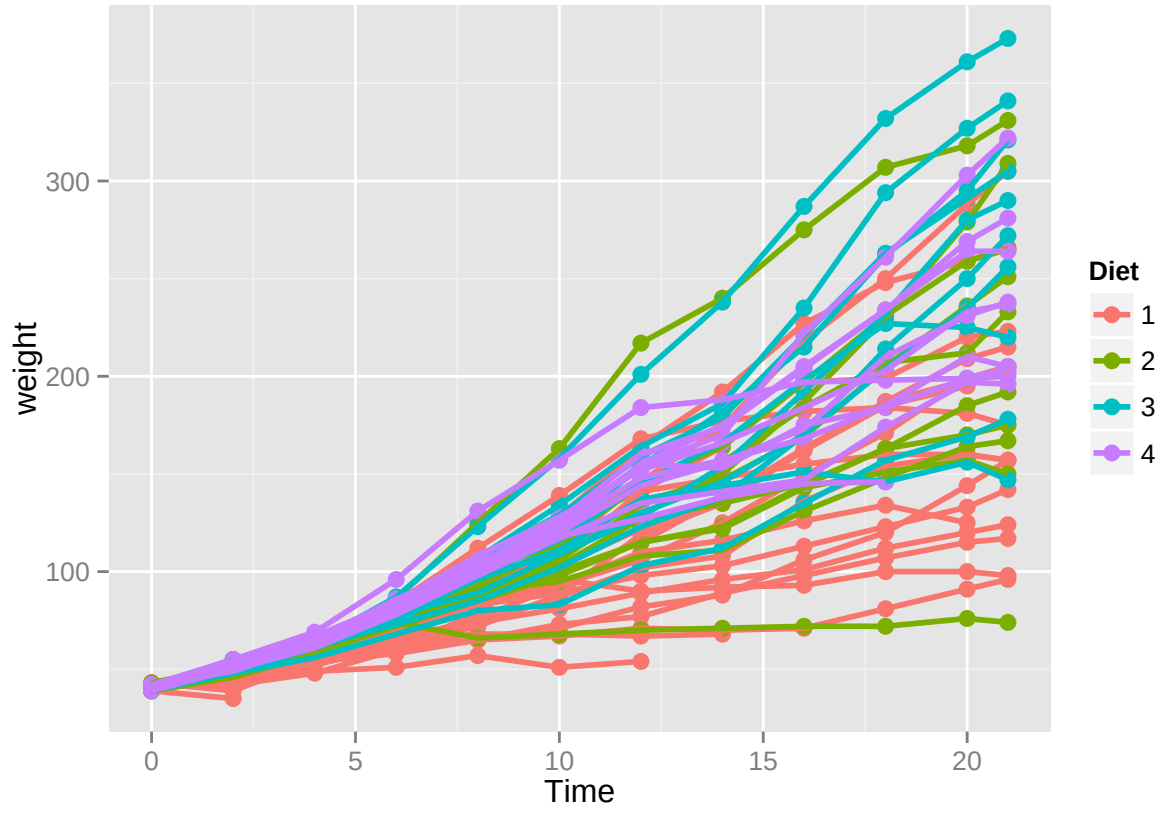



```
# Stacked bar graph
ggplot(cw, aes(x=Time, y=weight, fill=Diet)) +
  geom_bar(stat="identity")
```



2.4 Line graphs and stacked area graphs

```
# If your line graph looks wrong, specify the grouping variable with `group`.
# Problems occur with line graphs because ggplot() is unable to determine
# how to group the variables
# A sawtooth pattern results from improper grouping
ggplot(cw, aes(x=Time, y=weight, group=Chick, colour=Diet)) +
  geom_point(size=3) + geom_line(size=1)
```




```
cwd12 <- subset(cw, Diet==1:2)
summary(cwd12)
```

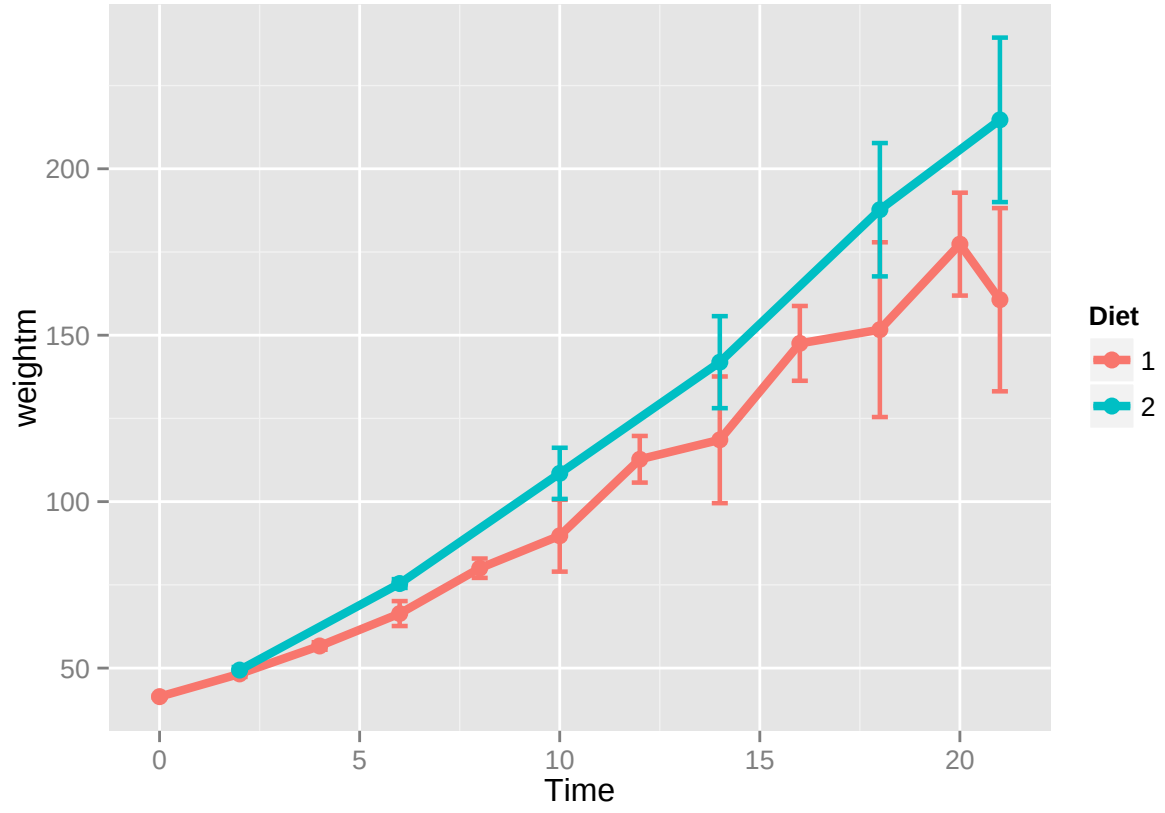
```
##      weight           Time           Chick     Diet
##  Min.   : 39.0   Min.   : 0.00   13     :  6   1:110
##  1st Qu.: 61.0   1st Qu.: 6.00   9      :  6   2: 60
##  Median : 95.5   Median :10.00   20     :  6
##  Mean   :111.3   Mean   :10.86   10     :  6
##  3rd Qu.:148.8   3rd Qu.:17.50   8      :  6
##  Max.   :331.0   Max.   :21.00   17     :  6
##                                  (Other):134
```

```
cwse12 <- ddply(cwd12, c("Diet", "Time"), summarise,
  weightm=mean(weight, na.rm=TRUE),
  sd=sd(weight, na.rm=TRUE),
  n=sum(!is.na(weight)),
  se=sd/sqrt(n))
cwse12
```

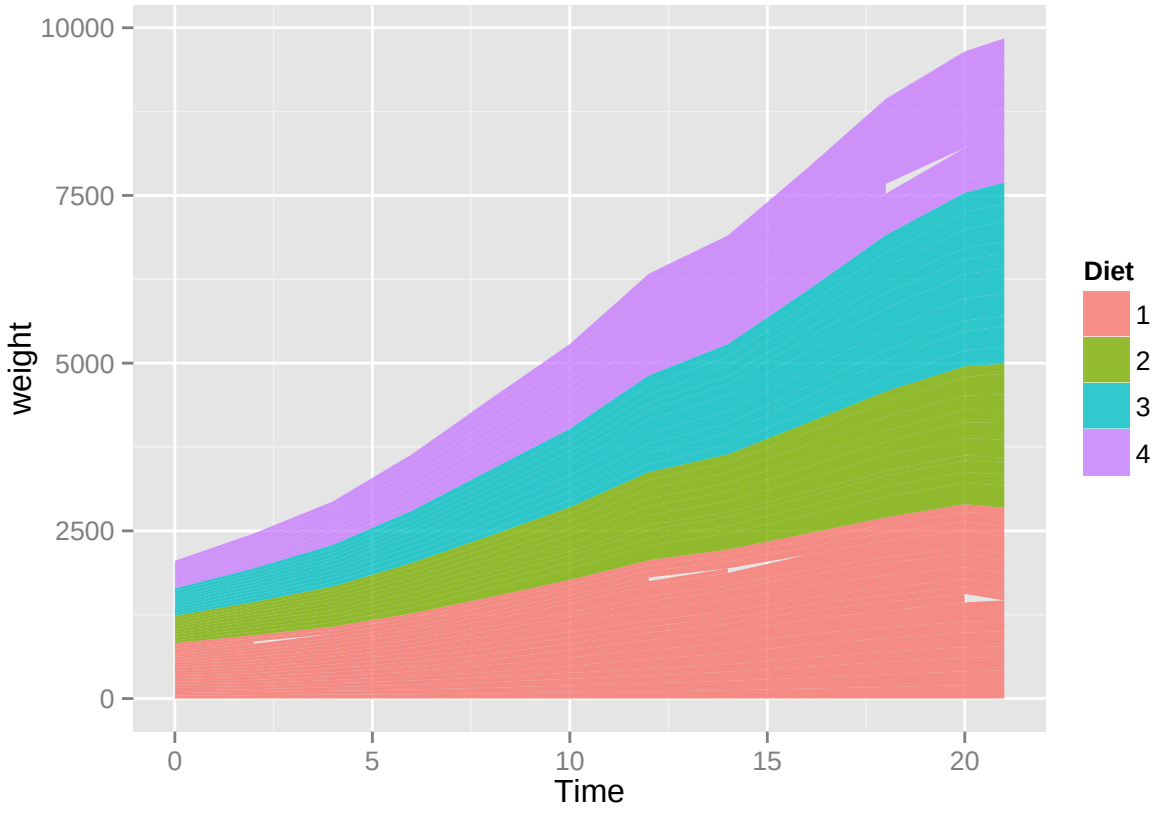
```
##    Diet Time   weightm        sd  n         se
## 1     1    0  41.41667  1.164500 12  0.3361622
## 2     1    2  48.25000  2.549510  8  0.9013878
## 3     1    4  56.63636  3.722169 11  1.1222763
## 4     1    6  66.37500 10.595653  8  3.7461290
## 5     1    8  80.00000  9.643651 11  2.9076701
```

```
## 6     1   10  89.75000 30.471298  8 10.7732307
## 7     1   12 112.72727 23.242594 11  7.0079058
## 8     1   14 118.57143 50.345000  7 19.0286215
## 9     1   16 147.54545 37.264899 11 11.2357899
## 10    1   18 151.66667 64.301374  6 26.2509259
## 11    1   20 177.36364 51.252849 11 15.4533155
## 12    1   21 160.66667 67.408209  6 27.5192862
## 13    2    2  49.40000  2.875181 10  0.9092121
## 14    2    6  75.40000  4.168666 10  1.3182480
## 15    2   10 108.50000 24.295633 10  7.6829537
## 16    2   14 141.90000 43.697063 10 13.8182247
## 17    2   18 187.70000 63.331667 10 20.0272315
## 18    2   21 214.70000 78.138126 10 24.7094449
```

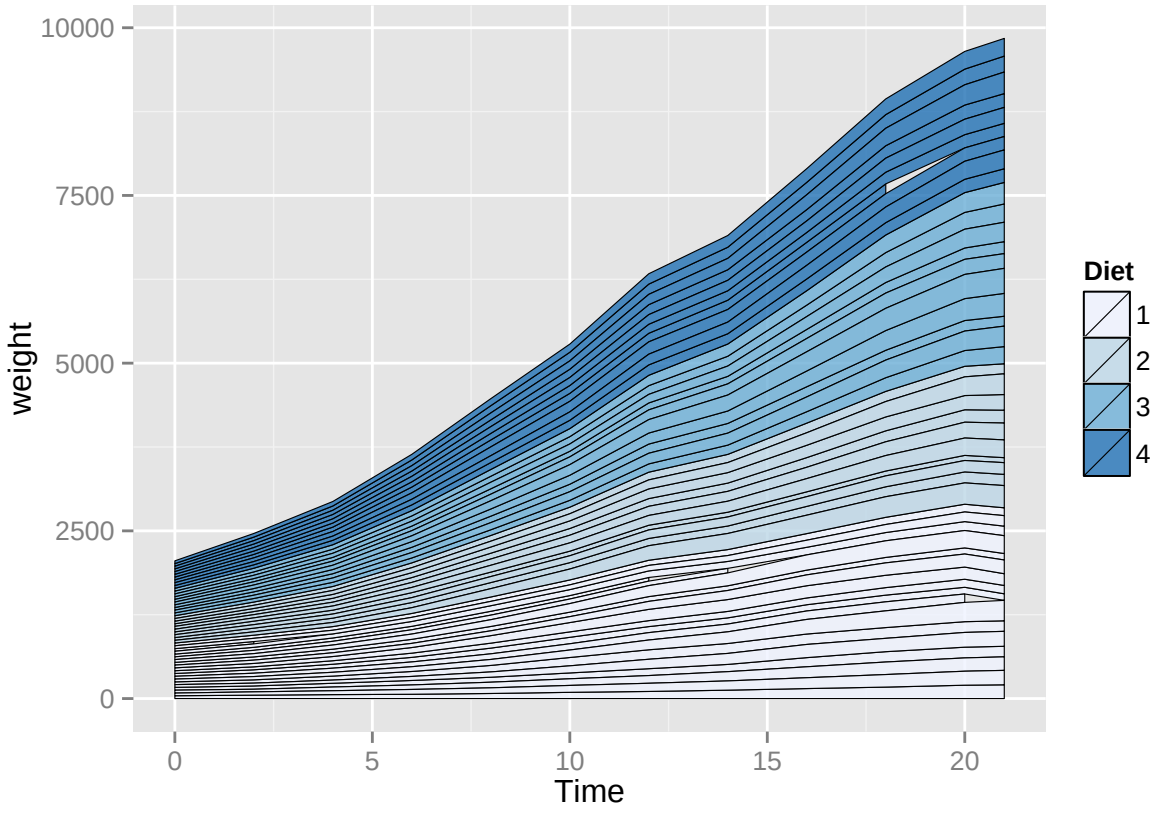
```
ggplot(cwse12, aes(x=Time, y=weightm, colour=Diet)) +
    geom_errorbar(aes(ymin=weightm-se, ymax=weightm+se), width=.4, size=0.8) +
    geom_line(size=1.5) +
    geom_point(size=3)
```




```
# Stacked area graph
# Area graphs represent cumulated totals over time
#tapply(cw$weight, cw$Diet, FUN=sum)
#cwd4 <- subset(cw, Diet==4)
#tapply(cwd4$weight, cwd4$Time, FUN=sum)
ggplot(cw, aes(x=Time, y=weight, group=Chick, fill=Diet)) +
  geom_area(alpha=0.8)
```



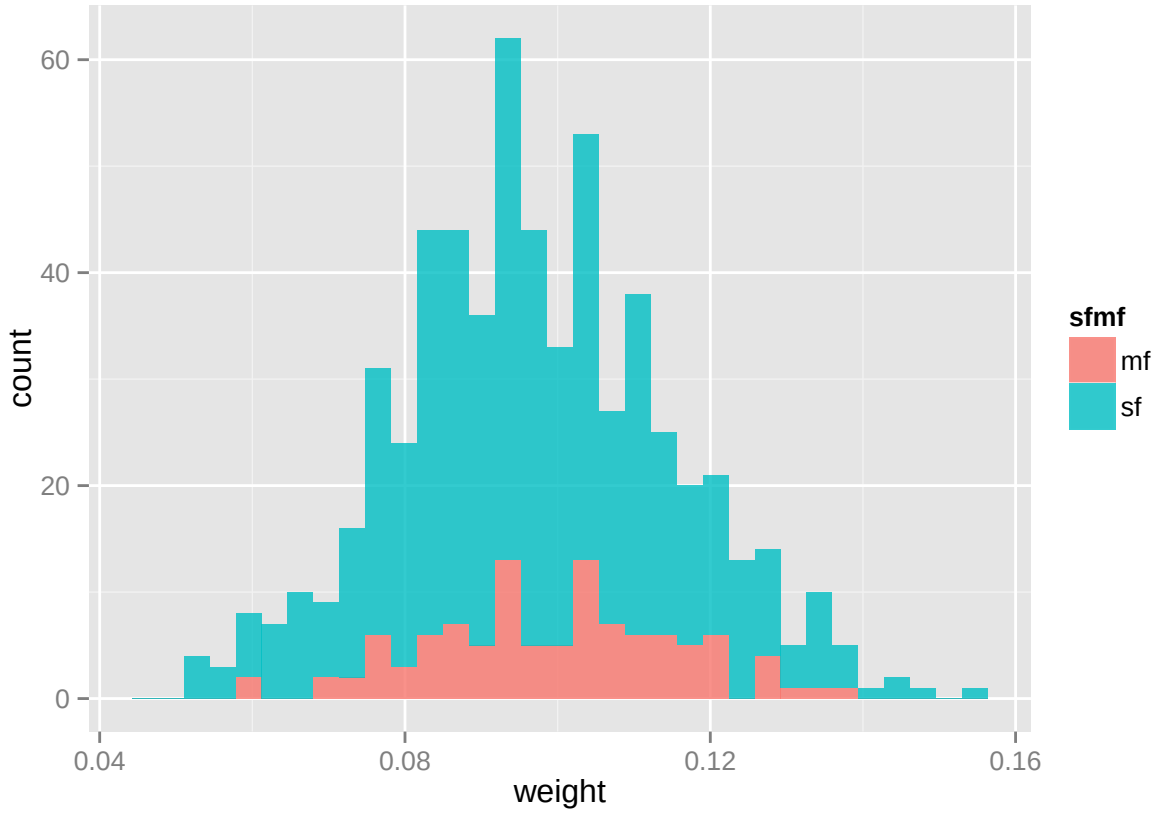

```
ggplot(cw, aes(x=Time, y=weight, group=Chick, fill=Diet)) +
  geom_area(colour="black", size=0.2, alpha=0.8) +
  scale_fill_brewer(palette="Blues")
```



2.5 Histograms

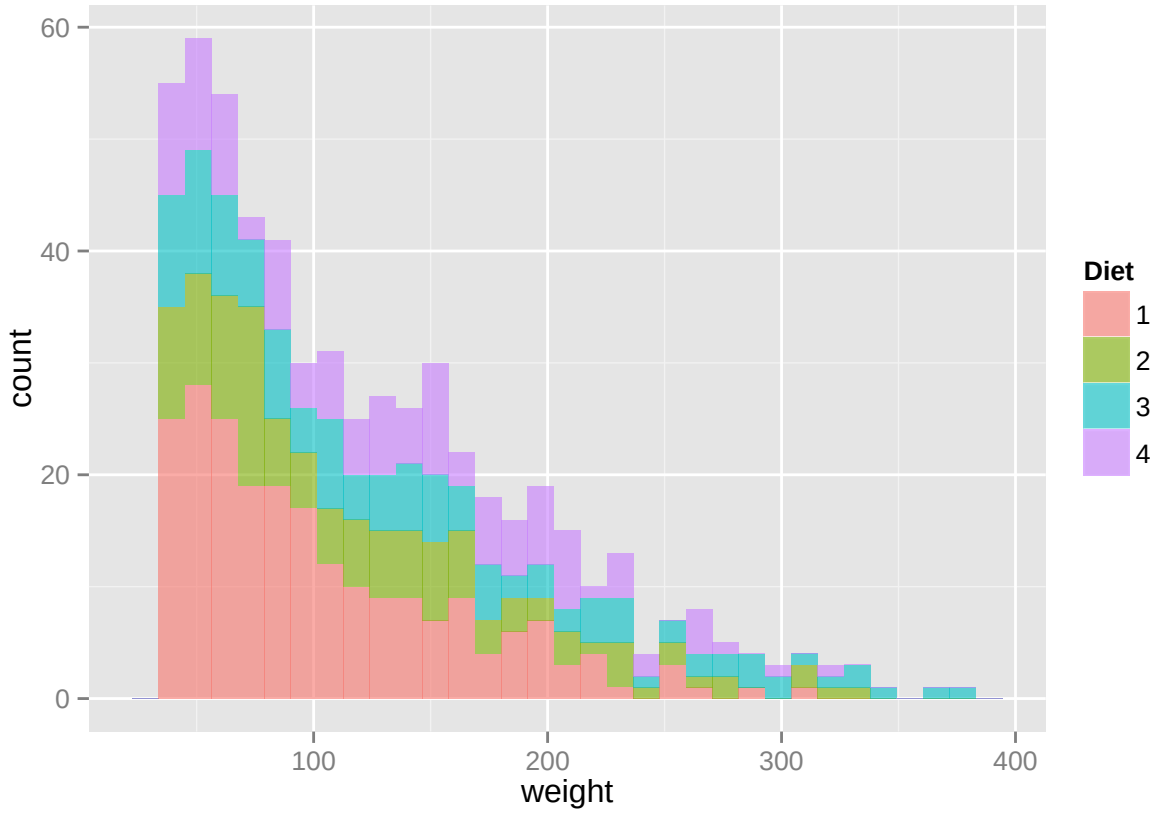
```
ggplot(polistes.data, aes(x=weight, fill=sfmf)) + geom_histogram(alpha=0.8)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
ggplot(cw, aes(x=weight, fill=Diet)) + geom_histogram(alpha=0.6)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

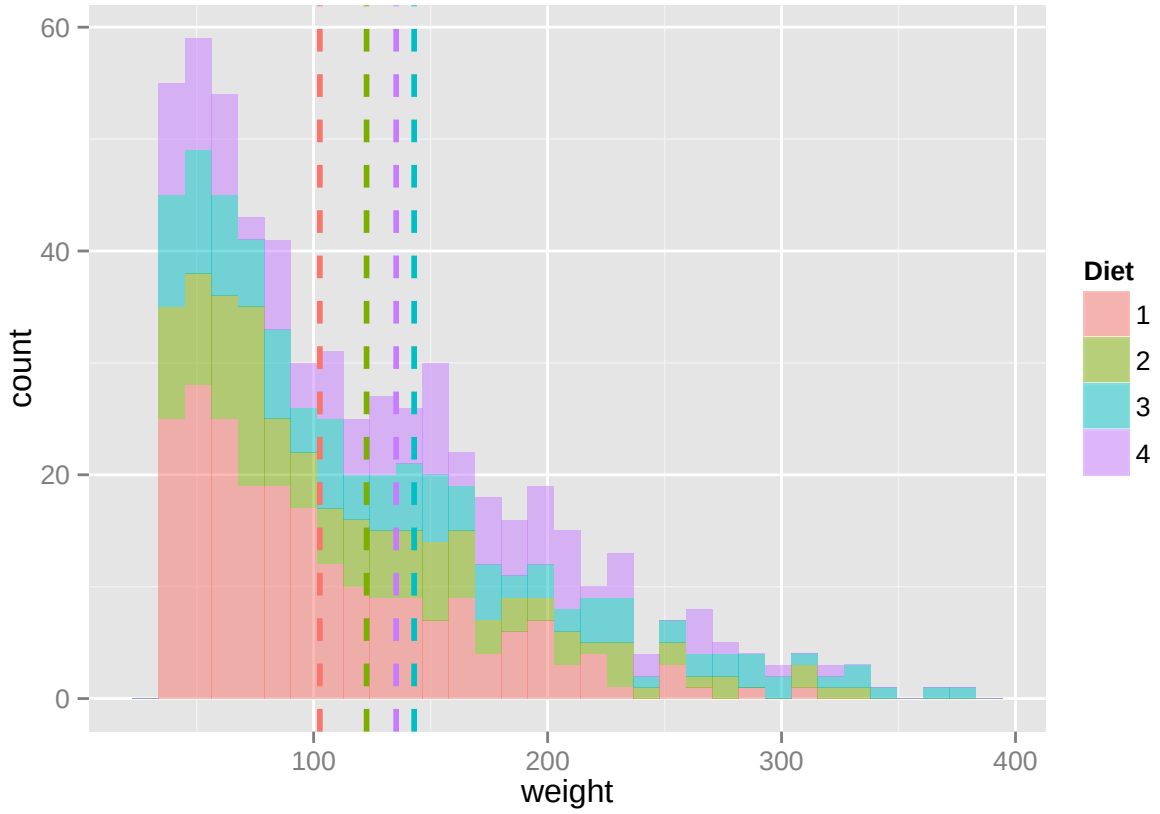



```
# Find the mean of each group
library(plyr)
# For each subset of a data frame, ddply applies a function and then combines
# results into a data frame
mwt <- ddply(cw, "Diet", summarise, weight.mean=mean(weight))
mwt
```

```
##   Diet weight.mean
## 1    1    102.6455
## 2    2    122.6167
## 3    3    142.9500
## 4    4    135.2627
```

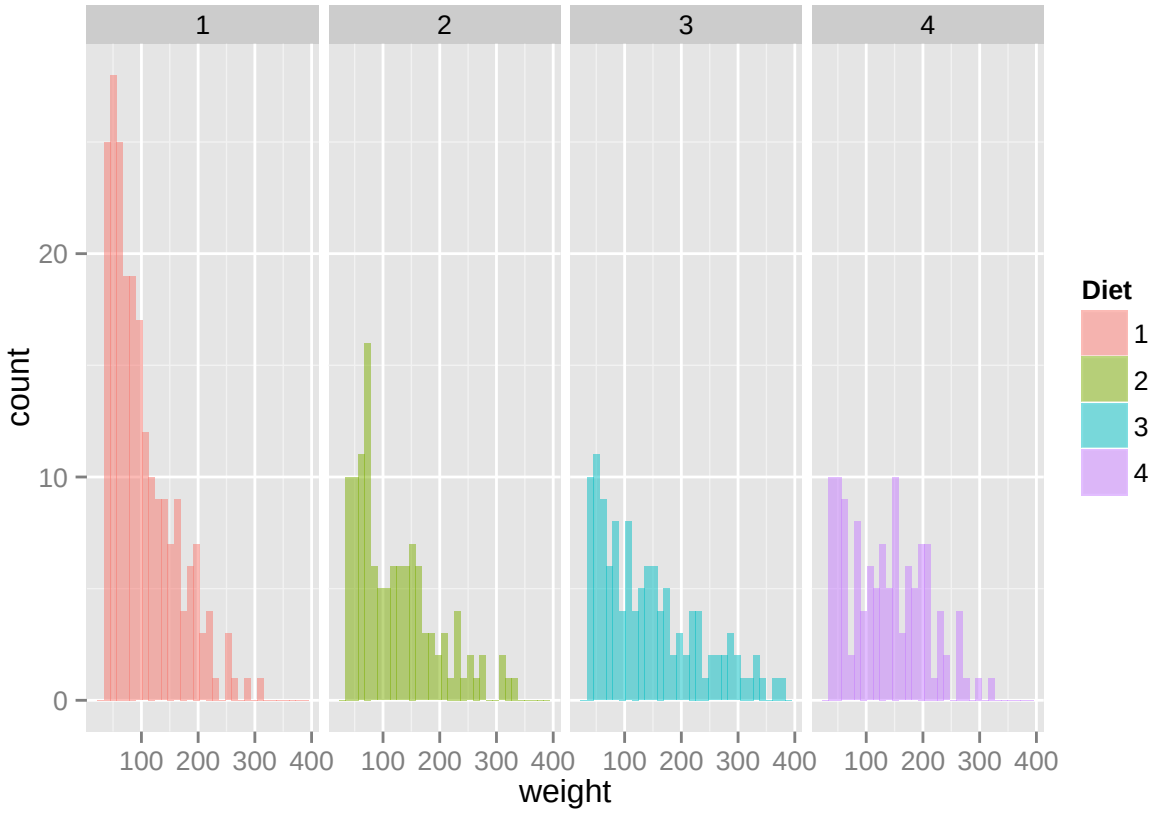
```
# Overlaid histograms with means
ggplot(cw, aes(x=weight, fill=Diet)) + geom_histogram(alpha=0.5) +
    geom_vline(data=mwt, aes(xintercept=weight.mean, colour=Diet),
      linetype="dashed", size=1)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```




```
# Use facets to display subsets of the dataset in different panels
ggplot(cw, aes(x=weight, fill=Diet)) + geom_histogram(alpha=0.5) +
  facet_grid(. ~ Diet)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



2.6 Scatterplots with marginal density plots

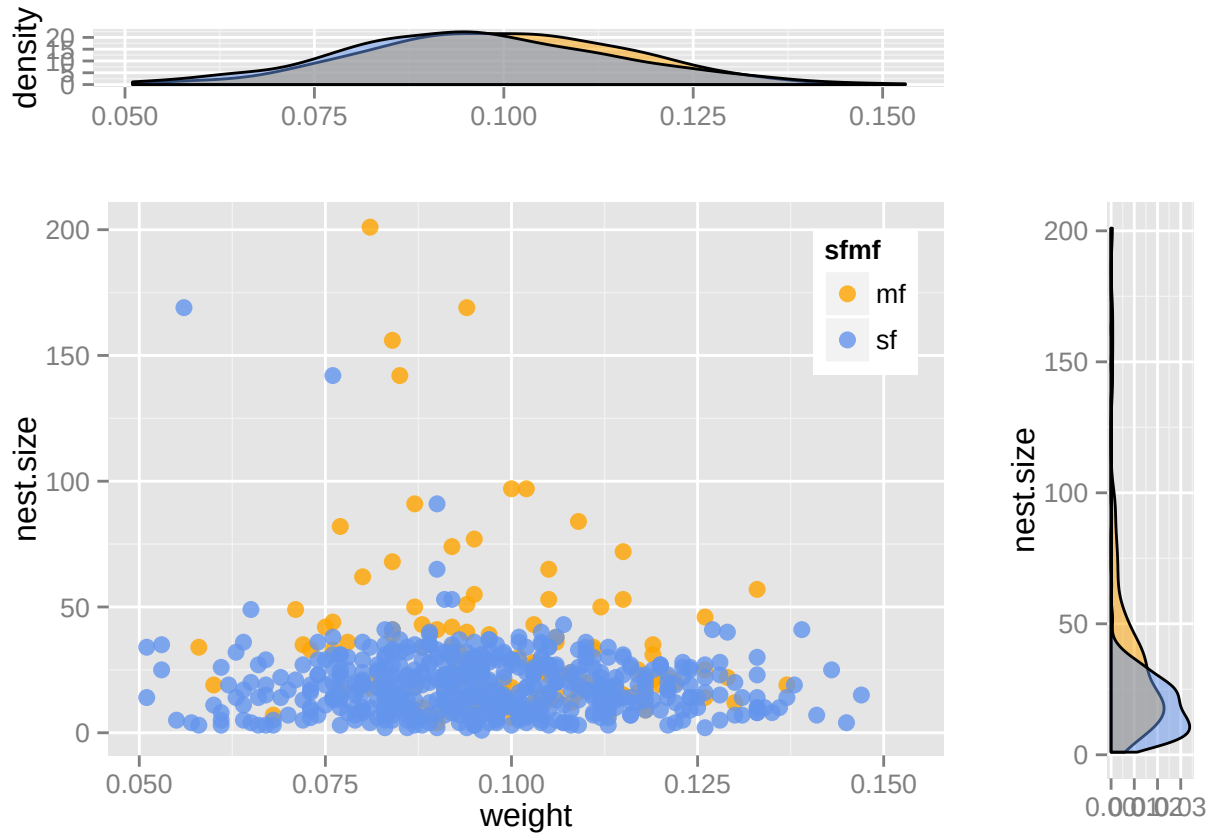
```
empty.plot <- ggplot() + geom_point(aes(1,1), colour="white") +
  theme(plot.background=element_blank(),
    panel.grid.major=element_blank(),
    panel.grid.minor=element_blank(),
    panel.border=element_blank(),
    panel.background=element_blank(),
    axis.title.x=element_blank(),
    axis.title.y=element_blank(),
    axis.text.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks=element_blank())

xyscatter <- ggplot(polistes.data, aes(x=weight, y=nest.size,
  colour=sfmf)) + geom_point(size=3, alpha=.8) +
  scale_color_manual(values=c("orange", "cornflowerblue")) +
  theme(legend.position=c(1,1), legend.justification=c(1,1))

xdensity.top <- ggplot(polistes.data, aes(weight, fill=sfmf)) +
  geom_density(alpha=0.5) +
  scale_fill_manual(values=c("orange", "cornflowerblue")) +
  theme(legend.position="none", axis.title.x=element_blank())
```

```
ydensity.right <- ggplot(polistes.data, aes(nest.size, fill=sfmf)) +
  geom_density(alpha=0.5) +
  coord_flip() +
  scale_fill_manual(values=c("orange", "cornflowerblue")) +
  theme(legend.position="none", axis.title.x=element_blank())

# Arrange the plots together
grid.arrange(xdensity.top, empty.plot, xyscatter, ydensity.right,
  ncol=2, nrow=2, widths=c(4, 1), heights=c(1, 4))
```

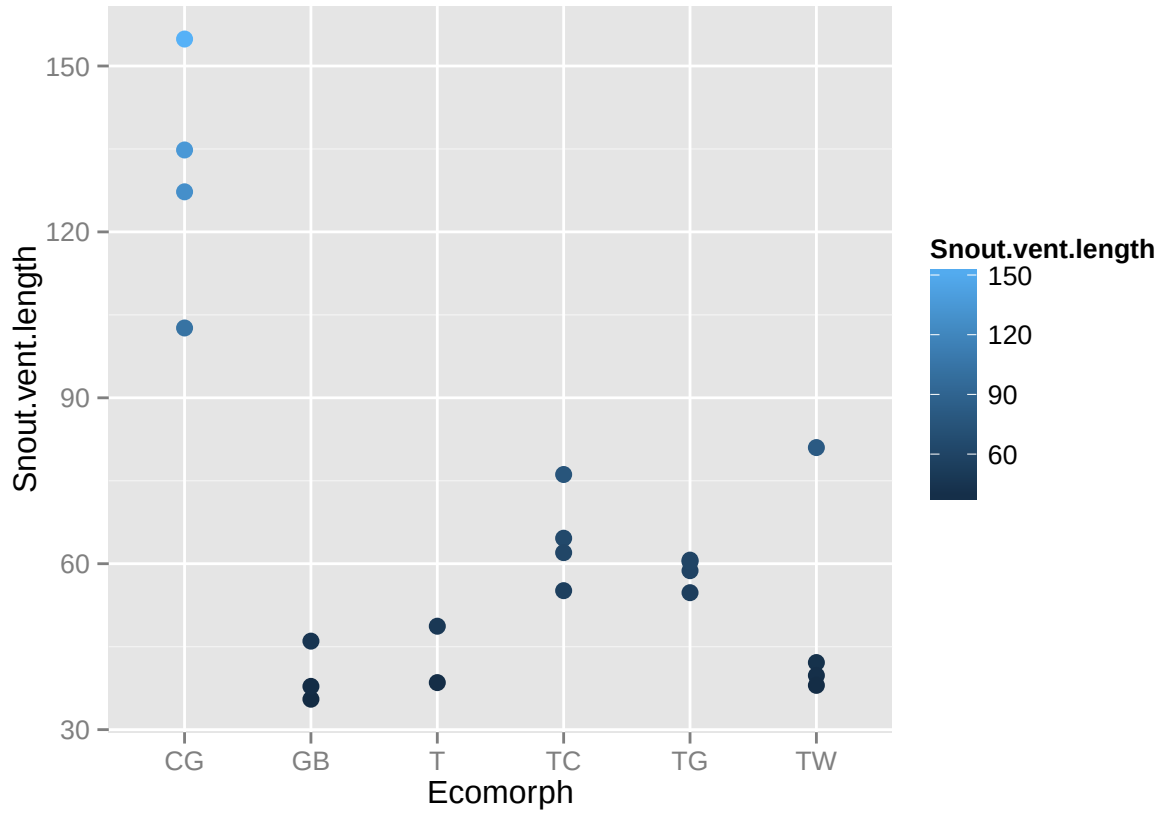


3 More ggplot and an introduction to rOpenSci packages

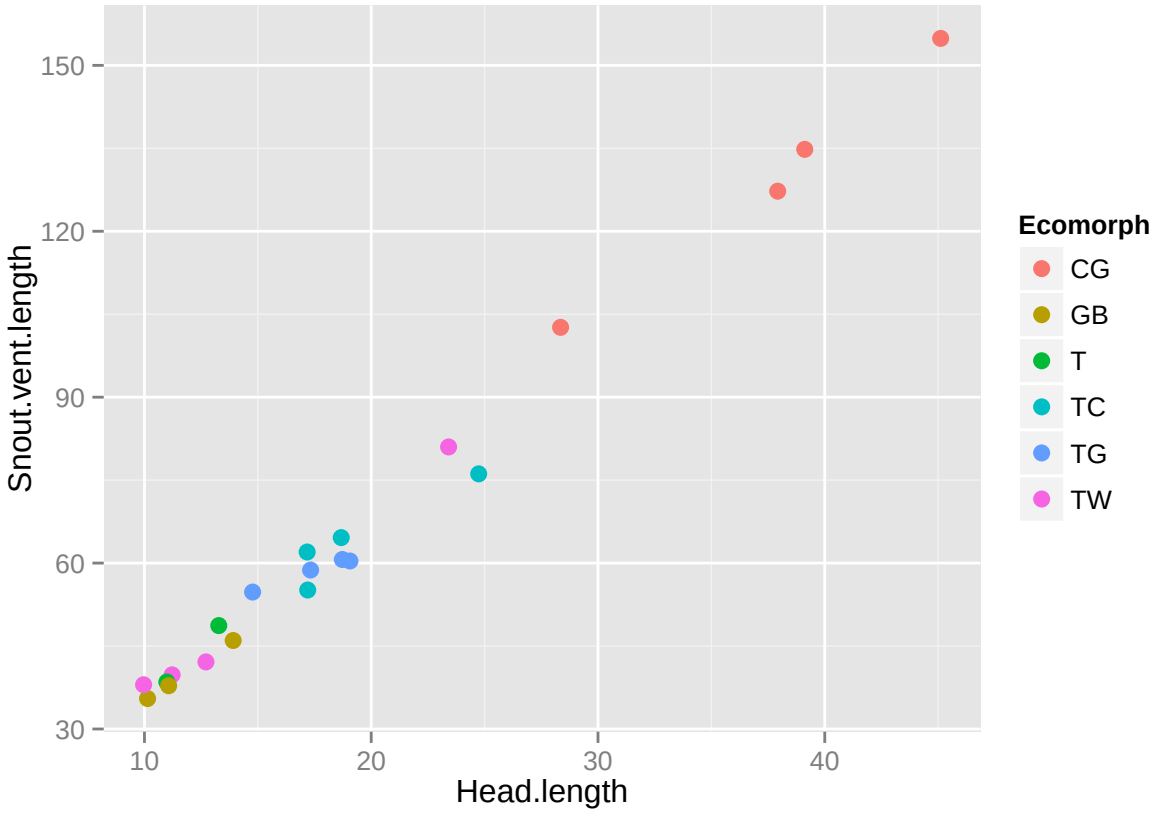
rOpenSci is an inititative to create R packages for accessing data repositories. The full list of packages is available here

```
# Here we will use the 'rdryad' package to retrieve a dataset from Dryad
# Retrieve the data using the Dryad identifier '10255/dryad.34389'
# which is at the end of the URL where the dataset is found
# http://datadryad.org/handle/10255/dryad.34389
# the original publication is Kolbe et al. 2011, Evolution 65(12): 3608-3624
anolis.data <- download_url("10255/dryad.34389")
anolis.data <- dryad_getfile(anolis.data)
# create plot with discrete and continous variables
```

```
anolis.scatter <- ggplot(anolis.data, aes(x=Ecomorph, y=Snout.vent.length,
  colour=Snout.vent.length)) + geom_point(size=3)
anolis.scatter
```




```
# create plot with continuous variables
anolis.scatter <- ggplot(anolis.data, aes(x=Head.length, y=Snout.vent.length,
  colour=Ecomorph)) + geom_point(size=3)
anolis.scatter
```



4 ggplot and rgbif

4.1 Plot GBIF occurrences of a species

```
# search for occurrences on GBIF
?occ_search
apicea.occ <- occ_search(scientificName="Aphaenogaster picea", limit=1000,
  return='data', hasCoordinate=TRUE)
# this dataset includes more than 100 columns
# show first 10 lines for columns 1 to 4
head(apicea.occ, n=10L)[,1:4]
```

```
##                    name        key decimalLatitude decimalLongitude
## 1  Aphaenogaster picea  899067422        41.98968        -71.78827
## 2  Aphaenogaster picea  899067424        41.98968        -71.78827
## 3  Aphaenogaster picea  899067434        41.98968        -71.78827
## 4  Aphaenogaster picea  910640573        41.63587        -71.55863
## 5  Aphaenogaster picea  910640541        41.63587        -71.55863
## 6  Aphaenogaster picea  910640582        41.63587        -71.55863
## 7  Aphaenogaster picea  910640546        41.63587        -71.55863
## 8  Aphaenogaster picea  910640559        41.63587        -71.55863
## 9  Aphaenogaster picea  891147998        36.89260        -82.62960
## 10 Aphaenogaster picea 1142484688        34.76128        -84.70864
```

```r
# select four columns from the complete data set
apicea.lat.lon <- apicea.occ[,c("name", "decimalLatitude", "decimalLongitude",
  "countryCode")]
# show the unique country codes in the data set
unique(apicea.lat.lon$countryCode)
```

```
## [1] "US" "MX"
```

```r
# select only US occurrences
apicea.lat.lon <- subset(apicea.lat.lon, countryCode=="US")
summary(apicea.lat.lon)
```

```
##      name           decimalLatitude decimalLongitude countryCode
##  Length:300         Min.   :33.49   Min.   :-94.25   Length:300
##  Class :character   1st Qu.:38.23   1st Qu.:-88.51   Class :character
##  Mode  :character   Median :40.11   Median :-83.89   Mode  :character
##                     Mean   :39.79   Mean   :-83.52
##                     3rd Qu.:41.50   3rd Qu.:-79.32
##                     Max.   :45.57   Max.   :-70.04
```

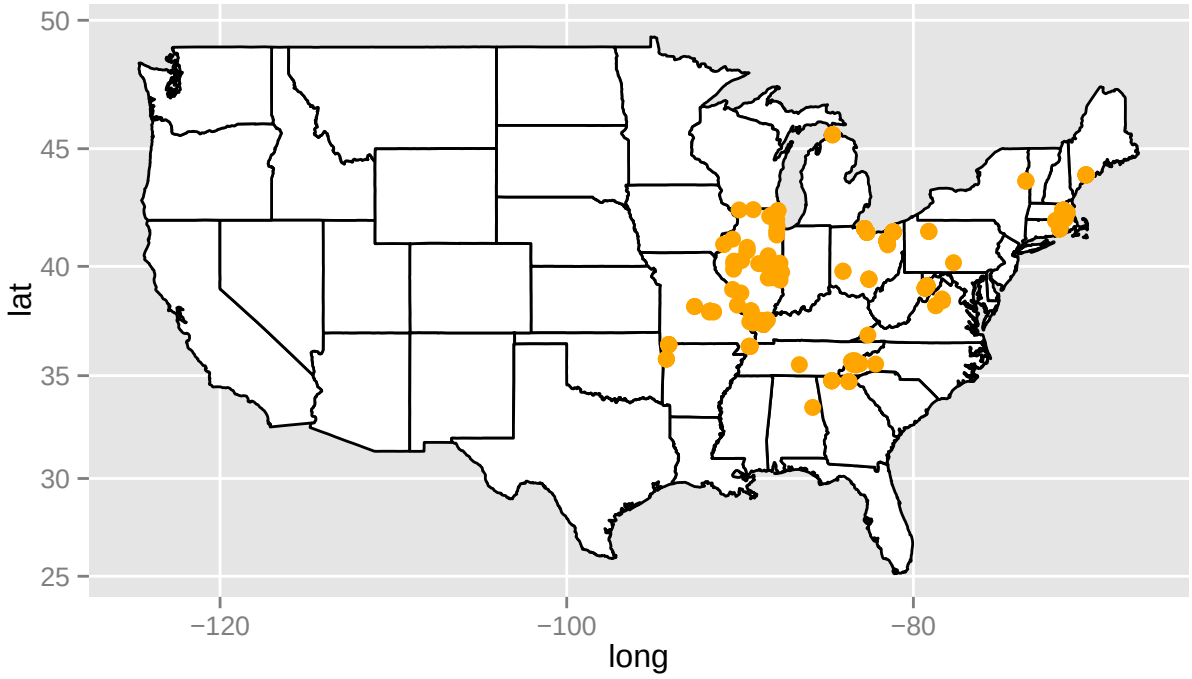
```r
# get map data for world
#world_map <- map_data("world")

# show unique regions
#sort(unique(world_map$region))

# get US map data
states.map <- map_data("state")
head(states.map)
```

```
##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

```r
#states.map <- subset(states.map, long > -100 & lat < 50)
states.map <- ggplot(states.map, aes(x=long, y=lat, group=group)) +
  geom_polygon(fill="white", colour="black") + coord_map("mercator")
#states.map
states.map +
geom_point(aes(x=decimalLongitude, y=decimalLatitude, group=name),
  colour="orange", size=3, data=apicea.lat.lon)
```




```
#gbifmap(apicea.lat.lon)
```

4.2 Plot GBIF occurrences of two or more species

```
splist <- c("Bombus fraternus", "Bombus perplexus")
# apply a function over a list or vector with sapply
bombus.keys <- sapply(splist, function(x) name_backbone(name=x)$speciesKey,
  USE.NAMES=FALSE)
bombus.occ <- occ_search(taxonKey=bombus.keys, limit=500, return="data",
  hasCoordinate=TRUE)
summary(bombus.occ)
```

```
##         Length Class      Mode
## 1340432 107    data.frame list
## 1340406 122    data.frame list
```

```
head(bombus.occ$`1340432`, n=10L)[,1:4]
```

```
##                name        key decimalLatitude decimalLongitude
## 1  Bombus fraternus 1098920735        33.72666        -95.68521
## 2  Bombus fraternus 1143521876        33.53094        -95.91432
## 3  Bombus fraternus  899954560        32.87881        -93.81990
```

```
## 4  Bombus fraternus  923923296        33.51451        -95.91873
## 5  Bombus fraternus  923923624        33.48436        -95.99989
## 6  Bombus fraternus  923923680        33.48885        -95.97723
## 7  Bombus fraternus 1065596357        33.69256        -96.76983
## 8  Bombus fraternus 1143569911        33.48602        -95.95263
## 9  Bombus fraternus 1024185501        33.55605        -95.92782
## 10 Bombus fraternus 1024191102        32.71433        -97.46752
```

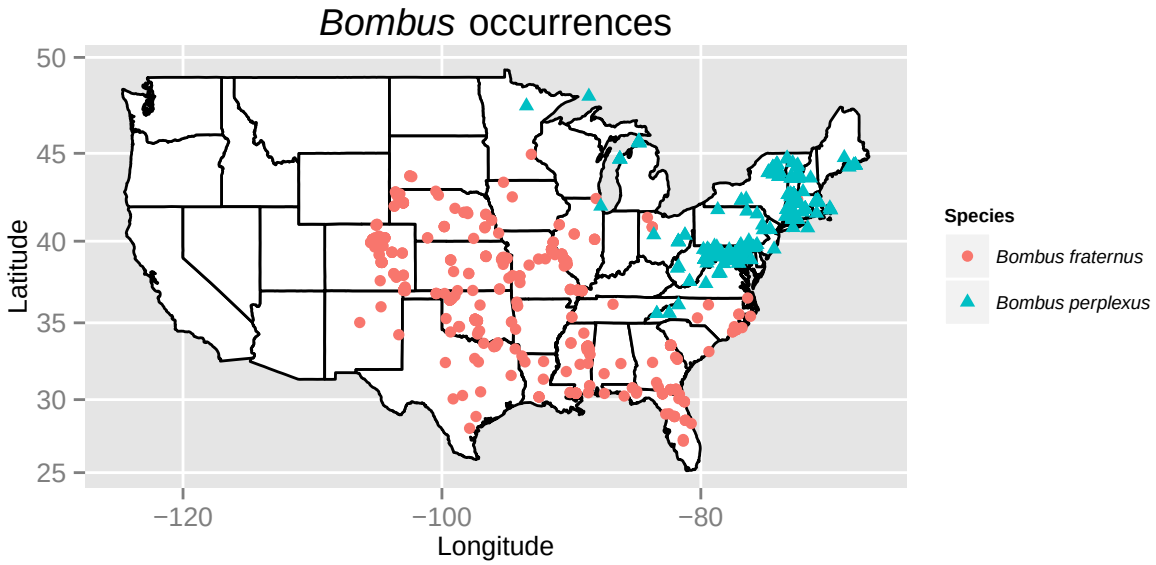
```
bfraternus.lat.lon <- bombus.occ$`1340432`[,c("name", "decimalLatitude",
  "decimalLongitude", "countryCode")]
head(bombus.occ$`1340406`, n=10L)[,1:4]
```

```
##                name        key decimalLatitude decimalLongitude
## 1  Bombus perplexus 1143520430        46.64574        -60.95197
## 2  Bombus perplexus 1143520142        44.23823        -68.54755
## 3  Bombus perplexus 1143521611        43.70909        -73.03349
## 4  Bombus perplexus 1143532609        44.35068        -72.51181
## 5  Bombus perplexus 1122964996        44.69461        -73.34100
## 6  Bombus perplexus 1135213313        44.73630        -73.33179
## 7  Bombus perplexus 1143530758        44.69111        -73.34442
## 8  Bombus perplexus 1135350545        45.65765        -65.01472
## 9  Bombus perplexus 1143522783        46.71553        -60.39149
## 10 Bombus perplexus 1136227380        39.34410        -77.75340
```

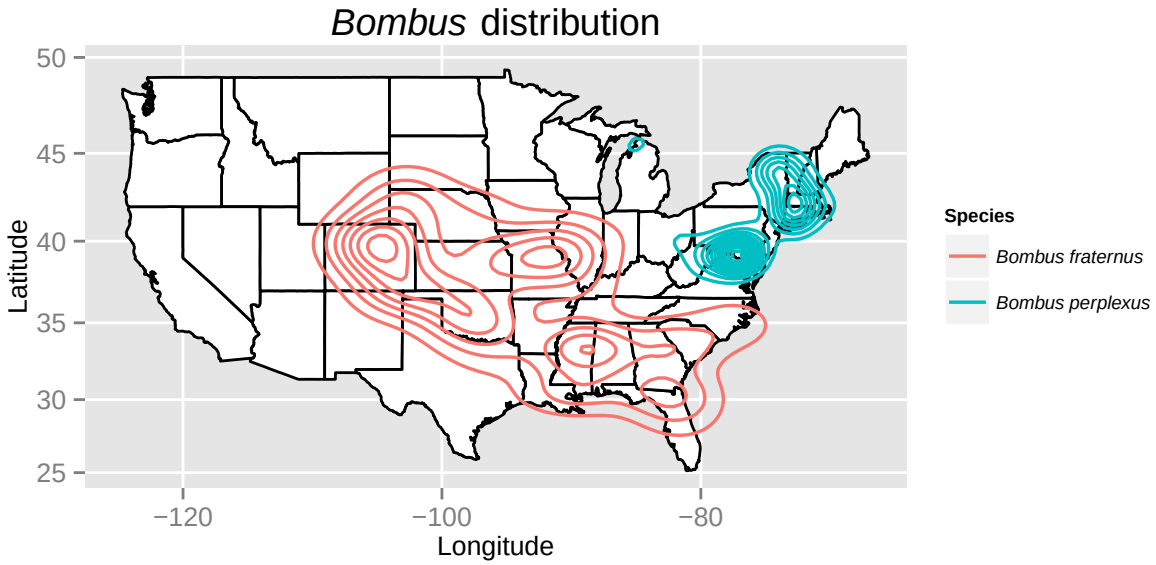
```
bperplexus.lat.lon <- bombus.occ$`1340406`[,c("name", "decimalLatitude",
  "decimalLongitude", "countryCode")]
bperplexus.lat.lon <- subset(bperplexus.lat.lon, decimalLatitude < 50)
bombus.lat.lon <- rbind(bfraternus.lat.lon, bperplexus.lat.lon)
bombus.lat.lon <- subset(bombus.lat.lon, countryCode=="US")
states.map <- map_data("state")
#states.map <- subset(states.map, long > -120 & lat < 50)
states.map <- ggplot(states.map, aes(x=long, y=lat, group=group)) +
  geom_polygon(fill="white", colour="black") + coord_map("mercator")
#states.map

# create a title for ggplot that italicizes only the genus name
bombus.occ.title <- expression(paste(italic("Bombus"), " occurrences"))
bombus.map.occ <- geom_point(aes(x=decimalLongitude, y=decimalLatitude,
  group=name, colour=name, shape=name), size=2, data=bombus.lat.lon)
bombus.maplabels <- labs(title=bombus.occ.title, x="Longitude", y="Latitude",
  colour="Species", shape="Species")
bombus.maplegend <- theme(plot.title=element_text(face="italic",
  size=rel(1.2))) +
  theme(legend.title=element_text(size=rel(0.6))) +
  theme(legend.text=element_text(face="italic", size=rel(0.6))) +
  theme(axis.title.x=element_text(size=rel(0.8))) +
  theme(axis.title.y=element_text(size=rel(0.8)))

states.map + bombus.map.occ + bombus.maplabels + bombus.maplegend
```



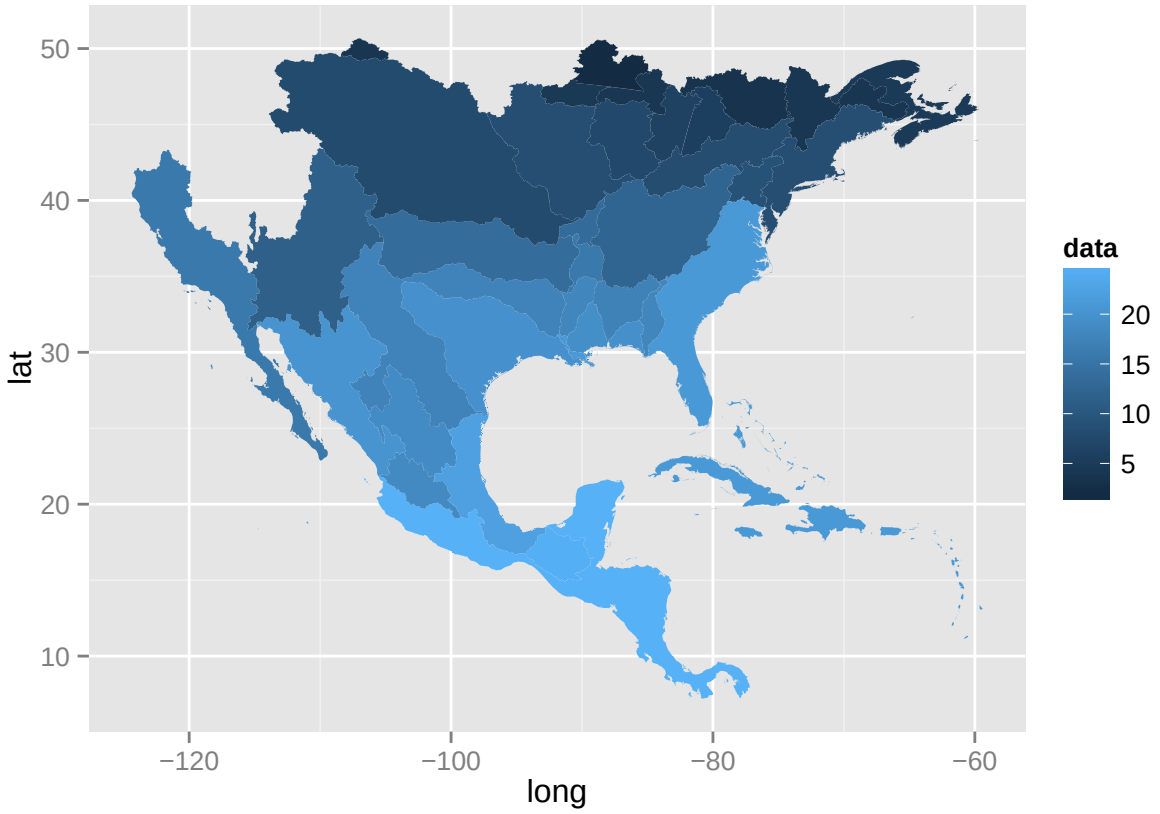

```
bombus.map.den <- stat_density2d(aes(x=decimalLongitude, y=decimalLatitude,
  group=name, colour=name), size=0.6, data=bombus.lat.lon, geom="density2d")
bombus.den.title <- expression(paste(italic("Bombus"), " distribution"))
bombus.maplabels <- labs(title=bombus.den.title, x="Longitude", y="Latitude",
  colour="Species", shape="Species")
states.map + bombus.map.den + bombus.maplabels + bombus.maplegend
```



4.3 Map environmental data

```
# Create a directory for kml files
dir.create("~/Desktop/kmltemp")
options(kmlpath="~/Desktop/kmltemp")
#http://data.worldbank.org/developers/climate-data-api
#http://unstats.un.org/unsd/methods/m49/m49alpha.htm
#http://data.worldbank.org/sites/default/files/climate_data_api_basins.pdf
# Create a vector of World Bank basin IDs
nam <- 328:365
# Download map for vector of basins
nam.basin <- create_map_df(nam)
```

```
# Retrieve historical precipitation data
temp.dat <- get_historical_temp(nam, "decade")
# Create a subset using only data from one year
temp.dat <- subset(temp.dat, temp.dat$year==2000)
# Create maps of climate data
nam.map <- climate_map(nam.basin, temp.dat, return_map=TRUE)
nam.map
```

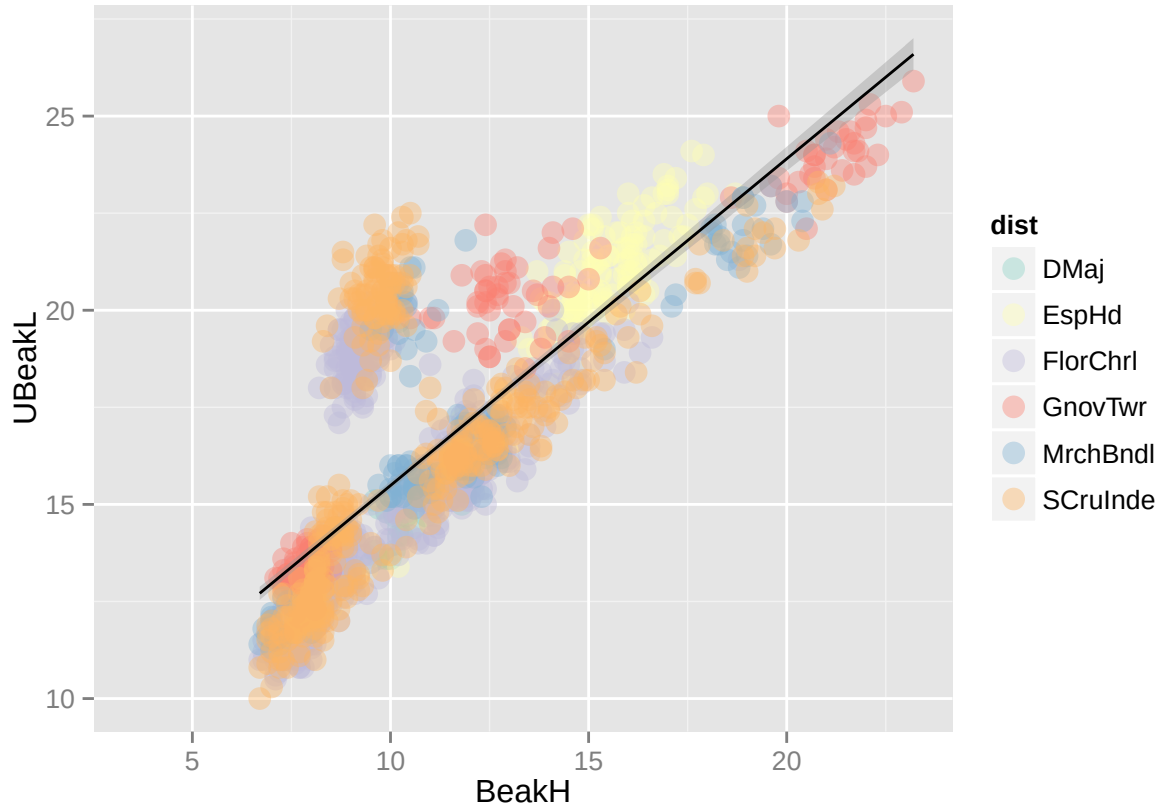


5 Other examples using morphological data for Darwin's finches

```
data(finch.ind)
tr <- traits.finch; sp <- sp.finch; dist <- ind.plot.finch
tr$sp <- sp; tr$dist <- dist
head(tr)
```

```
##     WingL BeakH UBeakL N.UBkL              sp dist
## 165    69  10.8   15.2   10.6 Geospiza_fortis DMaj
## 166    65  10.3   14.2    9.8 Geospiza_fortis DMaj
## 167    65  10.0   13.6    9.9 Geospiza_fortis DMaj
## 168    68   9.5   13.5    9.9 Geospiza_fortis DMaj
## 169    66  11.0   15.0   10.7 Geospiza_fortis DMaj
## 170    68  10.4   15.0   10.2 Geospiza_fortis DMaj
```

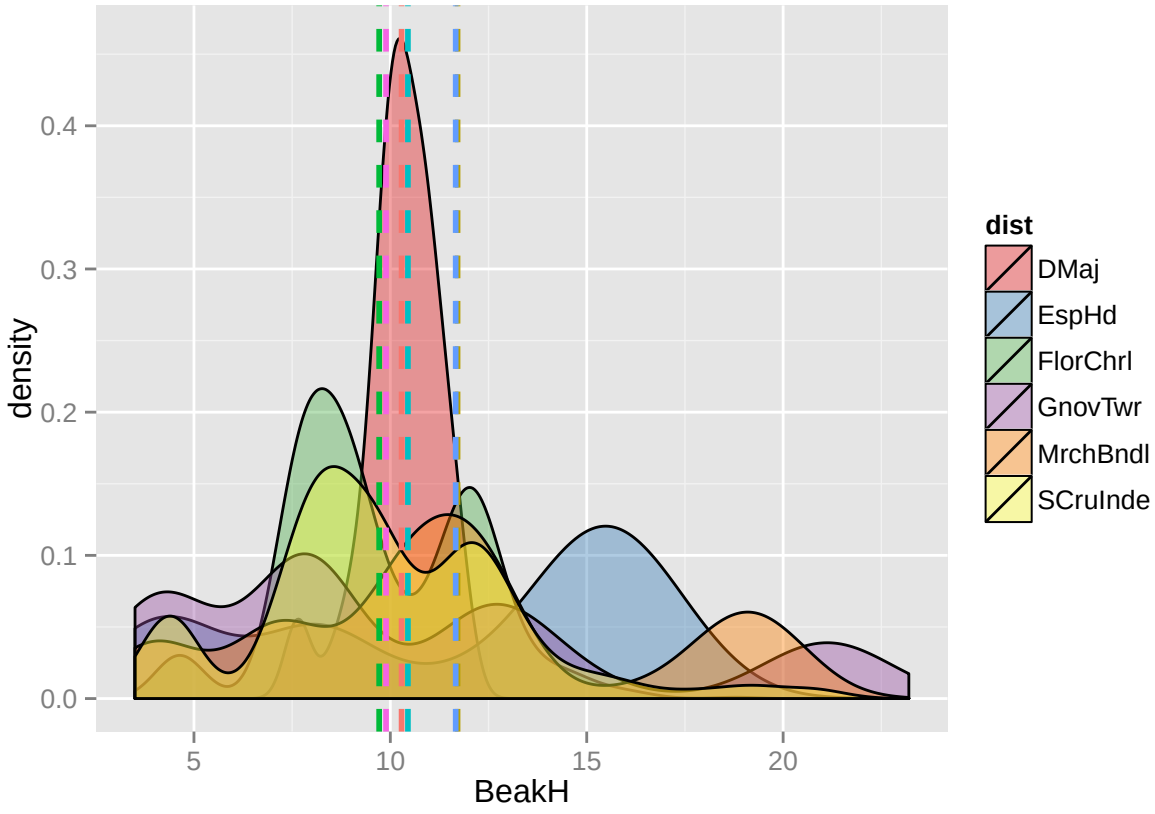
```
ggplot(tr, aes(x=BeakH, y=UBeakL, colour=dist)) +
  geom_point(size=4, alpha=0.4) +
  stat_smooth(method=lm, colour="black", level=0.95) +
  scale_colour_brewer(palette="Set3")
```




```
mbh <- ddply(tr, "dist", summarise, bh.mean=mean(BeakH, na.rm=TRUE))
mbh
```

```
##       dist   bh.mean
## 1     DMaj 10.286047
## 2    EspHd 11.711345
## 3 FlorChrl  9.715032
## 4  GnovTwr 10.447945
## 5 MrchBndl 11.663877
## 6 SCruInde  9.890248
```

```
ggplot(tr, aes(x=BeakH, fill=dist)) + geom_density(alpha=0.4) +
  scale_fill_brewer(palette="Set1") +
  geom_vline(data=mbh, aes(xintercept=bh.mean, colour=dist),
    linetype="dashed", size=1)
```


density
0.4
0.3
0.2
0.1
0.0
5
10
15
20
BeakH
dist
DMaj
EspHd
FlorChrl
GnovTwr
MrchBndl
SCruInde