LSC 563 Lecture 10: Visualizing Associations Among Two or More Quantitative Variables

Doug Joubert

Table of Contents

[Introduction 2](#_Toc115702788)

[Learning Objectives 2](#_Toc115702789)

[Bivariate Data 2](#_Toc115702790)

[Blue Jay Data 3](#_Toc115702791)

[Scatterplots 6](#_Toc115702792)

[Scatterplot Tasks 7](#_Toc115702793)

[Bubble Graph 10](#_Toc115702794)

[Linear Relationships and Regression 13](#_Toc115702795)

[Simple Linear Regression 13](#_Toc115702796)

[Correlation Coefficient 18](#_Toc115702797)

[Correlation Matrix (Correlogram) 20](#_Toc115702798)

[Paired data 24](#_Toc115702799)

[Handling Overlapping Points 25](#_Toc115702800)

[Partial Transparency and Jittering 25](#_Toc115702801)

[References 33](#_Toc115702802)

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(RColorBrewer)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

# Introduction

Many datasets contain two or more quantitative variables, and we may be interested in how these variables relate to each other. To plot the relationship of just two such variables, we will normally use a scatterplot.

If we want to show more than two variables at once, we may opt for a bubble chart, a scatterplot matrix, or a correlogram.

For very high-dimensional datasets, it may be useful to perform dimension reduction, for example in the form of principal components analysis.

## Learning Objectives

1. Define “bivariate data”
2. Define “scatter plot”
3. Distinguish between a linear and a nonlinear relationship
4. Identify positive and negative associations from a scatter plot
5. Describe what Pearson’s correlation measures
6. Give the symbols for Pearson’s correlation in the sample and in the population
7. State the possible range for Pearson’s correlation
8. Identify a perfect linear relationship

# Bivariate Data

A dataset with two variables contains called bivariate data (D. M. Lane, 2007). This section discusses ways to describe the relationship between two variables. For example, you may wish to describe the relationship between the heights and weights of people to determine the extent to which taller people weigh more.

The introductory section gives more examples of bivariate relationships and presents the most common way of portraying these relationships graphically. The next section discusses Pearson’s correlation, the most common index of the relationship between two variables. If you need a refresher, you might want to review the following chapters in Lane: [Variables](https://onlinestatbook.com/2/introduction/variables.html), [Distributions](https://onlinestatbook.com/2/introduction/distributions.html), [Histograms](https://onlinestatbook.com/2/graphing_distributions/histograms.html), [Measures of Central Tendency](https://onlinestatbook.com/2/summarizing_distributions/measures.html), [Variability](https://onlinestatbook.com/2/summarizing_distributions/variability.html), [Shape](https://onlinestatbook.com/2/summarizing_distributions/shapes.html)

As we have already learned, measures of central tendency, variability, and spread summarize a single variable by providing important information about its distribution (D. M. Lane, 2007). Often, more than one variable is collected on each individual. For example, in large health studies of populations it is common to obtain variables such as age, sex, height, weight, blood pressure, and total cholesterol on each individual (D. M. Lane, 2007).

In this section we consider bivariate data, which for now consists of two [quantitative variables](javascript:glossary('quantitative_variable')) for each individual (D. M. Lane, 2007). Our first interest is in summarizing such data in a way that is analogous to summarizing univariate (single variable) data.

## Blue Jay Data

Contains measurements performed on 123 blue jay birds. The dataset contains information such as the head length (measured from the tip of the bill to the back of the head), the skull size (head length minus bill length), and the body mass of each bird.

We expect that there are relationships between these variables.

blue\_jays <- read\_csv("../raw\_data/blue\_jays.csv")

## Rows: 123 Columns: 8  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): bird\_id, sex  
## dbl (6): bill\_depth\_mm, bill\_width\_mm, bill\_length\_mm, head\_length\_mm, body\_...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

str(blue\_jays)

## spec\_tbl\_df [123 × 8] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ bird\_id : chr [1:123] "0000-00000" "1142-05901" "1142-05905" "1142-05907" ...  
## $ sex : chr [1:123] "M" "M" "M" "F" ...  
## $ bill\_depth\_mm : num [1:123] 8.26 8.54 8.39 7.78 8.71 7.28 8.74 8.72 8.2 7.67 ...  
## $ bill\_width\_mm : num [1:123] 9.21 8.76 8.78 9.3 9.84 9.3 9.28 9.94 9.01 9.31 ...  
## $ bill\_length\_mm: num [1:123] 25.9 25 26.1 23.5 25.5 ...  
## $ head\_length\_mm: num [1:123] 56.6 56.4 57.3 53.8 57.3 ...  
## $ body\_mass\_g : num [1:123] 73.3 75.1 70.2 65.5 74.9 ...  
## $ skull\_size\_mm : num [1:123] 30.7 31.4 31.2 30.3 31.9 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. bird\_id = col\_character(),  
## .. sex = col\_character(),  
## .. bill\_depth\_mm = col\_double(),  
## .. bill\_width\_mm = col\_double(),  
## .. bill\_length\_mm = col\_double(),  
## .. head\_length\_mm = col\_double(),  
## .. body\_mass\_g = col\_double(),  
## .. skull\_size\_mm = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

head(blue\_jays, 5)

## # A tibble: 5 × 8  
## bird\_id sex bill\_depth\_mm bill\_width\_mm bill\_le…¹ head\_…² body\_…³ skull…⁴  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0000-00000 M 8.26 9.21 25.9 56.6 73.3 30.7  
## 2 1142-05901 M 8.54 8.76 25.0 56.4 75.1 31.4  
## 3 1142-05905 M 8.39 8.78 26.1 57.3 70.2 31.2  
## 4 1142-05907 F 7.78 9.3 23.5 53.8 65.5 30.3  
## 5 1142-05909 M 8.71 9.84 25.5 57.3 74.9 31.8  
## # … with abbreviated variable names ¹​bill\_length\_mm, ²​head\_length\_mm,  
## # ³​body\_mass\_g, ⁴​skull\_size\_mm

Table 1 shows blue jay data.

Let us explore the blue jay data uses histograms. Let us first look at the body mass variables. This [page](https://homepage.divms.uiowa.edu/~luke/classes/STAT4580/histdens.html) has more information about using density functions in geom\_histogram.

bj\_body\_mass\_hist <- blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g)) +  
 geom\_histogram(aes(y = ..density..), color = "black", fill = "white") +  
 geom\_density(alpha = .2, fill = "#91918d") +  
 geom\_vline(mapping = aes(xintercept = mean(body\_mass\_g, na.rm = TRUE)), color = "red", linetype = "dashed", size = 1)  
bj\_body\_mass\_hist

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

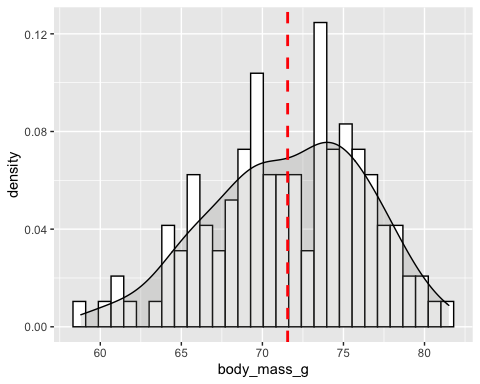


Figure 10.1: Histogram of the body mass variable.

Let us save this as a high-res image

ggsave("../figures/bj\_body\_mass\_hist.pdf", plot = bj\_body\_mass\_hist)

## Saving 5 x 4 in image

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Now, let us look at a histogram of the head length variable [Figure 10.2].

bj\_head\_length\_hist <- blue\_jays %>%   
 ggplot(mapping = aes(x = head\_length\_mm)) +  
 geom\_histogram(aes(y = ..density..), color = "black", fill = "white") +  
 geom\_density(alpha = .2, fill = "#91918d") +  
 geom\_vline(mapping = aes(xintercept = mean(head\_length\_mm, na.rm = TRUE)), color = "red", linetype = "dashed", size = 1)  
bj\_head\_length\_hist

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

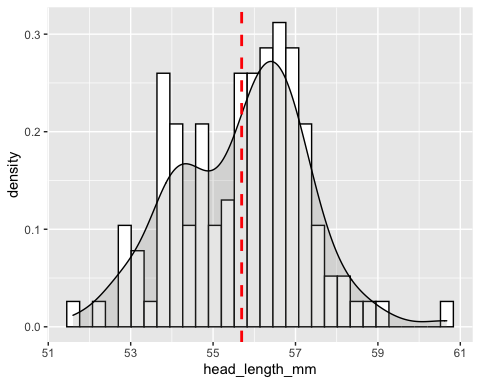


Figure 10.2: Histogram of the head length variable.

ggsave("../figures/bj\_head\_length\_hist.pdf", plot = bj\_head\_length\_hist)

## Saving 5 x 4 in image

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

In contrast to the histogram of the body mass variable, this data appears to be normally distributed.

We can learn much more by displaying the bivariate data in a graphical form that maintains the pairing (D. M. Lane, 2007). This graph types is called a scatterplot.

# Scatterplots

Scatterplot are designed to emphasize the spatial distribution of data plotted in two-dimensions (Sarikaya & Gleicher, 2018). Cleveland notes three factors that may affect the design decisions that are made by the designer of a scatterplot (Cleveland & McGill, 1984):

1. Marks or points are designed with preattentive features in mind,
2. Scatterplots are designed with the detection of individual objects in mind
3. Scatterplots are designed such that the distances between objects represent a notion of similarity

## Scatterplot Tasks

A primary consideration common in many information visualization taxonomies is task, abstraction, or how a viewer interacts with and obtains information from a visualization (Munzner, 2014).

Figure 10.3 lists scatterplot tasks they identified (Sarikaya & Gleicher, 2018).

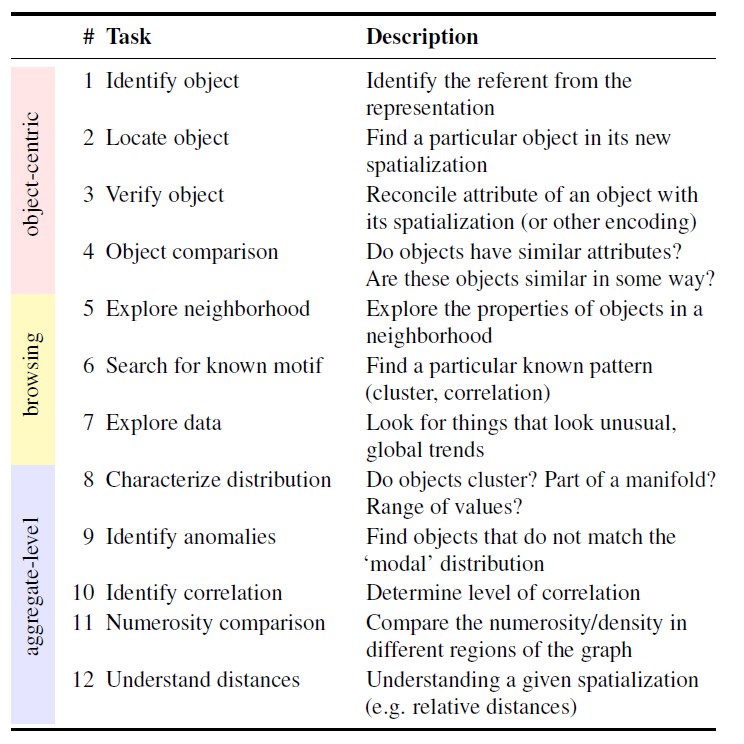


Figure 10.3: List of abstracted analysis tasks that are performed with scatterplots.

Now, let us look at the blue jay data as a scatterplot. Figure 10.4 is a plot of head length against body mass. In this plot, head length is shown along the y-axis and body mass along the x axis, and each bird is represented by one dot [Figure 10.4].

Note the terminology: we say that we plot the variable shown along the y-axis against the variable shown along the x-axis.

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = head\_length\_mm)) +  
 geom\_point(size = 1.5) +  
 labs(y = "Head length in millimeters", x = "Body mass in grams")

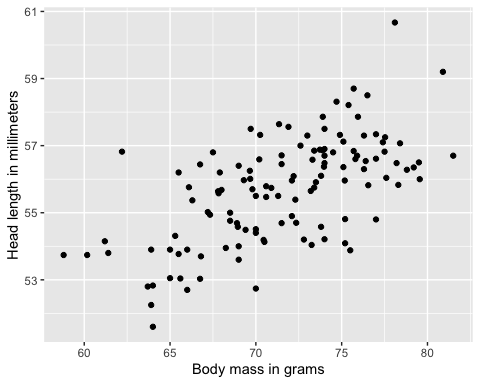


Figure 10.4: Head length (measured from the tip of the bill to the back of the head, in mm) versus body mass (in grams),for 123 blue jays. Each dot corresponds to one bird. Data source: Keith Tarvin, Oberlin College.

From Figure 10.4, we can see There is a moderate tendency for heavier birds to have longer heads.

The blue jay dataset contains both male and female birds, so we may want to look at the relationship between head length and body mass, by sex.

To address this question, we can color the points in the scatterplot by the sex of the bird. I did not like the default colors that R provided so I used Set1 from the color brewer package [Figure 10.5].

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = head\_length\_mm,   
 color = sex)) +  
 geom\_point(size = 1.5) +  
 scale\_color\_brewer(palette="Set1") +  
 labs(y = "Head length in millimeters", x = "Body mass in grams")

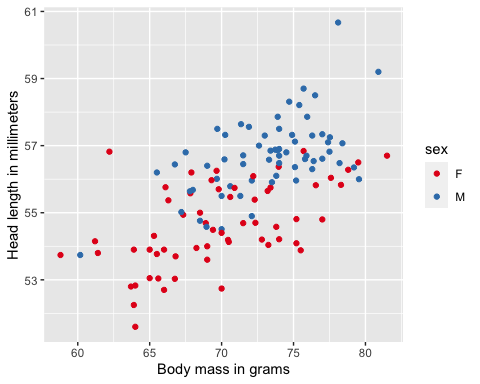


Figure 10.5: Head length versus body mass for 123 blue jays. The birds’ sex is indicated by color.

This figure reveals that the overall trend in head length and body mass is at least in part driven by the sex of the birds. Meaning that at the same body mass, male birds tend to have longer heads than female birds.

Because the head length is defined as the distance from the tip of the bill to the back of the head, a larger head length could imply a longer bill, a larger skull, or both.

We can disentangle bill length and skull size by looking at another variable in the dataset, the skull size, which is like the head length, but excludes the bill [Figure 10.6].

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = bill\_length\_mm,   
 color = sex)) +  
 geom\_point(size = 1.5) +  
 scale\_color\_brewer(palette="Set1") +  
 labs(y = "Bill length in millimeters", x = "Body mass in grams")

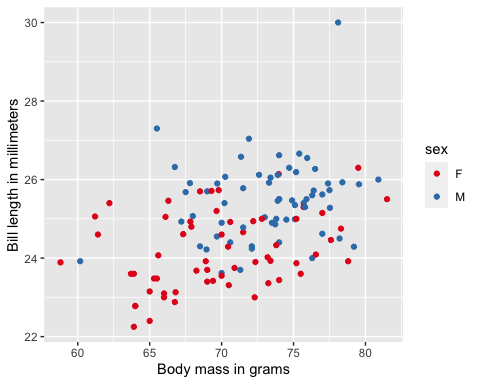


Figure 10.6: Bill length versus body mass for 123 blue jays. The birds’ sex is indicated by color.

# Bubble Graph

If we wanted to look at head size and bill size, we can do that too. As we are already using the x position for body mass, the position for head length, and the dot color for bird sex, we need another aesthetic to which we can map skull size.

What might be another channel that we can use for this graph?

A Bubble Chart is a multi-variable graph that is a cross between a [Scatterplot](https://datavizcatalogue.com/methods/scatterplot.html) and a [Proportional Area Chart](https://datavizcatalogue.com/methods/area_chart.html).

Like a Scatterplot, Bubble Charts use a Cartesian coordinate system to plot points along a grid where the X and Y axis are separate variables (Ribecca, 2019). However. unlike a Scatterplot, each point is assigned a label or category (either displayed alongside or on a legend). Each plotted point then represents a third variable by the area of its circle (Ribecca, 2019).

Figure 10.7 is showing the blue jay data as a bubble graph. Not sure how Wilke created this chart, I used [facet\_wrap](https://ggplot2.tidyverse.org/reference/facet_wrap.html).

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = head\_length\_mm,   
 size = skull\_size\_mm, color = sex)) +  
 geom\_point() +  
 scale\_size(range = c(.1, 5), name= "Skull Size (mm)") +   
 facet\_wrap(vars(sex)) +  
 scale\_color\_brewer(palette="Set1") +  
 labs(y = "Head length (mm)", x = "Body mass (g)") +  
 guides(col = FALSE) +  
 theme(legend.position="bottom")

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.

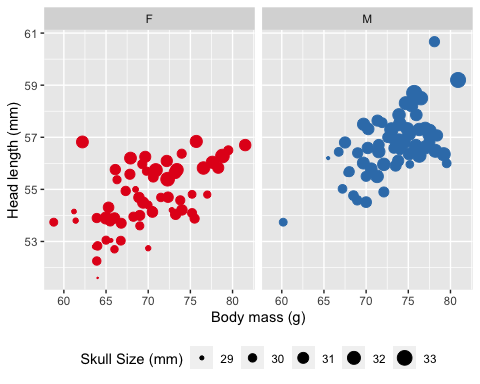


Figure 10.7: Head length versus body mass for 123 blue jays. Data source: Keith Tarvin, Oberlin College

The birds’ sex is indicated by color, and the birds’ skull size by symbol size. Head-length measurements include the length of the bill while skull-size measurements do not. Head length and skull size tend to be correlated, but there are some birds with unusually long or short bills given their skull size.

Tip: This [page](https://statisticsglobe.com/remove-legend-ggplot2-r) has guidance on how to only display certain legends. You can change the position of the legend using these [guidelines](sthda.com/english/wiki/ggplot2-legend-easy-steps-to-change-the-position-and-the-appearance-of-a-graph-legend-in-r-software#change-the-legend-position).

Bubble charts have the disadvantage that they show the same types of variables, quantitative variables, with two different types of scales, position and size. As we have already learned, this makes it difficult to visually ascertain the strengths of associations between the various variables. Moreover, differences between data values encoded as bubble size are harder to perceive than differences between data values encoded as position.

Like with Proportional Area Charts, the sizes of the circles need to be drawn based on the circle’s area, not its radius or diameter. Not only will the size of the circles change exponentially, but this will lead to misinterpretations by the human visual system.

As an alternative to a bubble chart, it may be preferable to show an all-against-all matrix of scatter plots, where each individual plot shows two data dimensions. Figure 10.8 is displaying an all-against-all scatter plot matrix of head length, body mass, and skull size, for 123 blue jays.

 Figure 10.8: Scatterplot matrix of head length, body mass, and skull size. Data source: Keith Tarvin, Oberlin College

This figure shows the exact same data as Figure 10.7. However, because we are better at judging position than symbol size, correlations between skull size and the other two variables are easier to perceive in the pairwise scatter plots than in Figure 10.7.

The code for generating a scatterplot matrix is complex and beyond the scope of this class. However, I still wanted you to see what one looks like.

# Linear Relationships and Regression

## Simple Linear Regression

In a simple linear regression, we predict values on one variable from the values on a second variable (D. Lane, 2007):

* Variable we are predicting is referred to as Y (D. Lane, 2007)
* Variable we are basing our predictions on is referred to as X (D. Lane, 2007)
* When there is only one predictor variable, the prediction method is called simple regression (D. Lane, 2007)
* Simple linear regression, the predictions of Y when plotted as a function of X form a straight line (D. Lane, 2007).

Let us look at some dummy data.

| X | Y |
| --- | --- |
| 1.00 | 1.00 |
| 2.00 | 2.00 |
| 3.00 | 1.30 |
| 4.00 | 3.75 |
| 5.00 | 2.25 |

The figure below is the plotted data [Figure 10.9].

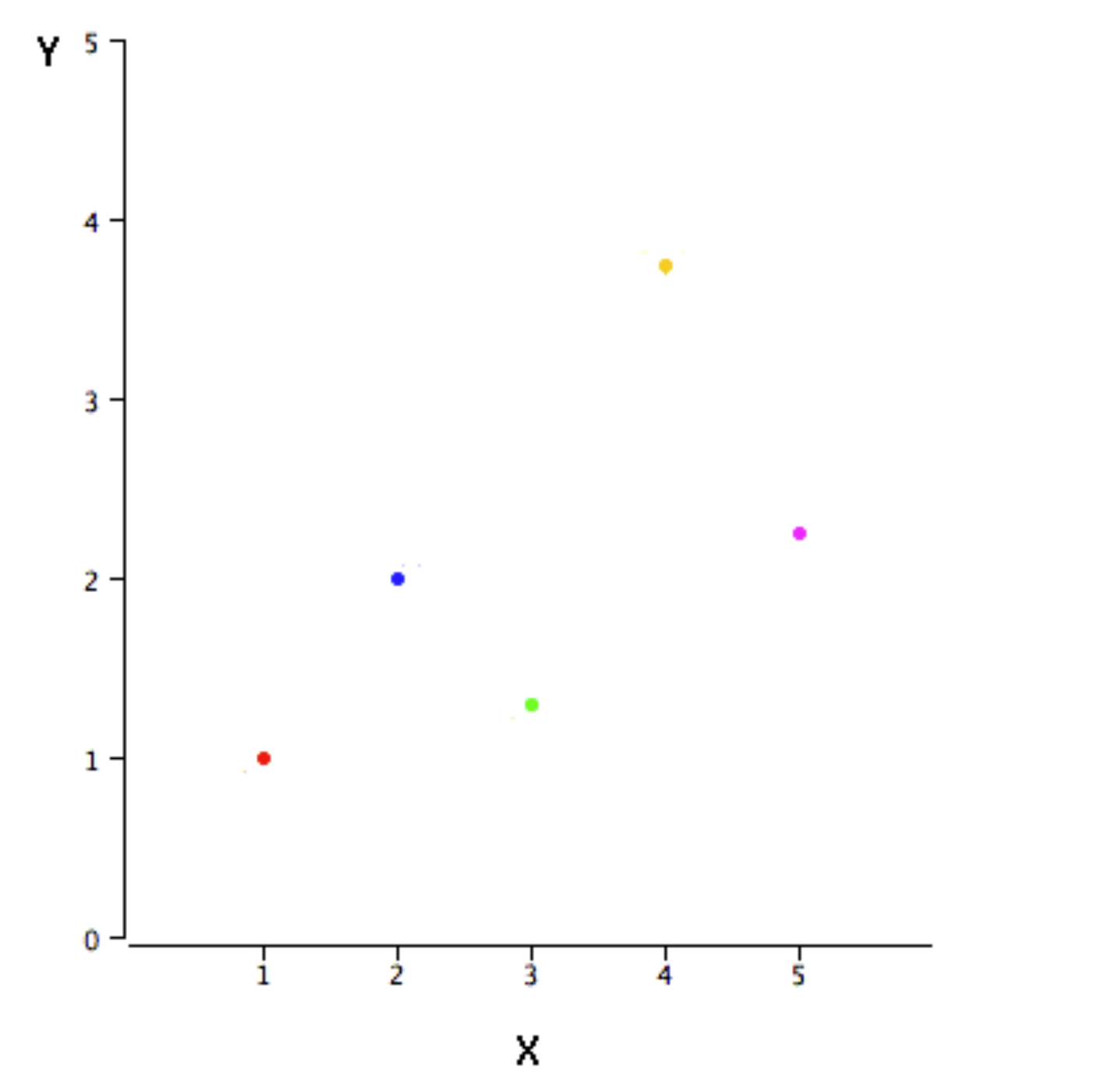


Figure 10.9: Plotted sample data.

Linear regression consists of finding the best-fitting straight line through the points. Lines or curves are fitted within the graph to aid in analysis and are drawn as close to all the points as possible and to show how all the points were condensed into a single line would look.

You may have noticed that I did not specify what is meant by “best-fitting line.” We will look at some options used by R, in the section below. You can read more about this in the [Lane, Chapter 14](https://onlinestatbook.com/2/regression/intro.html).

The black diagonal line in Figure 10.10 is the regression line and consists of the predicted score on Y for each possible value of X (D. Lane, 2007). The vertical lines from the points to the regression line represent the errors of prediction (D. Lane, 2007). As you can see, the red point is very near the regression line; its error of prediction is small (Osherson & Lane, 2007). By contrast, the yellow point is much higher than the regression line and therefore its error of prediction is large (D. Lane, 2007).

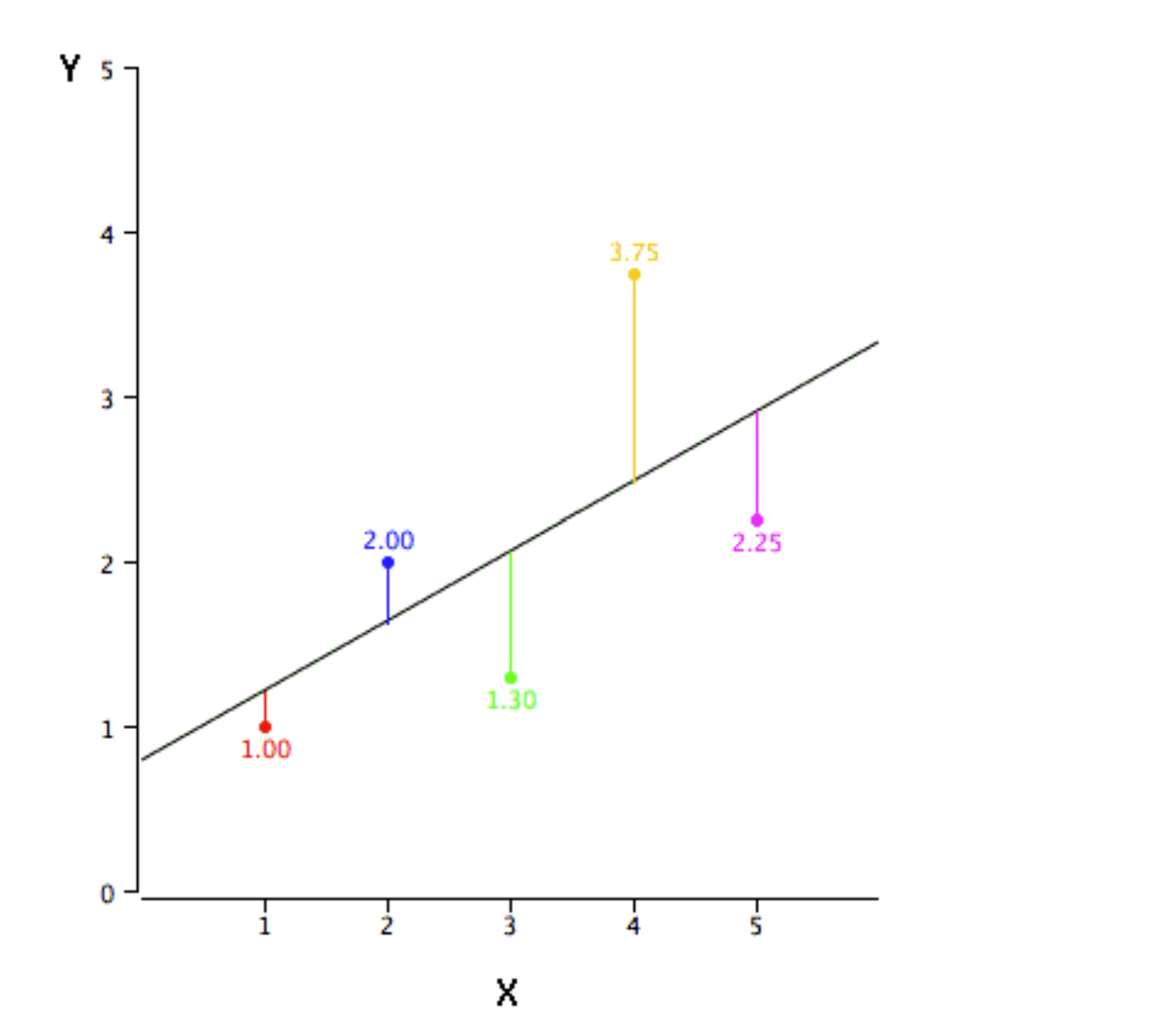
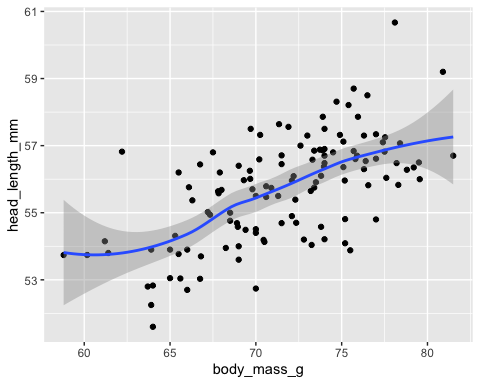


Figure 10.10: Scatterplot of the same data with the error lines.

Let us try this out with the blue jay data, focusing on body mass and head length [Figure 10.11].

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = head\_length\_mm)) +  
 geom\_point(size = 1.5) +  
 geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



labs(y = "Head length in millimeters", x = "Body mass in grams")

## $y  
## [1] "Head length in millimeters"  
##   
## $x  
## [1] "Body mass in grams"  
##   
## attr(,"class")  
## [1] "labels"

Figure 10.11: Head length versus body mass (in grams), for 123 blue jays. Each dot corresponds to one bird. I have added a smoothed line to the graph.

As we can see in Figure 10.11, this overlays the scatterplot with a smooth curve, including an assessment of uncertainty in the form of point-wise confidence intervals shown in grey (Wickham & Sievert, 2016). If you’re not interested in the confidence interval, turn it off with geom\_smooth(se = FALSE).

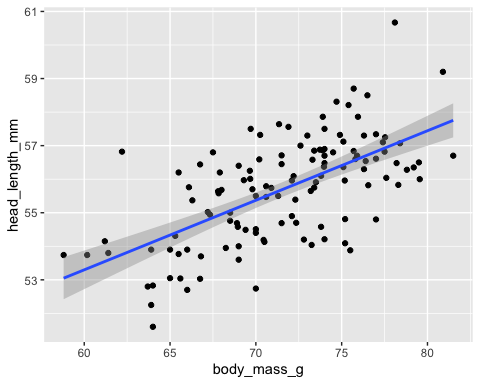
An important argument to [geom\_smooth()](https://ggplot2.tidyverse.org/reference/geom_smooth.html) is the method, which allows you to choose which type of model is used to fit the smooth curve:

* method = "loess", the default for small n, uses a smooth local regression (as described in [?loess](https://rdrr.io/r/stats/loess.html)). The wiggliness of the line is controlled by the span parameter, which ranges from 0 (exceedingly wiggly) to 1 (not so wiggly)

Loess does not work well for large datasets, so an alternative smoothing algorithm is used when n is greater than 1,000. method = "lm" fits a linear model, giving the line of best fit. Let us try this out and compare the two options [Figure 10.12].

blue\_jays %>%   
 ggplot(mapping = aes(x = body\_mass\_g, y = head\_length\_mm)) +  
 geom\_point(size = 1.5) +  
 geom\_smooth(method = lm)

## `geom\_smooth()` using formula 'y ~ x'



labs(y = "Head length in millimeters", x = "Body mass in grams")

## $y  
## [1] "Head length in millimeters"  
##   
## $x  
## [1] "Body mass in grams"  
##   
## attr(,"class")  
## [1] "labels"

Figure 10.12: Head length versus body mass (in grams), for 123 blue jays. Each dot corresponds to one bird. I have added a smoothed line to the graph using the lm method.

Below are some of the terms for linear regression. These are for reference only, you will not be tested on these terms.

| Term | Meaning |
| --- | --- |
| Response | The variable we are trying to predict. Synonyms: dependent variable, Y-variable, target, outcome |
| Independent variable | The variable used to predict the response. Synonyms: independent variable, X-variable, feature, attribute |
| Record | The vector of predictor and outcome values for a specific individual or case. Synonyms row, case, instance, example |
| Intercept | The intercept of the regression line — that is, the predicted value when x = 0 |
| Regression coefficient | The slope of the regression line. Synonyms slope, parameter estimates, weights |
| Fitted values | The estimates obtained from the regression line. Synonyms predicted values |
| Residuals | The difference between the observed values and the fitted values. Synonyms errors |
| Least squares | The method of fitting a regression by minimizing the sum of squared residuals. Synonyms ordinary least squares |

## Correlation Coefficient

The Pearson’s correlation or simply as the correlation coefficient is a measure of the strength of the linear relationship between two variables (D. M. Lane, 2007). There is a perfect linear relationship between two variables if a scatterplot of the points falls on a straight line (D. M. Lane, 2007). The relationship is linear even if the points diverge from the line as long as the divergence is random rather than being systematic (D. M. Lane, 2007). If the relationship between the variables is not linear, then the correlation coefficient does not adequately represent the strength of the relationship between the variables (D. M. Lane, 2007).

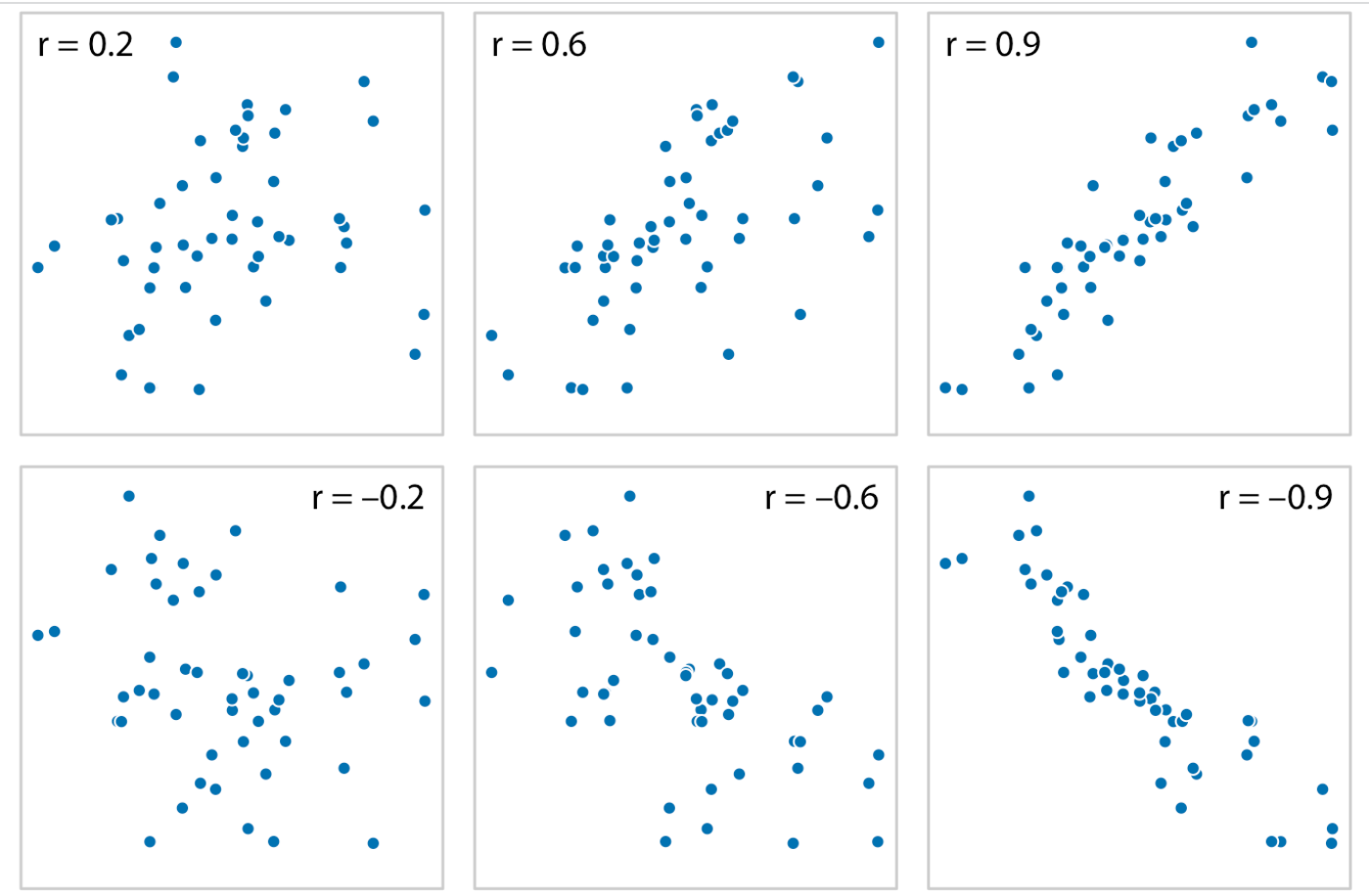
The symbol for Pearson’s correlation is “ρ” when it is measured in the population and “r” when it is measured in a sample (D. M. Lane, 2007). As a reminder, a population is the complete set of observations a researcher is interested in (D. M. Lane, 2007). Contrast this with a sample which is a subset of a population (D. M. Lane, 2007). A population can be defined in a manner convenient for a researcher. For example, one could define a population as all girls in fourth grade in Washington, DC (D. M. Lane, 2007).

Because we will be dealing almost exclusively with samples, we will use r to represent Pearson’s correlation.

Pearson’s r can range from -1 to 1. An r of -1 indicates a perfect negative linear relationship between variables, an r of 0 indicates no linear relationship between variables, and an r of 1 indicates a perfect positive linear relationship between variables.

It is very important to remember that **correlation says nothing about how much Y changes when X changes**.

Figure 10.13 shows randomly generated sets of points to illustrate different correlations.

 Figure 10.12: Examples of correlations of different magnitude and direction, with associated correlation coefficient r.

In both rows, from left to right correlations go from weak to strong. In the top row the correlations are positive (larger values for one quantity are associated with larger values for the other). In the bottom row they are negative (larger values for one quantity are associated with smaller values for the other). So, what does that mean, in terms of judging the strength of each type of relations. Andrew Heiss has a nice guide, that I find useful (Heiss, 2021):

| **r** | **Rough meaning** |
| --- | --- |
| ± 0.1–0.3 | Modest |
| ± 0.3–0.5 | Moderate |
| ± 0.5–0.8 | Strong |
| ± 0.8–0.9 | Very strong |

Let us try this with the blue jay data. We will find the correlation between our two variables:

cor(blue\_jays$body\_mass\_g, blue\_jays$head\_length\_mm)

## [1] 0.6294447

cor.test(blue\_jays$body\_mass\_g, blue\_jays$head\_length\_mm)

##   
## Pearson's product-moment correlation  
##   
## data: blue\_jays$body\_mass\_g and blue\_jays$head\_length\_mm  
## t = 8.9105, df = 121, p-value = 6.302e-15  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5091462 0.7256207  
## sample estimates:  
## cor   
## 0.6294447

As you can see from the analysis, that the correlation is: 0.6294447. Using the guide from Heiss, this is a strong positive correlation.

A critical property of Pearson’s r is that it is unaffected by linear transformations. This means that multiplying a variable by a constant and/or adding a constant does not change the correlation of that variable with other variables. For instance, the correlation of Weight and Height does not depend on whether Height is measured in inches, feet, or even miles. Similarly, adding five points to every student’s test score would not change the correlation of the test score with other variables such as GPA. It is also important to remember that correlation is not causation and another unnoticed variable may be influencing results.

# Correlation Matrix (Correlogram)

When there are more than three to four quantitative variables, all-against-all scatterplot matrices quickly become unwieldy. In this case, it is more useful to quantify the amount of association between pairs of variables and visualize these quantities rather than the raw data. One common way to do this is to calculate correlation coefficients.

Wilke, calls the visualization of correlation coefficients, correlograms. A correlogram or correlation matrix allows to analyse the relationship between each pair of numeric variables in a dataset. It gives a quick overview of the whole dataset. It is more used for exploratory purpose than explanatory.

The R Graph Gallery has a nice overview of this topic and explores options using the GGally package offers great options to build [correlograms](https://www.data-to-viz.com/graph/correlogram.html). [GGally](https://ggobi.github.io/ggally/) extends ggplot2 by adding several functions to reduce the complexity of combining geoms with transformed data. Some of these functions include a pairwise plot matrix, a scatterplot plot matrix, a parallel coordinates plot, a survival plot, and several functions to plot networks.

The [ggpairs()](https://ggobi.github.io/ggally/reference/ggpairs.html#examples) function build a [classic correlogram](https://www.r-graph-gallery.com/199-correlation-matrix-with-ggally.html) with scatterplot, correlation coefficient and variable distribution. On top of that, it is possible to inject [ggplot2](https://www.r-graph-gallery.com/ggplot2-package.html) code, for instance to [color categories](https://www.r-graph-gallery.com/199-correlation-matrix-with-ggally.html#category). Let us look at a simply example using the blue jay dataset . The first thing we need to do is turn on the package.

The first I need to do is subset the data so that I do not have too many columns in my plot. Anything more than 5 or 6, you plot might become hard to read. I am going to use select to only keep the head\_length\_mm, body\_mass\_g, skull\_size\_mm variables.

blue\_jays\_matrix <- blue\_jays %>%   
 select(head\_length\_mm, body\_mass\_g, skull\_size\_mm, sex)

OK, now let us draw our graph, and then we will talk about what this means [Figure 10.13].

ggpairs(blue\_jays\_matrix) # generate the correlation matrix

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

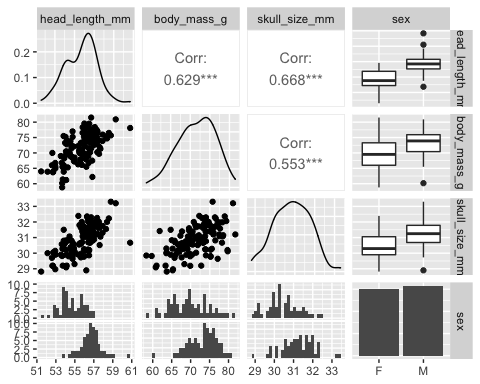


Figure 10.13: Correlation matrix of the filtered blue jay data.

What is happening is that ggpairs is plotting each variable against the other. ggpairs uses scatter plots for quantitative, quantitative pairs in the data. So, on the bottom part of the matrix we have our scatterplots, as we are only examing quanitative, quanitative pairs.

On the diagonal, we have the density functions for each of the variables. As you can see in Figure 10.13, on the top half, ggpairs also provides us with the correlation for each of the quanitative, quanitative pairs. Using the guide from Hiess, we can see that all of the variables are correlated, with signifigance. We are also presented with boxplots and histograms for the qualitative variable, sex.

It is also possible to inject [ggplot2](https://www.r-graph-gallery.com/ggplot2-package.html) code into a ggcor statement. For example, you can add [color categories](https://www.r-graph-gallery.com/199-correlation-matrix-with-ggally.html#category) [Figure 10.14].

ggpairs(blue\_jays\_matrix, ggplot2::aes(colour=sex))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

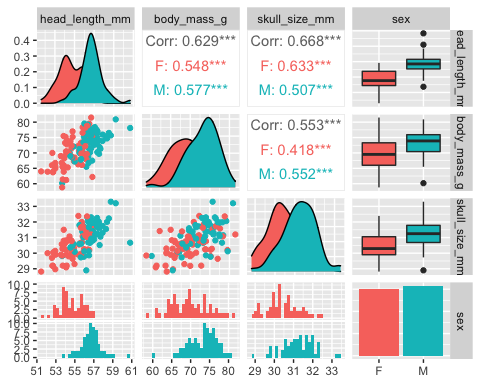


Figure 10.14: Correlation matrix of the filtered blue jay data, colored by sex.

The Oxford Protein Informatics Group has a nice overview (with examples) of using [ggpairs](https://www.blopig.com/blog/2019/06/a-brief-introduction-to-ggpairs/).

Another option is to use the ggcorr() function, allows to visualize the correlation of each pair of variable as tiles. You can use the method argument allows to pick the correlation type you desire [Figure 10.15].

blue\_jays\_matrix %>%  
 ggcorr(method = c("everything", "pearson"))

## Warning in ggcorr(., method = c("everything", "pearson")): data in column(s)  
## 'sex' are not numeric and were ignored

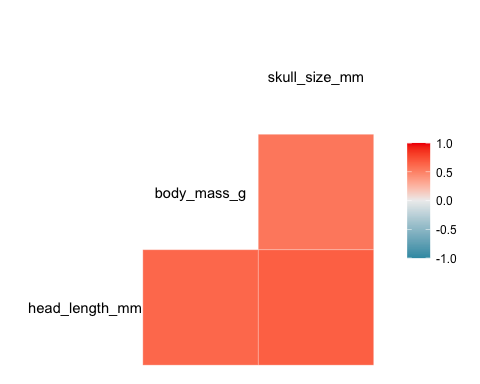


Figure 10.15: blue jay data correlation, as tiles.

Our data is only showing positive correlations. Figure 10.16 is demonstrating what this would look like if we had both positive and negative correlations between the variables.

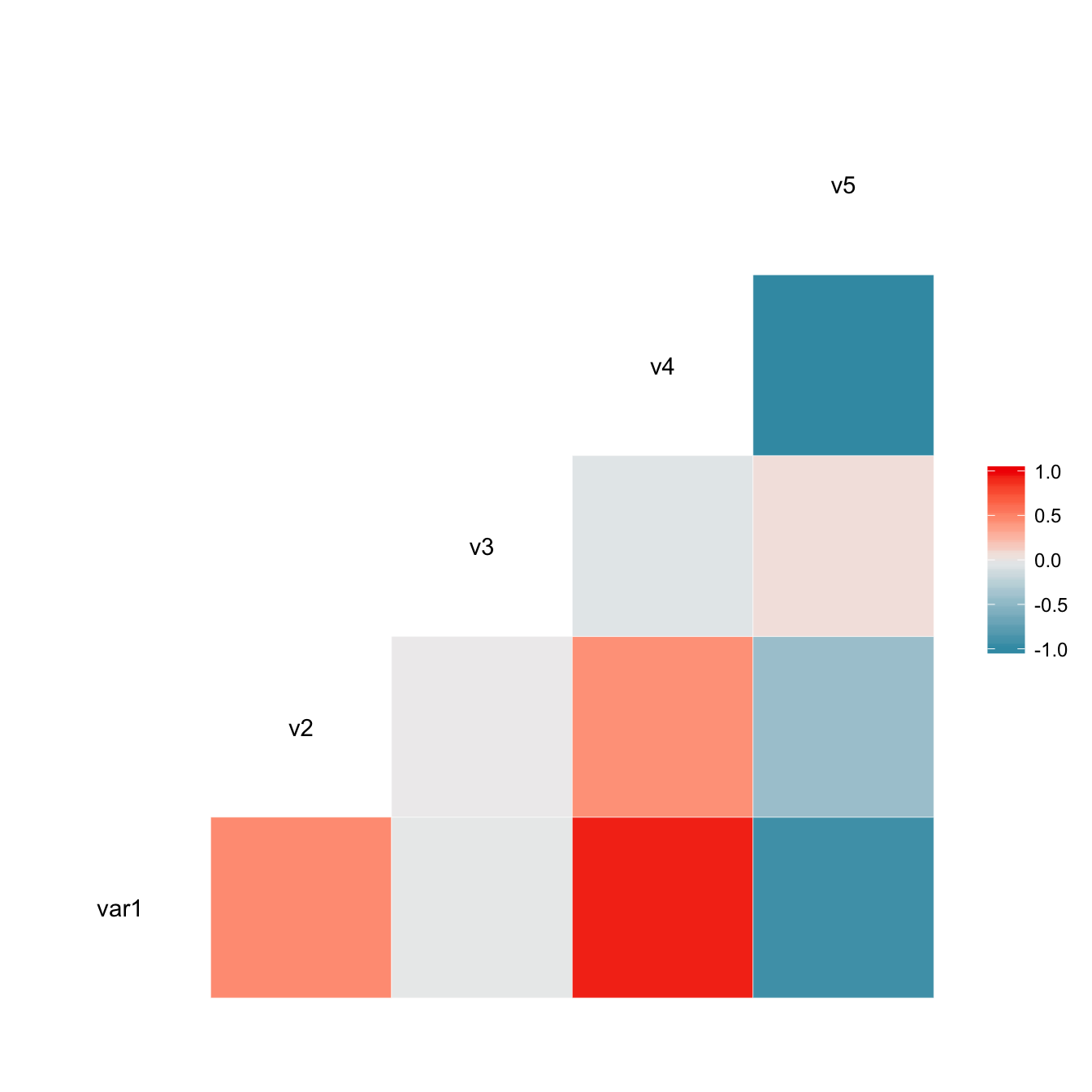


Figure 10.16: Data that includes both positive and negative correlations.

# Paired data

Paired data is where there are two or more measurements of the same quantity under slightly different conditions. Examples include two comparable measurements on each subject (e.g., the length of the right and the left arm of a person), repeat measurements on the same subject at different time points (e.g., a person’s weight at two different times during the year).

For paired data, it is reasonable to assume that the two measurements belonging to a pair are more similar to each other than to the measurements belonging to other pairs. Therefore, for paired data, we need to choose visualizations that highlight any differences between the paired measurements.

One option is a simply scatter plot, on top of a diagonal line marking x = y. In such a plot, if the only difference between the two measurements of each pair is random noise, then all points in the sample will be scattered symmetrically around this line. Any systematic differences between the paired measurements, by contrast, will be visible in a systematic shift of the data points up or down relative to the diagonal.

# Handling Overlapping Points

When we want to visualize some datasets, we often experience the challenge that simple x-y scatterplots do not work very well because many points lie on top of each other and partially or fully overlap. This is called overplotting, which means that we are plotting many points on top of each other. Strategies include reducing the data before mapping to a visual representation, simplifying the visual representation itself, or modifying the space of the plot (Sarikaya & Gleicher, 2018). We are only going to cover partial transparency and jittering.

## Partial Transparency and Jittering

For this we will be looking at fuel economy data are the result of vehicle testing done at the Environmental Protection Agency’s National Vehicle and Fuel Emissions Laboratory in Ann Arbor, Michigan, and by vehicle manufacturers with oversight by EPA. The data that I have downloaded is [dataset](https://fueleconomy.gov/feg/download.shtml) for all model years (1984–2022).

cars <- read\_csv("../raw\_data/vehicles.csv")

## Rows: 44432 Columns: 83  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (22): drive, eng\_dscr, fuelType, fuelType1, make, model, mpgData, trany,...  
## dbl (59): barrels08, barrelsA08, charge120, charge240, city08, city08U, city...  
## lgl (2): phevBlended, tCharger  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

Let us visualize the data [Figure 10.17].

cars %>%   
 filter(!is.na(drive)) %>%   
 filter(between(year, 2017, 2021)) %>%   
 ggplot(mapping = aes(x=displ, y = city08, color = drive))+  
 geom\_point(size = 3) +  
 labs(x = "displacement (l)", y = "fuel economy (mpg)") +  
 scale\_color\_brewer(palette = "Dark2")

## Warning: Removed 178 rows containing missing values (geom\_point).

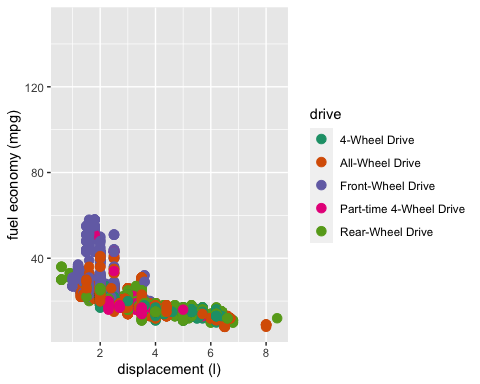


Figure 10.17: City fuel economy versus engine displacement, colored by the type of drive. Each point represents one car (2017-2021). Data source: US Environmental Protection Agency (EPA).

It is very easy to see why this is not an effective visualization. Specifically, the points are plotted on top of others and obscure them. Figure 10.18 presents the data with jitter turned on. I also added transparency (alpha) to avoid overplotting. For large datasets with overplotting the [alpha](https://tidyverse.github.io/ggplot2-docs/reference/geom_point.html) aesthetic will make the points more transparent. Alpha values range from 0 to 1.

cars %>%   
 filter(!is.na(drive)) %>%   
 filter(between(year, 2017, 2021)) %>%   
 ggplot(mapping = aes(x=displ, y = city08, color = drive))+  
 geom\_point(size = 3, alpha = 1/5) +  
 geom\_jitter() +  
 labs(x = "displacement (l)", y = "fuel economy (mpg)") +  
 scale\_color\_brewer(palette = "Dark2")

## Warning: Removed 178 rows containing missing values (geom\_point).  
## Removed 178 rows containing missing values (geom\_point).

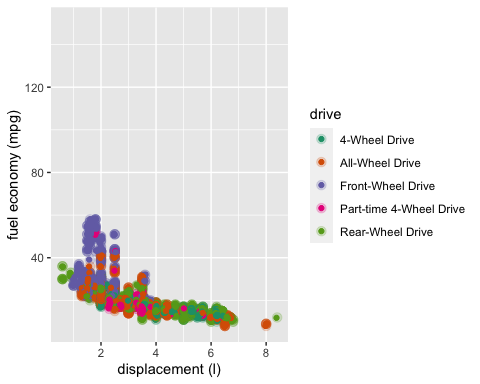


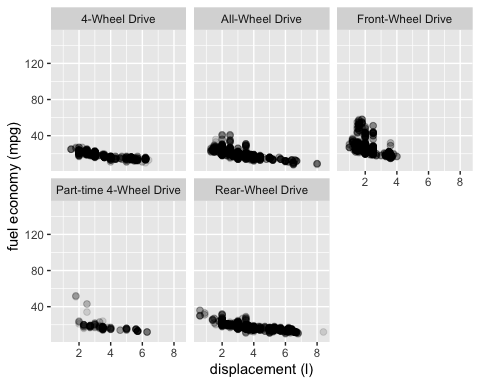
Figure 10.18: City fuel economy versus engine displacement, colored by the type of drive, using jitter, and adding the alpha aesthetic. Each point represents one car (2017-2021). Data source: US Environmental Protection Agency (EPA).

However, making points partially transparent is not always sufficient to solve the issue of overplotting. For example, even though we can see in Figure 10.18 that some points have a darker shade than others, it is difficult to estimate how many points were plotted on top of each other in each location. In addition, while the differences in shading are clearly visible, they are not self-explanatory. A reader who sees this figure for the first time will likely wonder why some points are darker than others and will not realize that those points are in fact multiple points stacked on top of each other.

What might be another solution to our problem?

cars %>%   
 filter(!is.na(drive)) %>%   
 filter(between(year, 2017, 2021)) %>%   
 ggplot(mapping = aes(x=displ, y = city08))+  
 geom\_point(alpha = 1/8, size = 2) +  
 facet\_wrap(vars(drive)) +  
 labs(x = "displacement (l)", y = "fuel economy (mpg)")

## Warning: Removed 178 rows containing missing values (geom\_point).



scale\_color\_brewer(palette = "Dark2")

## <ggproto object: Class ScaleDiscrete, Scale, gg>  
## aesthetics: colour  
## axis\_order: function  
## break\_info: function  
## break\_positions: function  
## breaks: waiver  
## call: call  
## clone: function  
## dimension: function  
## drop: TRUE  
## expand: waiver  
## get\_breaks: function  
## get\_breaks\_minor: function  
## get\_labels: function  
## get\_limits: function  
## guide: legend  
## is\_discrete: function  
## is\_empty: function  
## labels: waiver  
## limits: NULL  
## make\_sec\_title: function  
## make\_title: function  
## map: function  
## map\_df: function  
## n.breaks.cache: NULL  
## na.translate: TRUE  
## na.value: NA  
## name: waiver  
## palette: function  
## palette.cache: NULL  
## position: left  
## range: <ggproto object: Class RangeDiscrete, Range, gg>  
## range: NULL  
## reset: function  
## train: function  
## super: <ggproto object: Class RangeDiscrete, Range, gg>  
## rescale: function  
## reset: function  
## scale\_name: brewer  
## train: function  
## train\_df: function  
## transform: function  
## transform\_df: function  
## super: <ggproto object: Class ScaleDiscrete, Scale, gg>

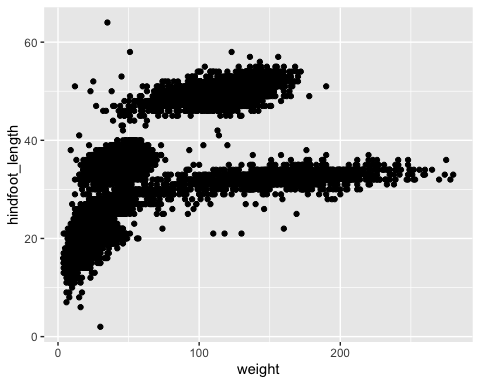
Figure 10.19: Car data represented as a small multiple (facted).

Let us look at a couple of examples, using the species data (surveys\_complete).

surveys <- read\_csv("../raw\_data/surveys\_complete.csv")

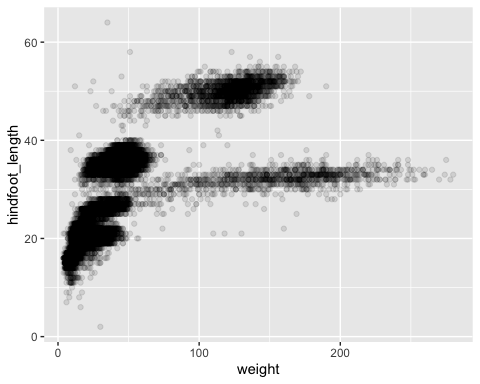
## Rows: 30463 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): species\_id, sex, genus, species, taxa, plot\_type  
## dbl (7): record\_id, month, day, year, plot\_id, hindfoot\_length, weight  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

ggplot(data = surveys, aes(x = weight, y = hindfoot\_length)) +  
 geom\_point()



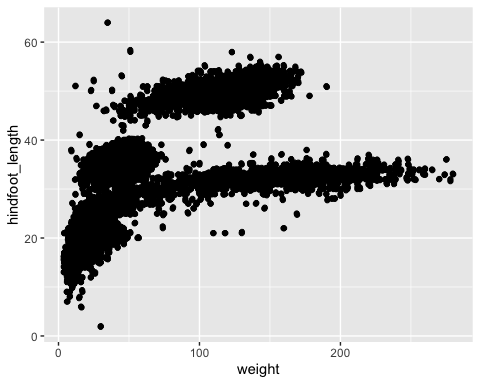
Another option is adding transparency (alpha) to avoid overplotting. For large datasets with overplotting the alpha aesthetic will make the points more transparent. Alpha values range from 0 to 1.

surveys %>%   
ggplot(mapping = aes(x = weight, y = hindfoot\_length)) +  
 geom\_point(alpha = 0.1)



Now let us try this with jitter:

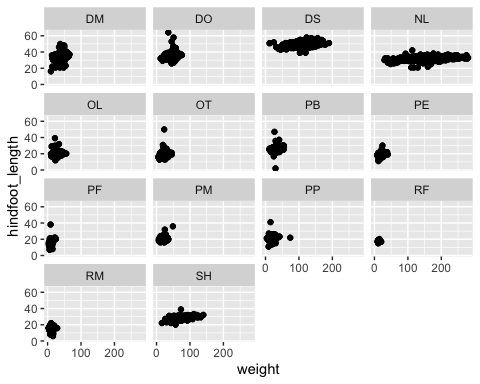
surveys %>%   
ggplot(mapping = aes(x = weight, y = hindfoot\_length)) +  
 geom\_point() +  
 geom\_jitter()



What do you think? Not the best solution for this data. One downside of jittering is that it does change the data and therefore has to be performed with care. If we jitter too much, we end up placing points in locations that are not representative of the underlying dataset. The result is a misleading visualization of the data.

Let us try facet\_wrap to see if this helps. In the example below, we are faceting by species ID.

surveys %>%  
 ggplot(mapping = aes(x = weight, y = hindfoot\_length)) +  
 geom\_point() +  
 facet\_wrap(vars(species\_id)) +  
 geom\_jitter()



# References

Cleveland, W. S., & McGill, R. (1984). The many faces of a scatterplot [Journal Article]. *Journal of the American Statistical Association*, *79*(388), 807–822. <https://doi.org/10.2307/2288711>

Heiss, A. (2021). (Web Page November 19; Vol. 2021). <https://datavizs21.classes.andrewheiss.com/content/07-content/#slides>

Lane, D. (2007). *Introduction to linear regression* (Electronic Book Section No. 14; pp. 207–229).

Lane, D. M. (2007). *Describing bivariate data* (Electronic Book Section No. 4; pp. 164–184).

Munzner, T. (2014). Why: Task abstraction [Book Section]. In *Visualization analysis and design* (pp. 42–64). CRC Press, Taylor & Francis Group.

Osherson, D., & Lane, D. M. (2007). *Introduction - levels of measurement* (pp. 34–39) [Electronic Book Section].

Ribecca, S. (2019). *The data visualisation catalogue* [Electronic Book].

Sarikaya, A., & Gleicher, M. (2018). Scatterplots: Tasks, data, and designs [Journal Article]. *IEEE Trans Vis Comput Graph*, *24*(1), 402–412. <https://doi.org/10.1109/TVCG.2017.2744184>

Wickham, H., & Sievert, C. (2016). Getting started with ggplot2 [Book Section]. In *Ggplot2 : Elegant graphics for data analysis*. Springer International Publishing AG. <http://ebookcentral.proquest.com/lib/cua/detail.action?docID=4546676>