Statistics with Sparrows - 10

Julia Schroeder

July 2020

HO 10

Variance, Covariance and all that

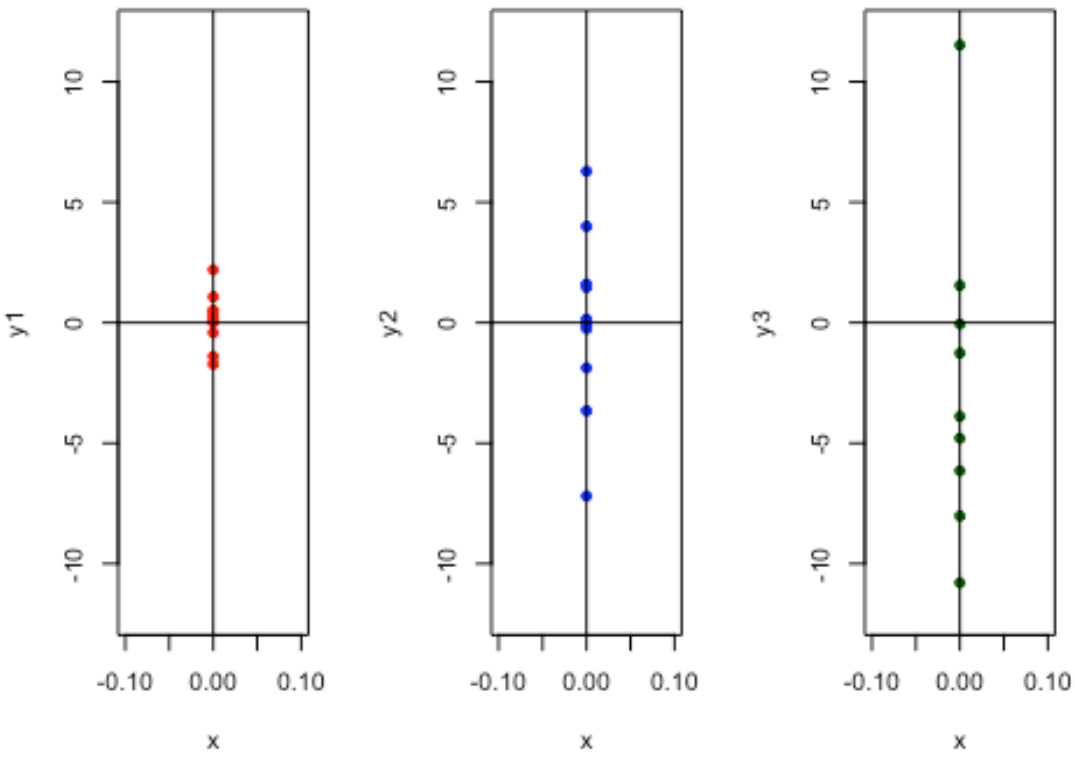
Learning aims

- Understand variance and covariance
- Know variance calculus rules
- Understand what a R^2 value is
- Remember why we often use sums of squared deviations

First, remember that the variance is really just the sum of the deviations from the mean. That means, for a given dataset, we take the mean, and then subtract that from each datapoint. That *deviation* then is squared, and we add all up. We then divide by the sample size (minus one). We can visualize that neatly:

```
rm(list = ls())
# create three data sets y with different variances (1, 10, 100)
# rnorm() requires sample size (20), mean and sd
y1<-rnorm(10, mean=0, sd=sqrt(1))
var(y1)

## [1] 1.277436

y2<-rnorm(10, mean=0, sd=sqrt(10))
var(y2)

## [1] 14.31409

y3<-rnorm(10, mean=0, sd=sqrt(100))
var(y3)

## [1] 68.8384

# create x variable for plotting
x<-rep(0,10)
# making a 1x3 plot using mfrow() (look it up if you don't know what that doe
s)
par(mfrow = c(1, 3))
plot(x, y1, xlim=c(-0.1,0.1), ylim=c(-12,12), pch=19, cex=0.8, col="red")
```

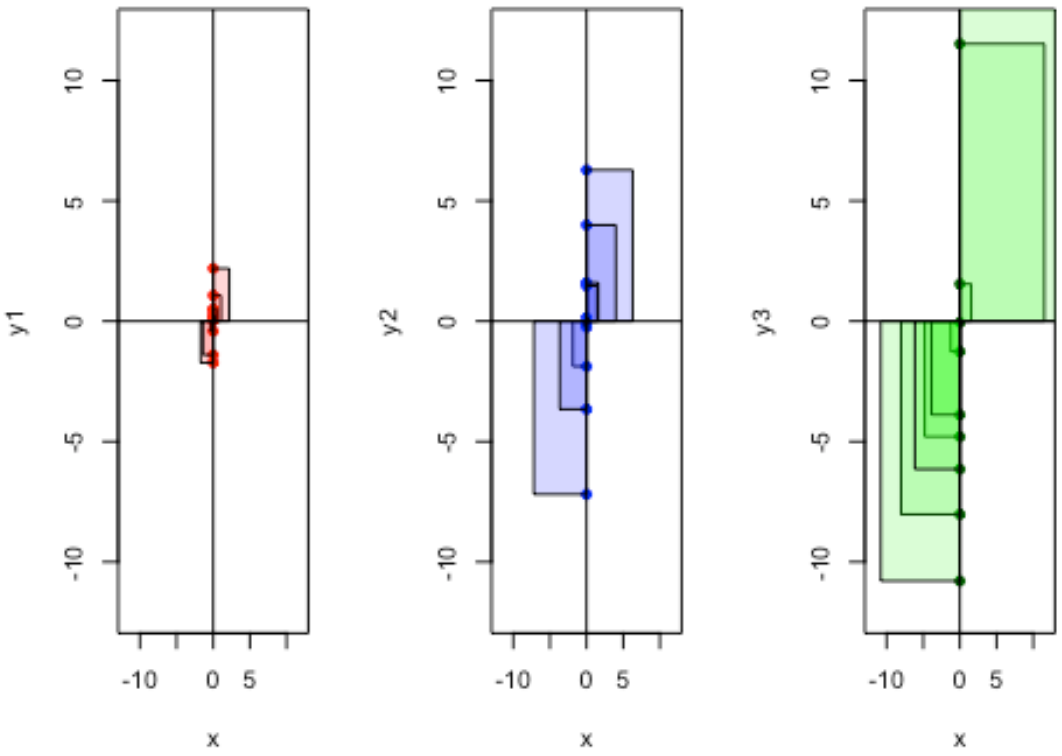
```
abline(v=0)
abline(h=0)

plot(x, y2, xlim=c(-0.1,0.1), ylim=c(-12,12), pch=19, cex=0.8, col="blue")
abline(v=0)
abline(h=0)

plot(x, y3, xlim=c(-0.1,0.1), ylim=c(-12,12), pch=19, cex=0.8,, col="darkgree
n")
abline(v=0)
abline(h=0)
```



From the plot alone it is clear that the y3 variable has a larger variance than the y1 variable, and yet all have the same mean - 0. If we now take the squares:

```
# lets plot this again, this time with the squares.
?polygon()
par(mfrow = c(1, 3))
plot(x, y1, xlim=c(-12,12), ylim=c(-12,12) ,pch=19, cex=0.8, col="red")
abline(v=0)
abline(h=0)
polygon(x=c(0,0,y1[1],y1[1]),y=c(0,y1[1],y1[1],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[2],y1[2]),y=c(0,y1[2],y1[2],0), col=rgb(1, 0, 0,0.2))
```

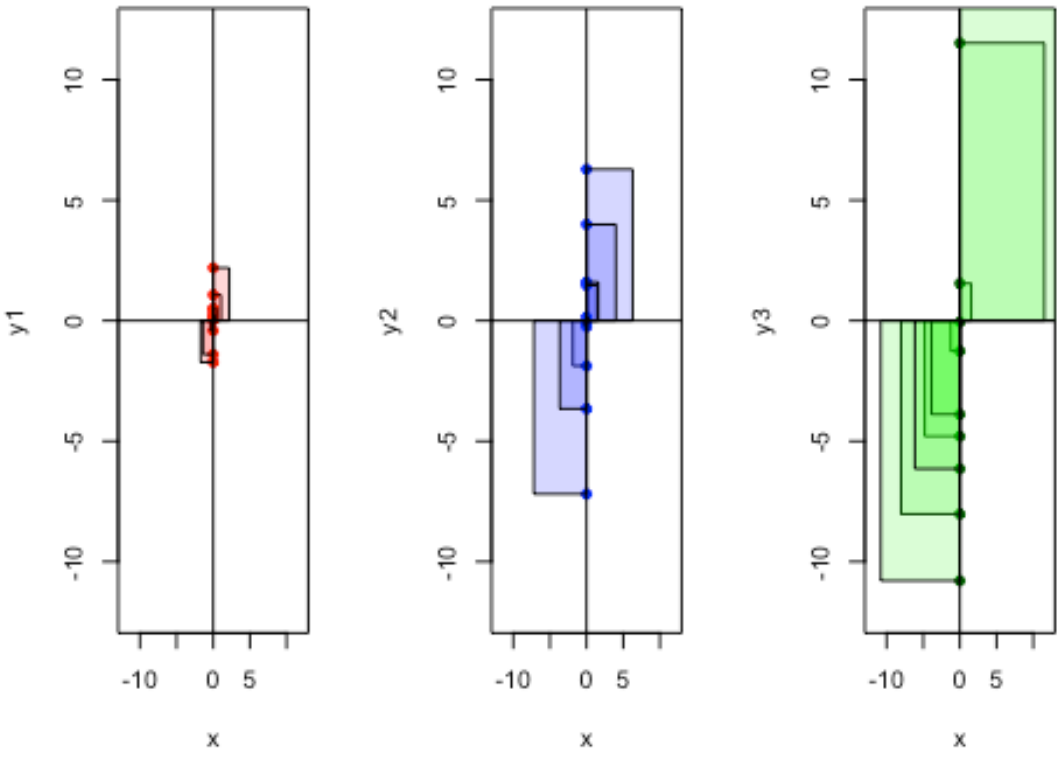
```
polygon(x=c(0,0,y1[3],y1[3]),y=c(0,y1[3],y1[3],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[4],y1[4]),y=c(0,y1[4],y1[4],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[4],y1[4]),y=c(0,y1[4],y1[4],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[5],y1[5]),y=c(0,y1[5],y1[5],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[6],y1[6]),y=c(0,y1[6],y1[6],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[7],y1[7]),y=c(0,y1[7],y1[7],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[8],y1[8]),y=c(0,y1[8],y1[8],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[9],y1[9]),y=c(0,y1[9],y1[9],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[10],y1[10]),y=c(0,y1[10],y1[10],0), col=rgb(1, 0, 0,0.2))

# that was hard work and I'm lazy, so I'll write a for loop for the next two:
plot(x, y2, xlim=c(-12,12), ylim=c(-12,12), pch=19, cex=0.8, col="blue")
abline(v=0)
abline(h=0)
for (i in 1:length(y2)) {
polygon(x=c(0,0,y2[i],y2[i]),y=c(0,y2[i],y2[i],0), col=rgb(0, 0, 1,0.2))
}

plot(x, y3, xlim=c(-12,12), ylim=c(-12,12), pch=19, cex=0.8,, col="darkgreen"
)
abline(v=0)
abline(h=0)
for (i in 1:length(y3)) {
polygon(x=c(0,0,y3[i],y3[i]),y=c(0,y3[i],y3[i],0), col=rgb(0, 1, 0,0.2))
}
```



It is evident that the green squares are much larger than the red ones, and also, that the sum of them will be much larger.

```
# lets plot this again, this time with the squares.
?polygon()
par(mfrow = c(1, 3))
plot(x, y1, xlim=c(-12,12), ylim=c(-12,12) ,pch=19, cex=0.8, col="red")
abline(v=0)
abline(h=0)
polygon(x=c(0,0,y1[1],y1[1]),y=c(0,y1[1],y1[1],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[2],y1[2]),y=c(0,y1[2],y1[2],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[3],y1[3]),y=c(0,y1[3],y1[3],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[4],y1[4]),y=c(0,y1[4],y1[4],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[4],y1[4]),y=c(0,y1[4],y1[4],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[5],y1[5]),y=c(0,y1[5],y1[5],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[6],y1[6]),y=c(0,y1[6],y1[6],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[7],y1[7]),y=c(0,y1[7],y1[7],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[8],y1[8]),y=c(0,y1[8],y1[8],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[9],y1[9]),y=c(0,y1[9],y1[9],0), col=rgb(1, 0, 0,0.2))
polygon(x=c(0,0,y1[10],y1[10]),y=c(0,y1[10],y1[10],0), col=rgb(1, 0, 0,0.2))

# that was hard work and I'm lazy, so I'll write a for loop for the next two:
plot(x, y2, xlim=c(-12,12), ylim=c(-12,12), pch=19, cex=0.8, col="blue")
```

```
abline(v=0)
abline(h=0)
for (i in 1:length(y2)) {
polygon(x=c(0,0,y2[i],y2[i]),y=c(0,y2[i],y2[i],0), col=rgb(0, 0, 1,0.2))
}

plot(x, y3, xlim=c(-12,12), ylim=c(-12,12), pch=19, cex=0.8,, col="darkgreen"
)
abline(v=0)
abline(h=0)
for (i in 1:length(y3)) {
polygon(x=c(0,0,y3[i],y3[i]),y=c(0,y3[i],y3[i],0), col=rgb(0, 1, 0,0.2))
}
```



Now, what happens if we have a second variable, x, for instance? We can now calculate the *covariances* between y and x as the product between the deviations of the mean. It's similar to the variance - the square of the deviations is really the product between the deviations (just that it's the same each time):

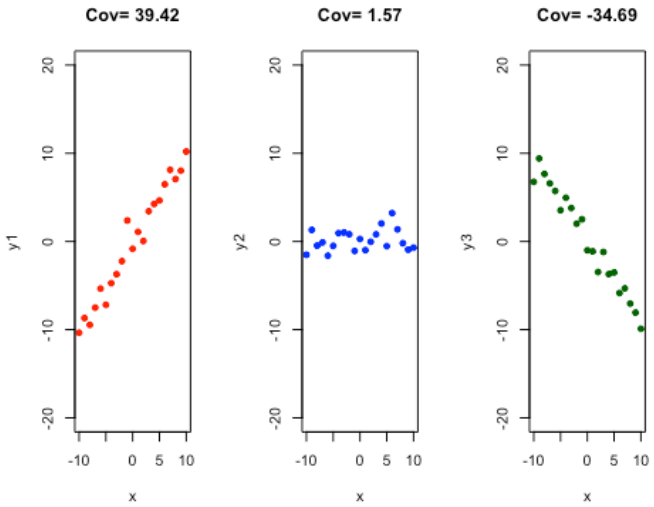
$$var(y1) = \sigma_{y1}^2 = \frac{\sum(y1_i - \overline{y})^2}{n-1} = \frac{\sum((y1_i - \overline{y}) * (y1_i - \overline{y}))}{n-1}$$

So, naturally, if we want to calculate the covariance between two variables, we swap one of these terms for the respective term of the second variable:

$$covar(x, y1) = \frac{\sum((x_i - \overline{x}) * (y1_i - \overline{y}))}{n - 1}$$

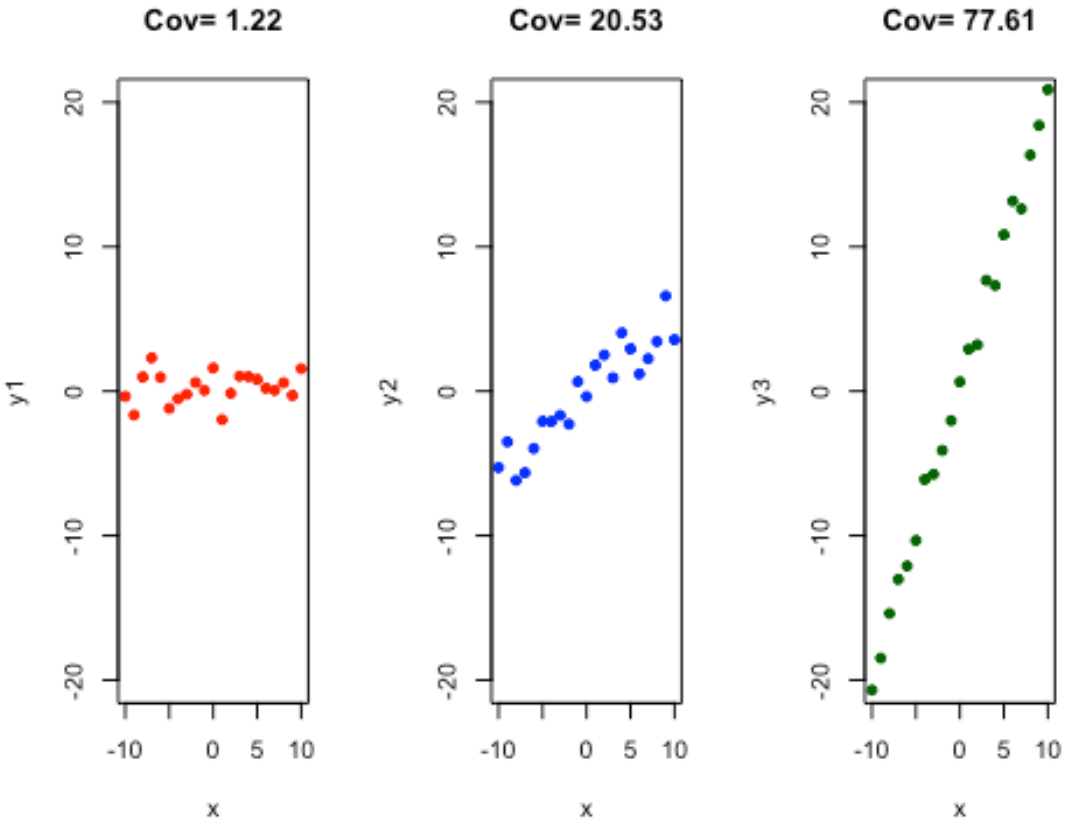
Let's simulate that y and x are related. We do this by simply multiplying x with y, but we change the intensity of the association. We'll have to clear our workspace because we're making new variables. Do this - otherwise you'll get nonsensical results.

```
rm(list = ls())
par(mfrow = c(1, 3))
x<-c(-10:10)
var(x)

## [1] 38.5

y1<-x*1 + rnorm(21, mean=0, sd=sqrt(1))
# here the association is 1:1
cov(x, y1)

## [1] 39.42318

plot(x, y1, xlim=c(-10,10), ylim=c(-20, 20), col="red", pch=19, cex=0.8, main
=paste("Cov=",round(cov(x,y1),digits=2)))


y2<-rnorm(21, mean=0, sd=sqrt(1))
# Here, there is no association
cov(x, y2)

## [1] 1.572258

plot(x, y2, xlim=c(-10,10), ylim=c(-20, 20), col="blue", pch=19, cex=0.8, mai
n=paste("Cov=",round(cov(x,y2),digits=2)))
y3<- x* (-1) +rnorm(21, mean=0, sd=sqrt(1))
# Here, the association is negative
cov(x, y3)

## [1] -34.6899

plot(x, y3, xlim=c(-10,10), ylim=c(-20, 20), col="darkgreen", pch=19, cex=0.8
, main=paste("Cov=",round(cov(x,y3),digits=2)))
```



You can see how the covariance changes from positive to negative. Now, what happens if we introduce stronger or weaker associations?

```
rm(list = ls())
par(mfrow = c(1, 3))
x<-c(-10:10)
var(x)

## [1] 38.5

y1<-x*0.1 + rnorm(21, mean=0, sd=sqrt(1))
# here the association is very weak, but not 0:
cov(x, y1)

## [1] 1.221025

plot(x, y1, xlim=c(-10,10), ylim=c(-20, 20), col="red", pch=19, cex=0.8, main
=paste("Cov=",round(cov(x,y1),digits=2)))
y2<-x*0.5+ rnorm(21, mean=0, sd=sqrt(1))
# Here, it is 0.5
cov(x, y2)

## [1] 20.53452

plot(x, y2, xlim=c(-10,10), ylim=c(-20, 20), col="blue", pch=19, cex=0.8, mai
n=paste("Cov=",round(cov(x,y2),digits=2)))
y3<- x*2 +rnorm(21, mean=0, sd=sqrt(1))
# Here, the association is 2
cov(x, y3)

## [1] 77.61198

plot(x, y3, xlim=c(-10,10), ylim=c(-20, 20), col="darkgreen", pch=19, cex=0.8
, main=paste("Cov=",round(cov(x,y3),digits=2)))
```



The covariance changes the stronger the variables are associated with each other.

However, the covariance is not really very useful, as it is linked to the units. We can fix this by standardizing the data such that the response variable has a standard deviation of 1. Another option that is a bit more elegant, is standardizing the covariance, using the standard deviations of both variables:

$$cor(x, y1) = \frac{1}{n-1} \frac{\sum((x_i - \overline{x}) * (y1_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}}$$

This correlation coefficient helps gauge the strength of an association between two variables that covary with each other. It can take on values between -1 and 1, where 0 indicates no association, and -1 and 1 the strongest association possible. This statistic is without units, so it can be compared between different datasets.

Naturally, it is easy to calculate in r:

```
cov(x,y1)

## [1] 1.221025

cor(x, y1)

## [1] 0.1838611
```

```
cov(x,y2)
```

```
## [1] 20.53452
```

```
cor(x,y2)
```

```
## [1] 0.9312524
```

```
cov(x,y3)
```

```
## [1] 77.61198
```

```
cor(x,y3)
```

```
## [1] 0.9977698
```

What happens if we introduce variation in y?

```
rm(list = ls())
par(mfrow = c(3, 1))
x<-c(-10:10)
var(x)
```

```
## [1] 38.5
```

```
y1<-x*1 + rnorm(21, mean=0, sd=sqrt(1))
# here the association is 1:1, with low variance in y
cov(x, y1)
```

```
## [1] 37.31275
```

```
plot(x, y1, xlim=c(-10,10), ylim=c(-20, 20), col="red", pch=19, cex=0.8, main
=paste("Cov=",round(cov(x,y1),digits=2)," Cor=",round(cor(x,y1),digits=2)))
y2<-x*1 + rnorm(21, mean=0, sd=sqrt(10))
# The association remains 1:1, but higher variance in y
cov(x, y2)
```

```
## [1] 39.81266
```

```
plot(x, y2, xlim=c(-10,10), ylim=c(-20, 20), col="blue", pch=19, cex=0.8, mai
n=paste("Cov=",round(cov(x,y2),digits=2)," Cor=",round(cor(x,y2),digits=2)))
y3<- x*1 + rnorm(21, mean=0, sd=sqrt(100))

cov(x, y3)
```

```
## [1] 34.8734
```

```
plot(x, y3, xlim=c(-10,10), ylim=c(-20, 20), col="darkgreen", pch=19, cex=0.8
, main=paste("Cov=",round(cov(x,y3),digits=2)," Cor=",round(cor(x,y3),digits=
2)))
```

The covariance goes up (not strongly) but the correlation goes down! This is because the association (the higher x, the higher y, 1:1) remains the same, meaning how x and y covary remains, but the correlation goes up - meaning there is less "noise". Notably, if the correlation is negative the association is negative, and vice versa. The same is also true for the covariance. The correlation coefficient is an effect size that is easily comparable between studies, because it is unit free. The closer to zero, the less likely there is a correlation, the closer to 1 or -1, the stronger the association. Note, the value however does not indicate how steep the association is.

Calculus rules for mean, variance and covariance

Rules for the mean

1) the mean of a constant is the constant

```
rm(list = ls())
mean(4)

## [1] 4
```

2) Adding a constant value to each term increases the mean, or expected value, by the constant.

```
y<-c(-3,5,8,-2)
mean(y)

## [1] 2

mean(y+4)

## [1] 6

mean(y)+4

## [1] 6
```

3) Multiplying each term by a constant value multiplies the mean by that constant.

```
mean(y*4)

## [1] 8

mean(y)*4

## [1] 8
```

4) The mean of the sum of two variables is the sum of the means.

```
y1<-runif(n=4)
mean(y1)

## [1] 0.4128074

mean(y)

## [1] 2
```

```
mean(y1) +mean (y)
```

```
## [1] 2.412807
```

```
mean(y1+y)
```

```
## [1] 2.412807
```

Rules for the Variance

1) The variance of a constant is 0

```
a<-c(4,4,4,4)
var(a)
```

```
## [1] 0
```

2) Adding a constant value to the variable does not change the variance, it only shifts the mean

```
var(y)
```

```
## [1] 28.66667
```

```
mean(y)
```

```
## [1] 2
```

```
var(y+4)
```

```
## [1] 28.66667
```

```
mean(y+4)
```

```
## [1] 6
```

3) Multiplying a random variable by a constant increases the variance by the square of the constant.

```
var(y)
```

```
## [1] 28.66667
```

```
var(y*2)
```

```
## [1] 114.6667
```

```
var(y*4)
```

```
## [1] 458.6667
```

4) The variance of the sum of two or more random variables is equal to the sum of each of their variances only when the random variables are independent. Independent means their covariance is zero.

```
var(y)
```

```
## [1] 28.66667
```

```
y2<-c(-2, -10, 20, 18)
var(y2)

## [1] 219.6667

var(y+y2)

## [1] 267

var(y)+var(y2)

## [1] 248.3333
```

Rules for covariances

1) The covariance of two constants, c and k, is 0.

```
rm(list = ls())
a<-rep(4,10)
b<-rep(6,10)
cov(a,b)

## [1] 0
```

2) The covariance of two independent random variables is 0.

```
rm(list = ls())
x<-runif(10)
y<-runif(10)
cov(x,y)

## [1] 0.001228788
```

3) The covariance is a combinative (not affected by order).

```
cov(x,y)

## [1] 0.001228788

cov(y,x)

## [1] 0.001228788
```

4) The covariance of a random variable with a constant is zero.

```
a<-rep(4,10)
cov(x,a)

## [1] 0
```

5) Adding a constant to either or both random variables does not change their covariances.

```
cov(x,y)

## [1] 0.001228788

cov((x+5),y)
```

```
## [1] 0.001228788

cov((x+5),(y+5))

## [1] 0.001228788
```

6) Multiplying a random variable by a constant multiplies the covariance by that constant.

```
cov(x,y)

## [1] 0.001228788

cov((x*2),y)

## [1] 0.002457576
```

7) The covariance of a random variable with a sum of random variables is just the sum of the covariances with each of the random variables.

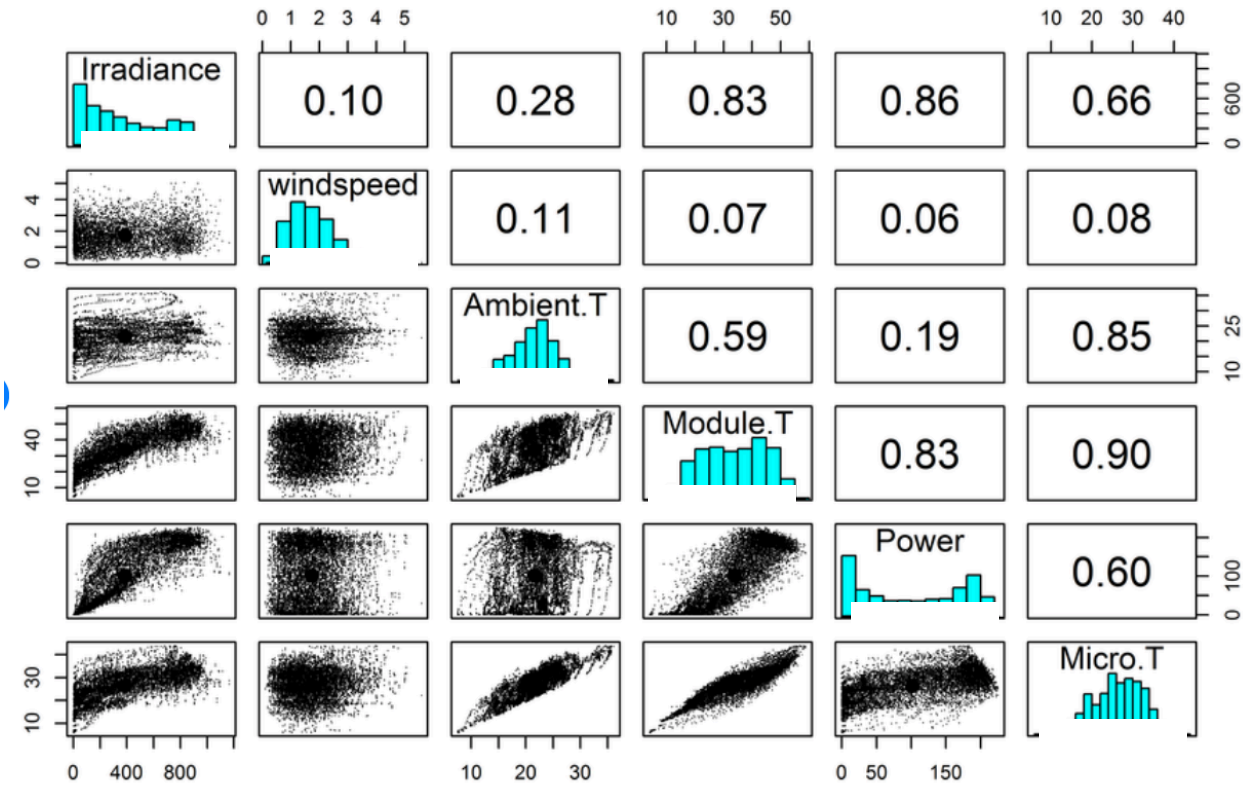
```
z<-x*0.4+0.1*runif(10)
cov((x+y),z)

## [1] 0.03555352

cov(x,z)+cov(y,z)

## [1] 0.03555352
```

Exercises:

1) Go to the website http://guessthecorrelation.com and play until you have a good "feel" for the correlation.

2) Check out this visualization of variances and covariances. Discuss in the group how this visualization helps, and also, what it does not show.
https://i.imgur.com/cWwxYa9.gifv

3) Find out how to produce a matrix of x,y scatterplots of multiple variables, histograms, and the correlation coefficient, like the one below. You can use the sparrow morphology data for it, or find your own data, or simulate your own data. Make one of these scatterplot matrices per group. Submit one pdf file with the graph, per group.


Irradiance
0.10
0.28
0.83
0.86
0.66
windspeed
0.11
0.07
0.06
0.08
Ambient.T
0.59
0.19
0.85
Module.T
0.83
0.90
Power
0.60
Micro.T
0 1 2 3 4 5
10 30 50
10 20 30 40
0 2 4
10 40
10 30
0 400 800
10 20 30
0 50 150
0 600
10 25
0 100