Describing Data Part 2

Last time

Population Variability

Sums of squares

$$SS = \Sigma(X_i - \mu_x)^2$$

Variance

$$\sigma^2 = \frac{\Sigma(X_i - \mu_x)^2}{N} = \frac{SS}{N}$$

Standard devation

$$\sigma = \sqrt{\frac{\Sigma(X_i - \mu_x)^2}{N}} = \sqrt{\frac{SS}{N}} = \sqrt{\sigma^2}$$

Sample variability

Sums of squares

$$SS = \Sigma(X_i - \bar{X})^2$$

Variance

$$s^2 = \frac{\Sigma(X_i - \bar{X})^2}{N-1} = \frac{SS}{N-1}$$

Standard devation

$$s = \sqrt{\frac{\Sigma(X_i - \bar{X})^2}{N-1}} = \sqrt{\frac{SS}{N-1}} = \sqrt{s^2}$$

Bi-variate descriptives

Covariation

"Sum of the cross-products"

Population

$$SP_{XY} = \Sigma(X_i - \mu_X)(Y_i - \mu_Y)$$

Sample

$$SP_{XY} = \Sigma(X_i - \bar{X})(Y_i - \bar{Y})$$

Covariance

Sort of like the variance of two variables

Population

$$\sigma_{XY} = \frac{\Sigma(X_i - \mu_X)(Y_i - \mu_Y)}{N}$$

Sample

$$s_{XY} = cov_{XY} = \frac{\Sigma(X_i - \bar{X})(Y_i - \bar{Y})}{N - 1}$$

Covariance table

$$\mathbf{K_{XX}} = \begin{bmatrix} \sigma_X^2 & cov_{XY} & cov_{XZ} \\ cov_{YX} & \sigma_Y^2 & cov_{YZ} \\ cov_{ZX} & cov_{ZY} & \sigma_Z^2 \end{bmatrix}$$

$cov_{xy} = cov_{yx}$

Covariance table

$$\mathbf{K_{XX}} = \begin{bmatrix} \sigma_X^2 & 126.5 & 5.2 \\ 126.5 & \sigma_Y^2 & cov_{YZ} \\ 5.2 & cov_{ZY} & \sigma_Z^2 \end{bmatrix}$$

Which variable, Y or Z, does X have greater relationship with?

Can't know because you don't know what units they're measured in!

Correlation

- Measure of association
- How much two variables are *linearly* related
- -1 to 1
- Sign indicates direction of relationship
- Invariant to changes in mean or scaling

Correlation

Pearson product moment correlation

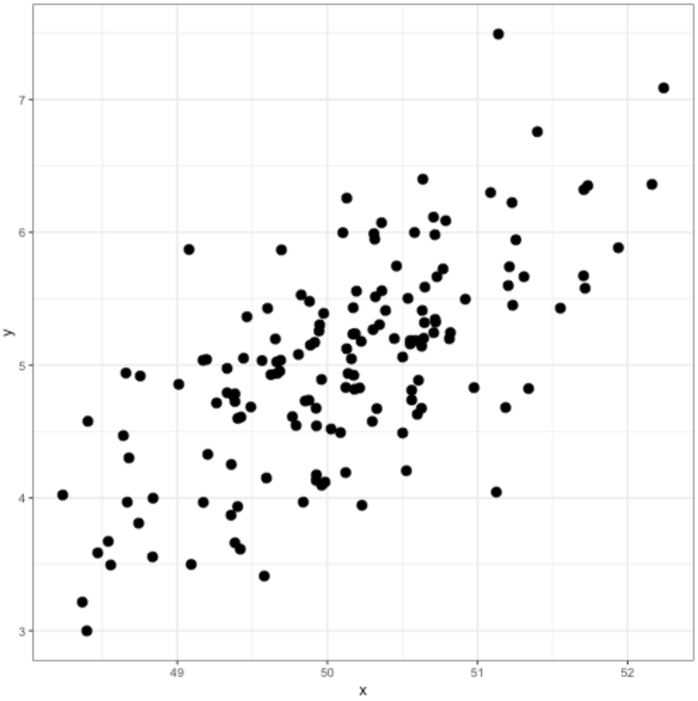
Population

$$\rho_{XY} = \frac{\Sigma z_X z_Y}{N} = \frac{SP}{\sqrt{SS_X}\sqrt{SS_Y}} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Sample

$$r_{XY} = \frac{\Sigma z_X z_Y}{n-1} = \frac{SP}{\sqrt{SS_X}\sqrt{SS_Y}} = \frac{s_{XY}}{s_X s_Y}$$

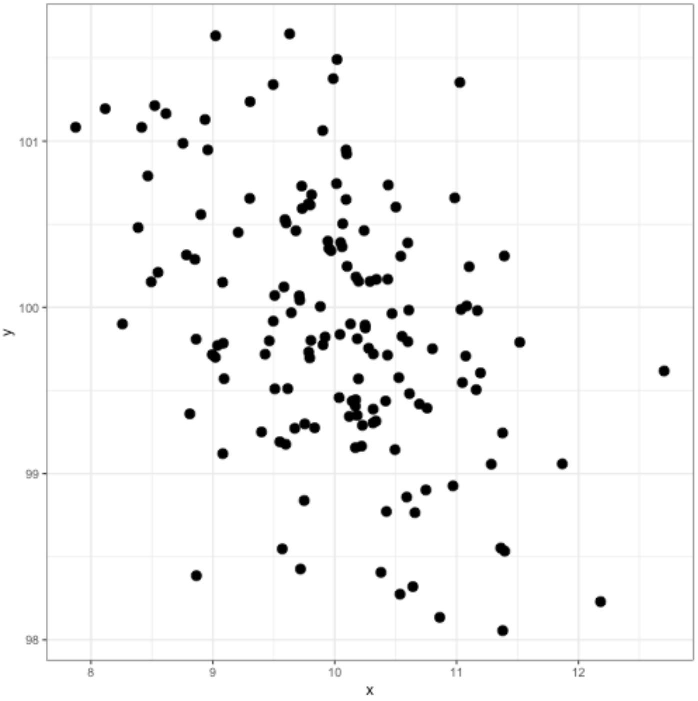
```r
data %>% ggplot(aes(x = x, y = y)) + geom_point(size = 3) + theme_bw()
```



What is the correlation between these two variables?

Correlation = 0.68

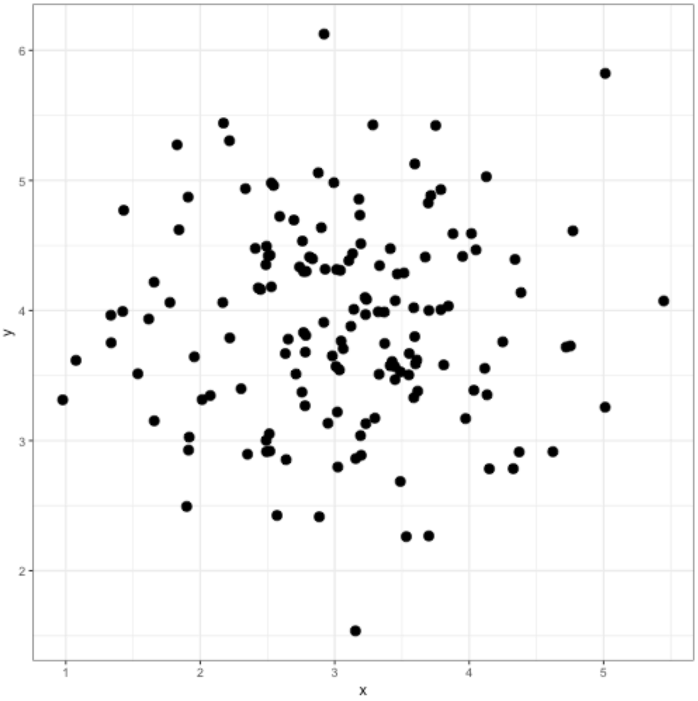
```r
data %>% ggplot(aes(x = x, y = y)) + geom_point(size = 3) + theme_bw()
```



What is the correlation between these two variables?

Correlation = -0.41

```r
data %>% ggplot(aes(x = x, y = y)) + geom_point(size = 3) + theme_bw()
```



What is the correlation between these two variables?

Correlation = 0

Effect size

- Recall that *z*-scores allow us to compare across units of measure; the products of standardized scores are themselves standardized.

- The correlation coefficient is a **standardized effect size** which can be used communicate the strength of a relationship.

- Correlations can be compared across studies, measures, constructs, time.

- Example: the correlation between age and height among children is $r = .70$. The correlation between self- and other-ratings of extraversion is $r = .25$.

What is a large correlation?

- Cohen (1988): .1 (small), .3 (medium), .5 (large)
  - Often forgot: Cohen said only to use them when you had nothing else to go on, and has since regretted even suggesting benchmarks to begin with.
- r^2: Proportion of variance "explained"
  - as Ozer & Funder (2019) discuss, we're not really explaining anything and the change in scale can mess up our interpretations if we're not careful.

What are good benchmarks?

From Ozer & Funder (2019)

- Classic social psych studies: $r = .36 - .42$
- Scarcity increases the perceived alue of a commodity $r = .12$
- People attribute failures to bad luck $r = .10$
- Communicators perceived as more credible are more persuasive $r = .10$
- People in a bad mood are more aggressive $r = .41$
- Antihistamine and symptom relief $r = .11$
- Ibuprofen and pain relief $r = .14$
- Height and weight $r = .44$

What are good benchmarks?

Implications

- Don't dismiss small effects
- Be skeptical of large effects

Recommendations

- Report effect sizes
- Use large samples -- remember bias?
- Report effect sizes in context
- Stop using empty terminology
- Revise guidelines

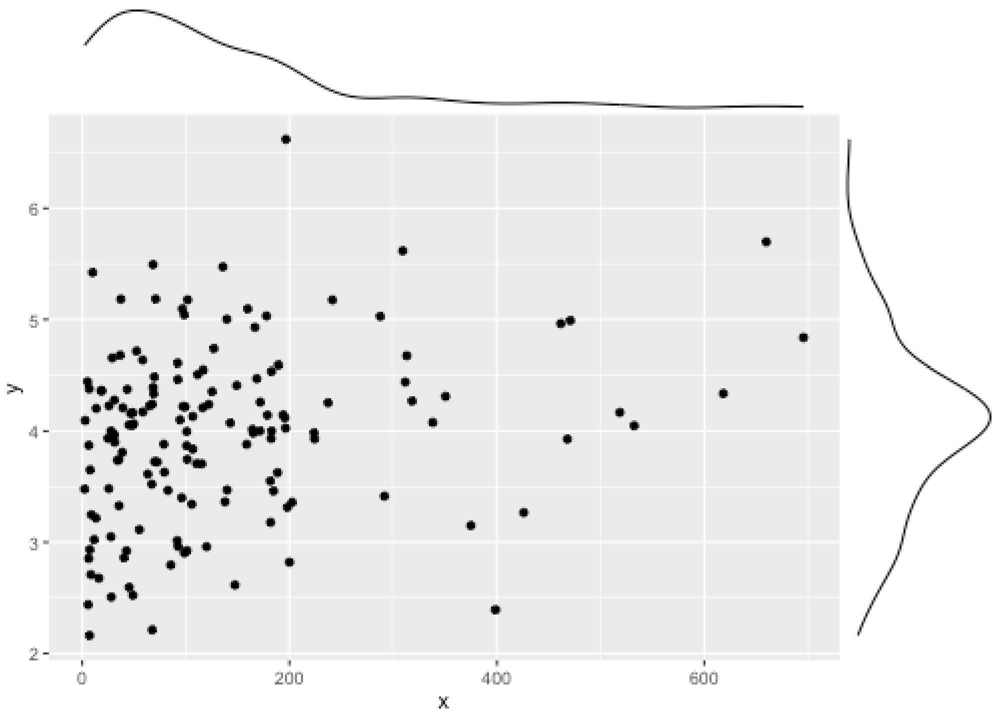
What affects correlations?

It's not enough to calculate a correlation between two variables. You should always look at a figure of the data to make sure the number accurately describes the relationship. Correlations can be easily fooled by qualities of your data, like:

- Skewed distributions
- Outliers
- Restriction of range
- Nonlinearity

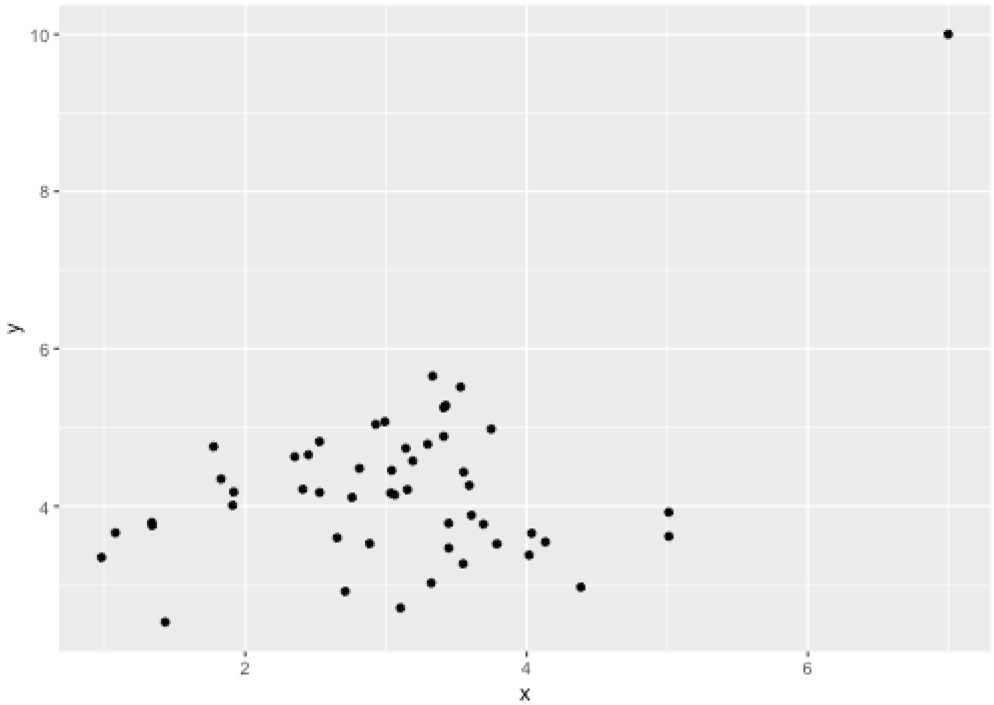
Skewed distributions

```
p = data %>% ggplot(aes(x=x, y=y)) + geom_point()
ggMarginal(p, type = "density")
```



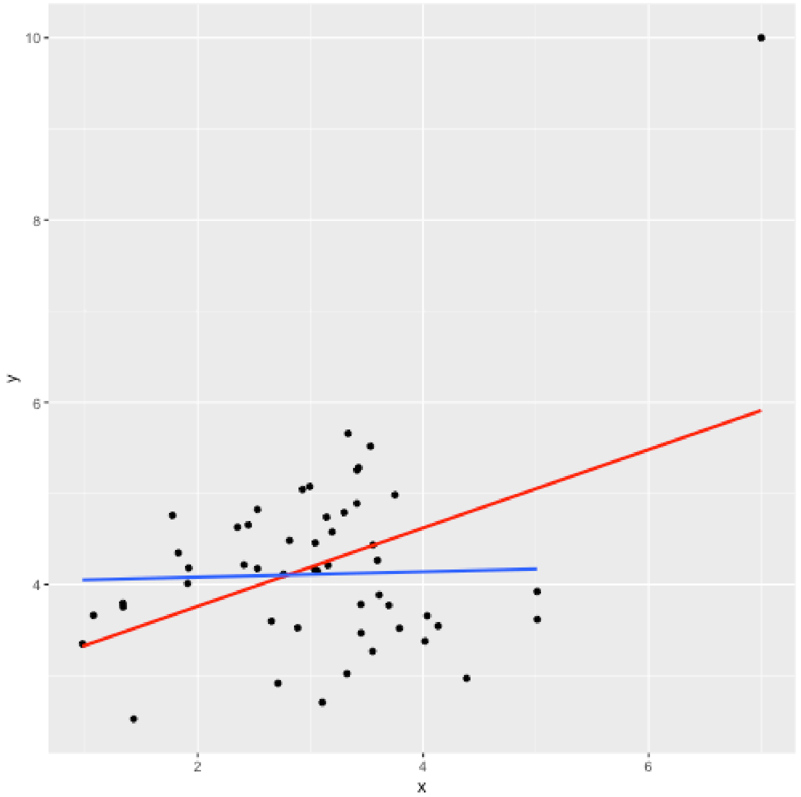
Outliers

```
data %>% ggplot(aes(x=x, y=y)) + geom_point()
```



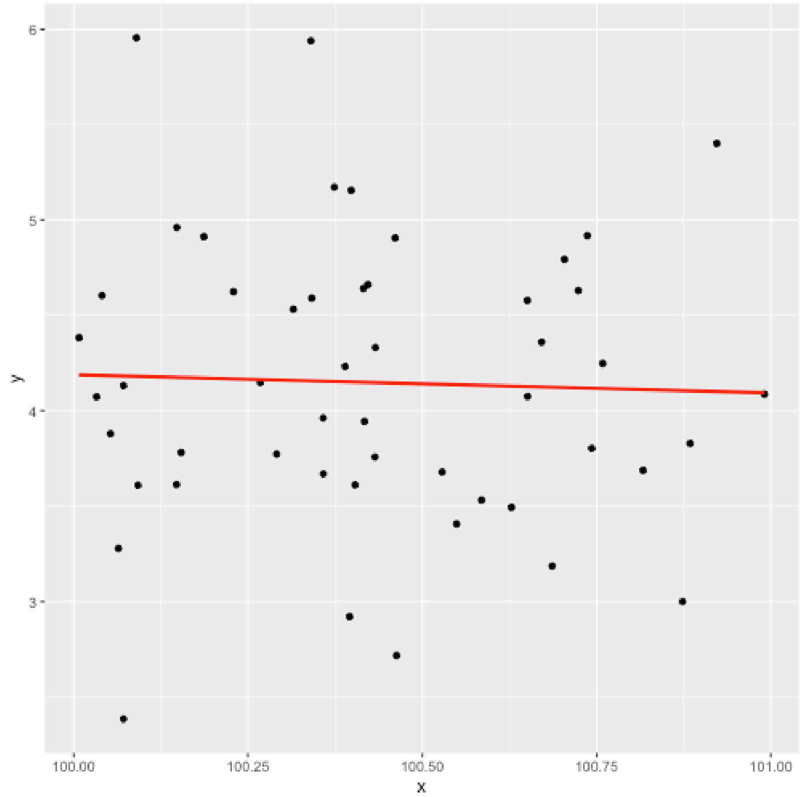
Outliers

```
data %>% ggplot(aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              color = "red") +
  geom_smooth(data = data[-51,],
              method = "lm",
              se = FALSE)
```



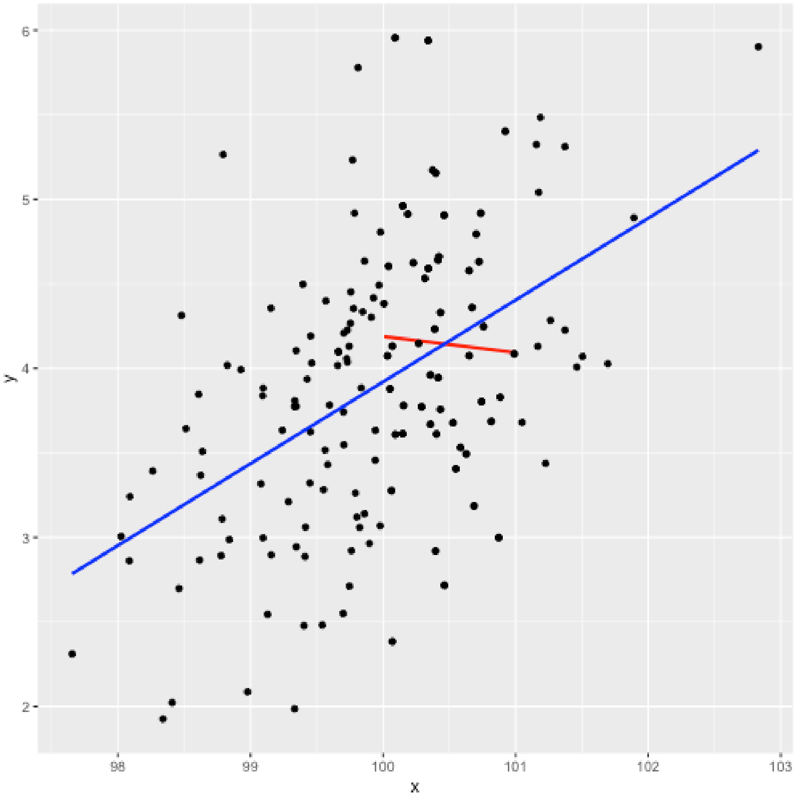
Restriction of range

```r
data %>%
ggplot(aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              color = "red")
```

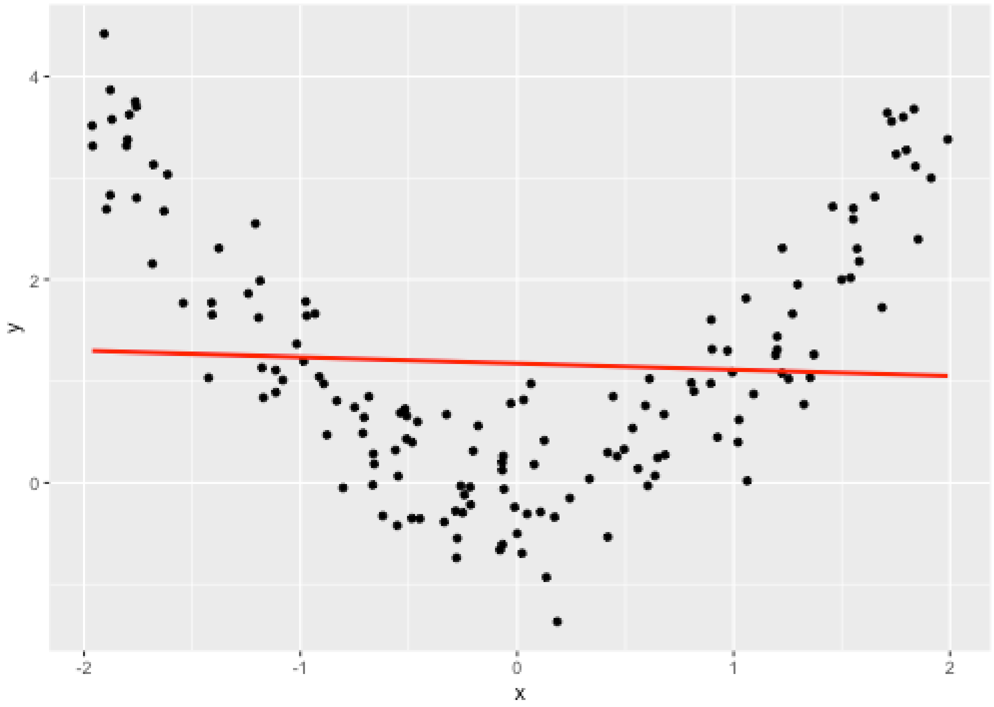


Restriction of range

```
data %>%
ggplot(aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              color = "red") +
  geom_point(data = real_data) +
  geom_smooth(method = "lm",
              se = FALSE,
              data = real_data,
              color = "blue")
```



Nonlinearity

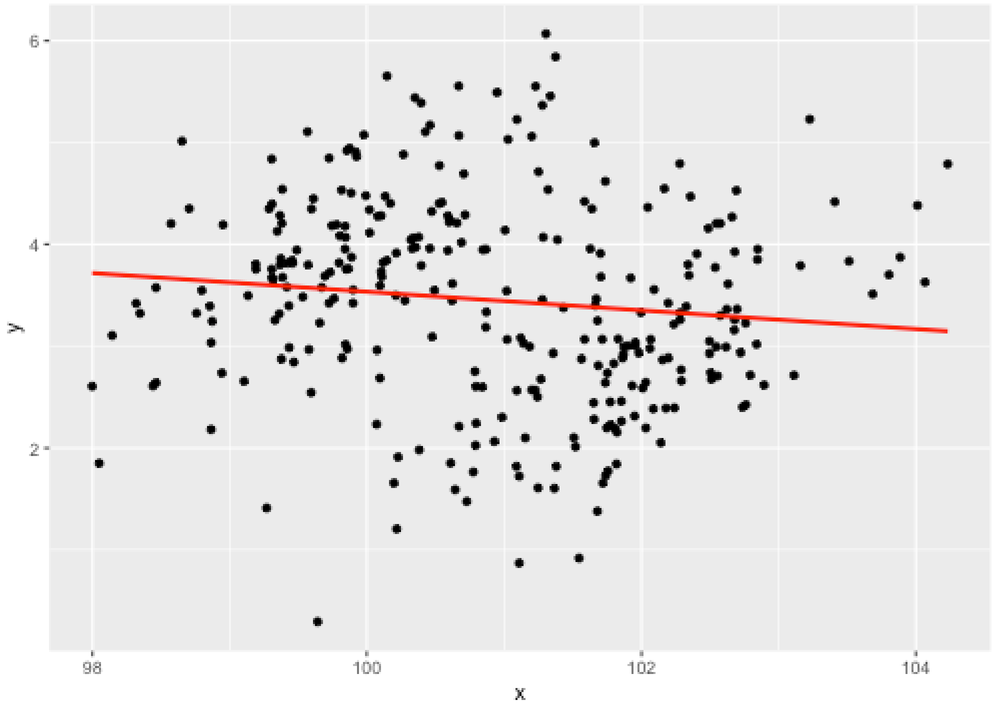


It's not always apparent

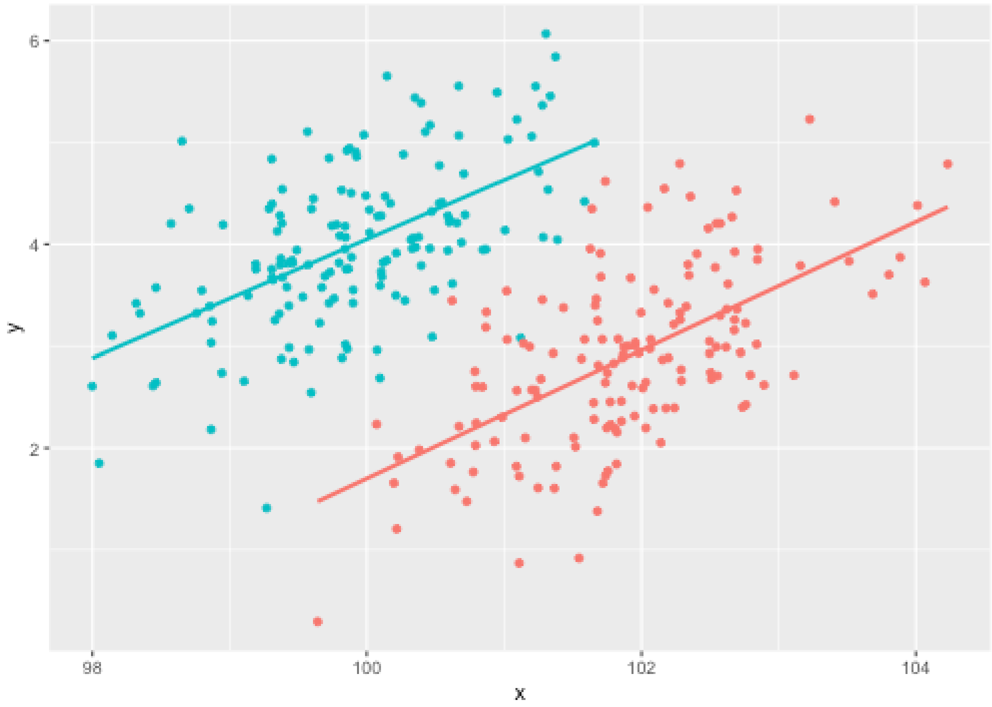
Sometimes issues that affect correlations won't appear in your graph, but you still need to know how to look for them.

- Low reliability
- Content overlap
- Multiple groups

Multiple groups



Multiple groups



Known as **Simpson's Paradox**

Special cases of the Pearson correlation

- **Spearman/Kendall correlation coefficient**
  - Applies when both X and Y are ranks (ordinal data) instead of continuous
  - Spearman for larger samples, Kendall for smaller samples (or a lot of ties in rank ordering).
- **Point-biserial correlation coefficient**
  - Applies when Y is binary.
    - NOTE: This is not an appropriate statistic when you artificially dichotomize data.
- **Phi (ϕ) coefficient**
  - Both X and Y are dichotomous.

Do the special cases matter?

For Spearman, you'll get a different answer.

```r
x = rnorm(n = 10); y = rnorm(n = 10) #randomly generate 10 numbers from
```

```r
head(cbind(x,y))
```

```
##               x          y
## [1,] -0.6682733 -0.3940594
## [2,] -1.7517951  0.9581278
## [3,]  0.6142317  0.8819954
## [4,] -0.9365643 -1.7716136
## [5,] -2.1505726 -1.4557637
## [6,] -0.3593537 -1.2175787
```

```r
cor(x,y, method = "pearson")
```

```
## [1] 0.2702894
```

```r
head(cbind(x,y, rank(x), rank(y)))
```

```
##               x          y
## [1,] -0.6682733 -0.3940594  5 7
## [2,] -1.7517951  0.9581278  2 9
## [3,]  0.6142317  0.8819954 10 8
## [4,] -0.9365643 -1.7716136  4 3
## [5,] -2.1505726 -1.4557637  1 4
## [6,] -0.3593537 -1.2175787  7 6
```

```r
cor(x,y, method = "spearman")
```

```
## [1] 0.3454545
```

Do the special cases matter?

If your data are naturally binary, no difference between Pearson and point-biserial.

```
x = rnorm(n = 10); y = rbinom(n = 10, size = 1, prob = .3)
head(cbind(x,y))
```

```
##                x y
## [1,] -0.48974849 1
## [2,] -2.53667101 0
## [3,]  0.03521883 1
## [4,]  0.03043436 0
## [5,] -0.27043857 0
## [6,] -0.55228283 1
```

```
cor(x,y, method = "pearson")
```

```
## [1] 0.1079188
```

```
ltm::biserial.cor(x,y, level = 2)
```

```
## [1] 0.1079188
```

Do the special cases matter?

If your data are artificially binary, there can be big differences.

```
x = rnorm(n = 10); y = rnorm(n = 10)
```

```
head(cbind(x,y))
```

```
##                x          y
## [1,]  1.27516603 -0.2012149
## [2,] -1.55729177  0.2925842
## [3,]  0.09364959  0.0821713
## [4,]  0.87343693  0.1879078
## [5,]  0.74807054  0.3794815
## [6,]  0.02831971 -1.2940189
```

```
cor(x,y, method = "pearson")
```

```
## [1] -0.1584301
```

```
d_y = ifelse(y < median(y), 0, 1)
head(cbind(x,y, d_y))
```

```
##                x          y d_y
## [1,]  1.27516603 -0.2012149   0
## [2,] -1.55729177  0.2925842   1
## [3,]  0.09364959  0.0821713   0
## [4,]  0.87343693  0.1879078   0
## [5,]  0.74807054  0.3794815   1
## [6,]  0.02831971 -1.2940189   0
```

```
ltm::biserial.cor(x,d_y, level = 2
```

```
## [1] -0.4079477
```

Next time...

Probability