Programming with R — A Beginners' Guide for Geoscientists

3 - Statistics

Tobias Stephan

09/02/2022

Contents

```r
pacman::p_load(
  # data manipulation
  dplyr,

  # Plotting
  ggplot2
)

# my functions and stuff for the course
source("R/read_geochron.R")
```

Descriptive statistics

One variable (Univariate statistics)

The goal of univariate statistics is to explore the spread (or distribution) of a variable.

Background

Types of Variables:

- *Categorical* = qualitative variables
  - *Nominal*: variable has two or more categories, but there is no ordering (*rank*) (e.g. binary variables (0 vs. 1, yes vs. no, true vs. false), blood type (A, B, AB, or O) rock type (magmatic, sedimentary, or metamorphic)).
  - *Ordinal*: similar as categorical, but there is a clear ordering (or rank) of the categories (e.g. economic status low-medium-high,).
  - *Discrete numerical*: variable can only be a finite number of real values within a given interval (e.g. a score of a judge between 0 and 10).
- *Continuous (numerical)* = quantitative variables (Similar to discrete numerical values, except that the variable can be any an infinite number of real values within a given interval)

  - *Intervals*: ordered units that have the same difference (e. g. temperatures in degrees Celsius or Fahrenheit as difference between 20°C and 30°C is the same as 30°C to 40°C)
  - *Ratios*: same as intervals but with an absolute zero (none of a variable). E.g. temperature in K, distance, weight, age, ...

```r
data <- read_geochron("Data/Geochron_sample_download_UPb.xls")
samples <- data$isotopes |>
  mutate(st.Pb206U238.perc = st.Pb206U238 / t.Pb206U238)

sample1 <- samples |>
  filter(Sample_ID == "Whitehorse Formation")
```

Important values to characterize the distribution of the variable `sample1$st.Pb206U238.perc` are:

```r
# minimum maximum
min(sample1$st.Pb206U238.perc)
```

```
## [1] 0.008135829
```

```r
# maximum
max(sample1$st.Pb206U238.perc)
```

```
## [1] 0.06985111
```

```r
# mean
mean(sample1$st.Pb206U238.perc)
```

```
## [1] 0.01932261
```

```r
# median
median(sample1$st.Pb206U238.perc)
```

```
## [1] 0.01752407
```

```r
# quantiles
quantile(sample1$st.Pb206U238.perc)
```

```
##          0%         25%         50%         75%        100%
## 0.008135829 0.012976014 0.017524072 0.022604296 0.069851111
```

```r
# variance
var(sample1$st.Pb206U238.perc)
```

```
## [1] 9.393132e-05
```

```r
# standard deviation
sd(sample1$st.Pb206U238.perc)
```

```
## [1] 0.009691817
```

Mathematical background

- Mean $|X| = \frac{\sum(x_i)}{n}$
- Variance $\sigma_X^2 = \frac{1}{n}\sum(x_i - |X|)^2$
- Standard deviation $\sigma_X = \sqrt{\sigma_X^2}$

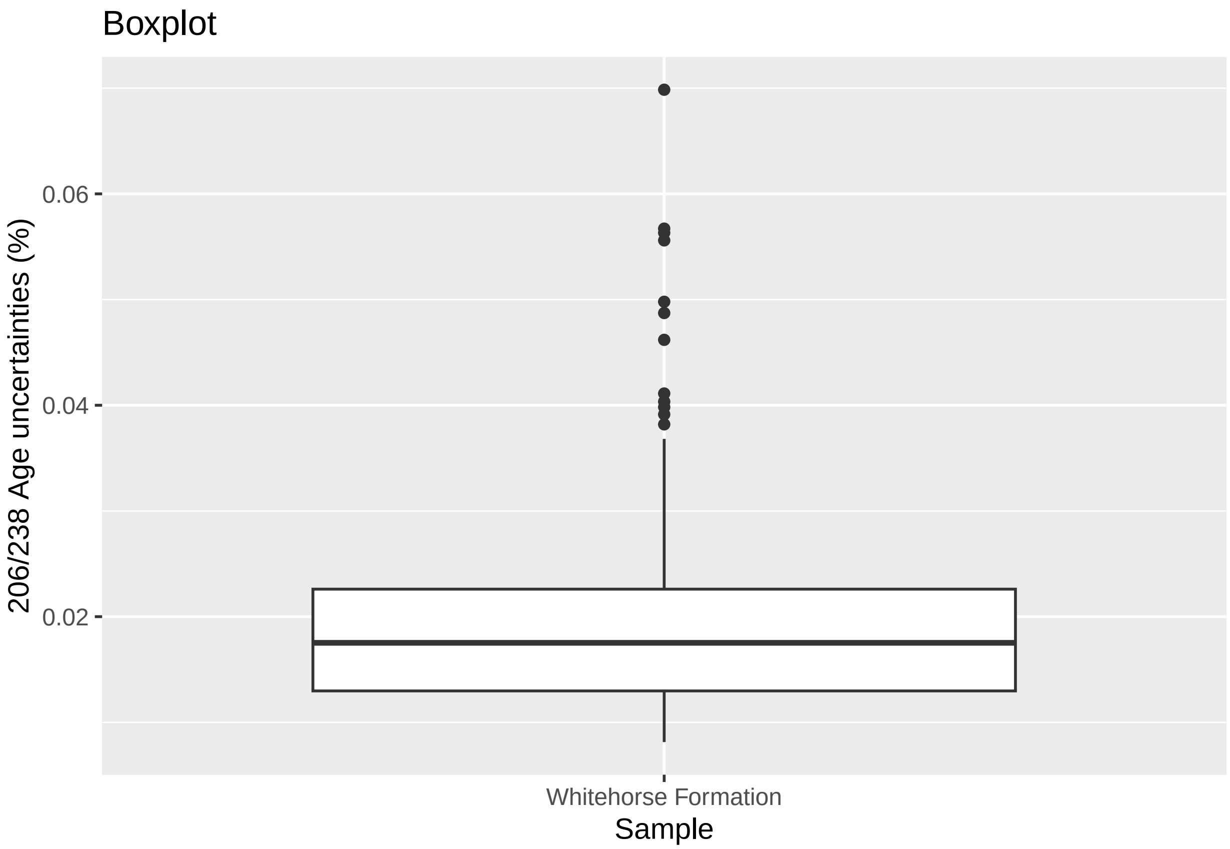
x the variable/observation

n is the number of observations (sample size)

Visualization of the distibution

```
# simple plot
# boxplot(sample1$st.Pb206U238.perc)

# ggplot
ggplot(data = sample1, aes(x = Sample_ID, y = st.Pb206U238.perc)) +
  geom_boxplot() +
  labs(title = "Boxplot", x = "Sample", y = "206/238 Age uncertainties (%)")
```

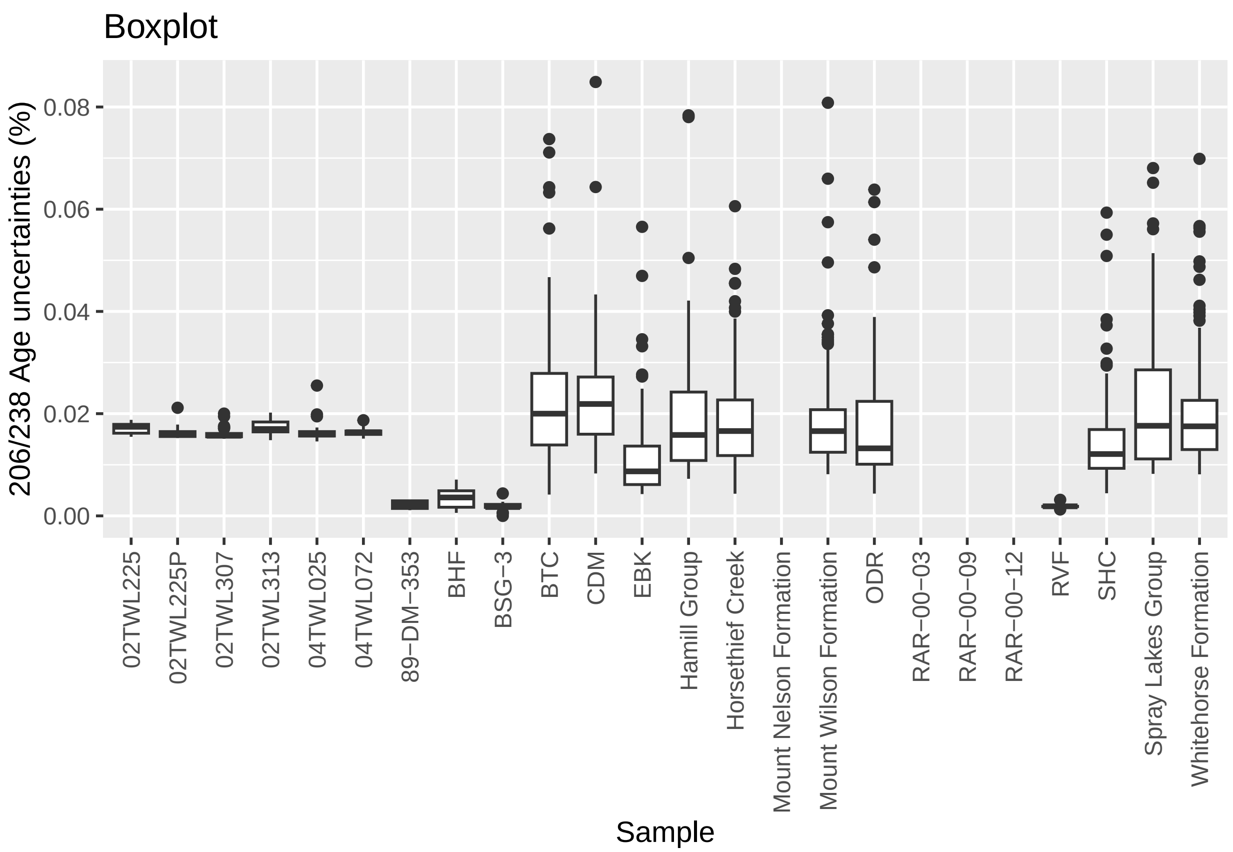


Boxplot

Btw, if you want to have the boxplot for more than one sample… R automatically identifies when there are more than one sample in your data set.

```
ggplot(data = samples, aes(x = Sample_ID, y = st.Pb206U238.perc)) +
  geom_boxplot() +
  labs(title = "Boxplot", x = "Sample", y = "206/238 Age uncertainties (%)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

```
## Warning: Removed 136 rows containing non-finite values (`stat_boxplot()`).
```

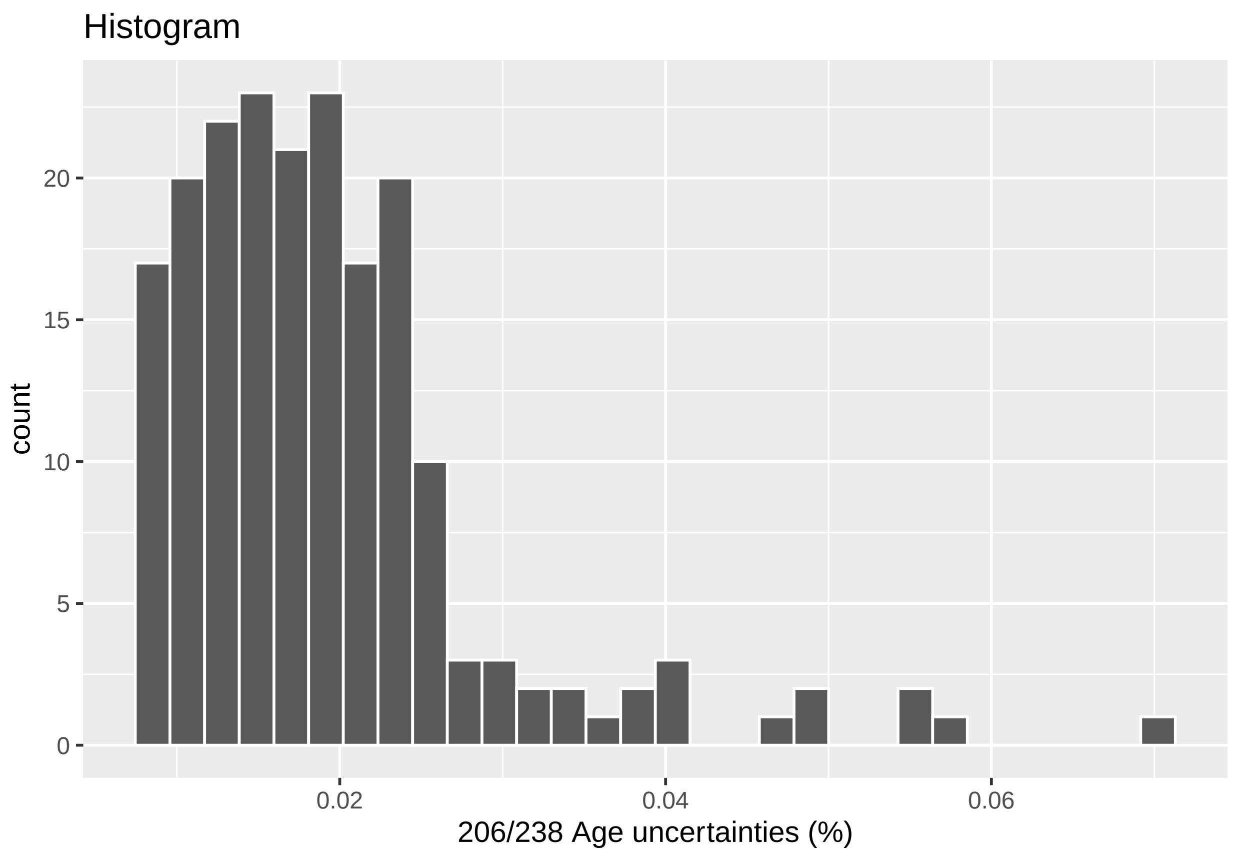


```
# simple plot
# hist(sample1$st.Pb206U238.perc)

# ggplot
ggplot(data = sample1, aes(x = st.Pb206U238.perc)) +
  geom_histogram(color = "white") + # adds the histogram
  labs(title = "Histogram", x = "206/238 Age uncertainties (%)")
```

Histogram

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



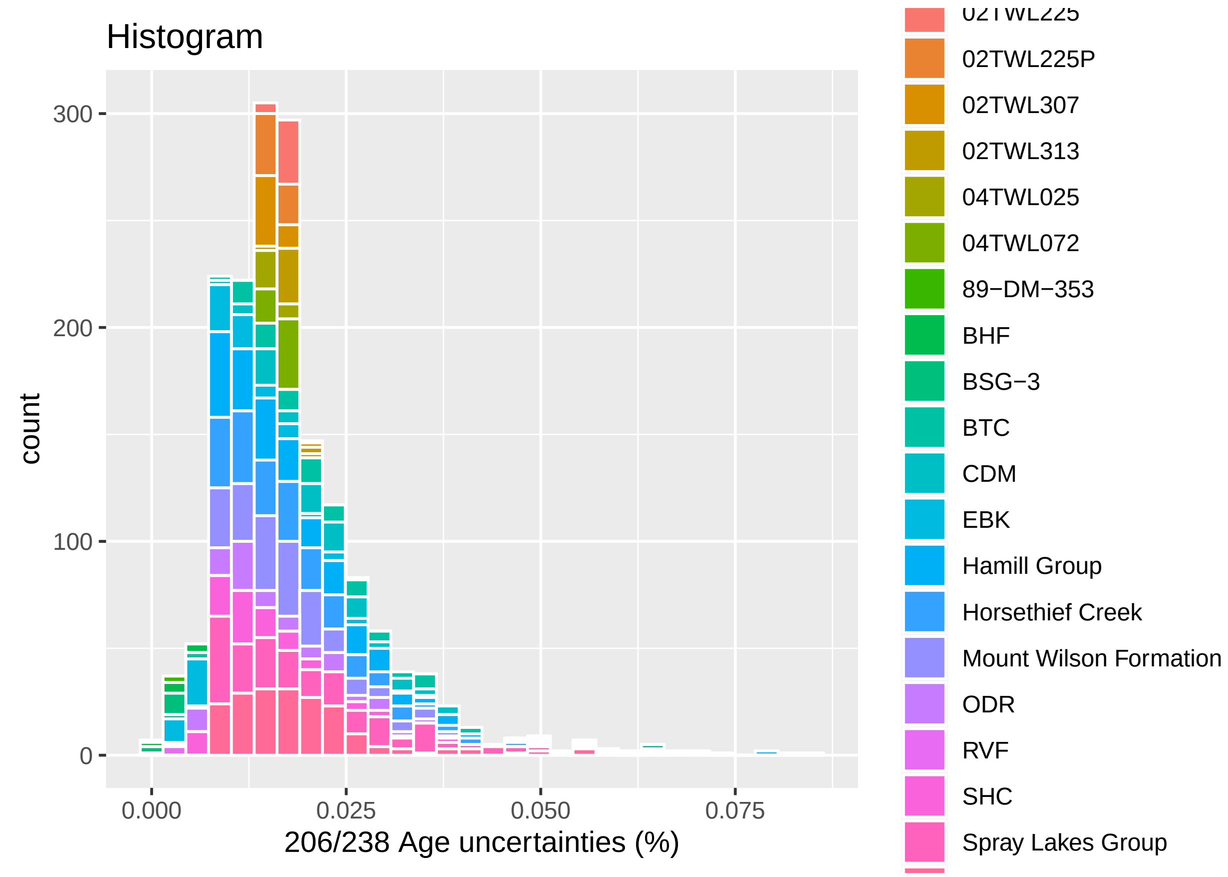
R finds the optimal width of the bins automatically. You can also give a specific bin width by defining `binwidth` or the number of bins by `bins` as options int in the `geom_histogram()` function. See `?geom_histogram()` for more information

If you want to show more than one sample in one histogram plot and color them according to their sample, just define the `fill` option.

```r
ggplot(data = samples, aes(x = st.Pb206U238.perc, fill = Sample_ID)) +
  geom_histogram(color = "white") + # adds the histogram
  labs(title = "Histogram", x = "206/238 Age uncertainties (%)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 136 rows containing non-finite values (`stat_bin()`).
```



Statistical tests How to check the normality? It's possible to use the **Shapiro-Wilk normality test** and to look at the normality plot.

Shapiro-Wilk test:

- Null hypothesis: the data are normally distributed
- Alternative hypothesis: the data are not normally distributed

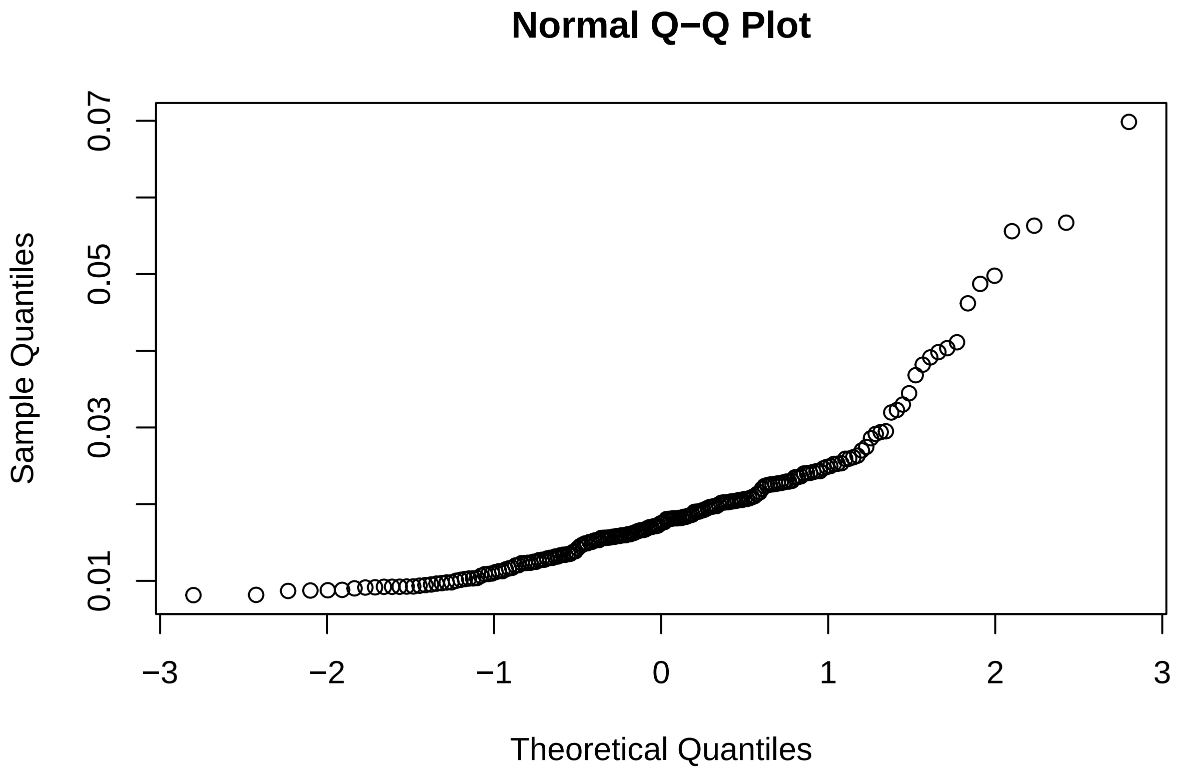
```
shapiro.test(sample1$st.Pb206U238.perc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample1$st.Pb206U238.perc
## W = 0.80824, p-value = 8.734e-15
```

From the output, the p-value is less than the significance level 0.05 implying that the distribution of the data is significantly different from a normal distribution.

Visual inspection of the data normality using **Q-Q plots** (quantile-quantile plots). Q-Q plot draws the correlation between a given sample and the normal distribution.

```
# simple R plot
qqnorm(sample1$st.Pb206U238.perc)
```



From the normality plots, we conclude that the data does not come from normal distributions.

Does the mean has any value for the variable?

- Normally distributed values: One Sample t-test `t.texst()`
- Non normally distributed values: Non parametric one-sample Wilcoxon rank test `wilcox.test()`

```
# t.test(sample1$st.Pb206U238.perc)
wilcox.test(sample1$st.Pb206U238.perc)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  sample1$st.Pb206U238.perc
## V = 19306, p-value < 2.2e-16
## alternative hypothesis: true location is not equal to 0
```

The **p-value** of the test is 2.2x10^{-16}, which is less than the significance level alpha = 0.05. We can conclude that the mean uncertainties of the reported uncertainties is significantly different from 0 with a **p-value** = 2.2x10^{-16}.

Two variables (Bivariate statistics)

The goal of bivariate statistics is to explore how two different variables relate to or differ from each other.

Values:

- **Covariance** indicates the direction of the linear relationship between variables.
- **Correlation** measures both the strength and direction of the linear relationship between two variables. The correlation coefficient **R** ranges between 0 (*variables do not correlate*) and 1 (*variables correlate*).

Mathematical background

- Covariance $\sigma_{XY} = \frac{1}{n}\sum(x_i - |X|)(y_i - |Y|)$
- Correlation $R_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$

```
# covariance
# cov(sample1$Age_206.238, sample1$Age_206.238)
# identical to var() with two input variables
var(sample1$st.Pb206U238, sample1$st.Pb207U235)
```

```
## [1] 75.68803
```

```
# correlation
cor(sample1$st.Pb206U238, sample1$st.Pb207U235, method = "pearson")
```

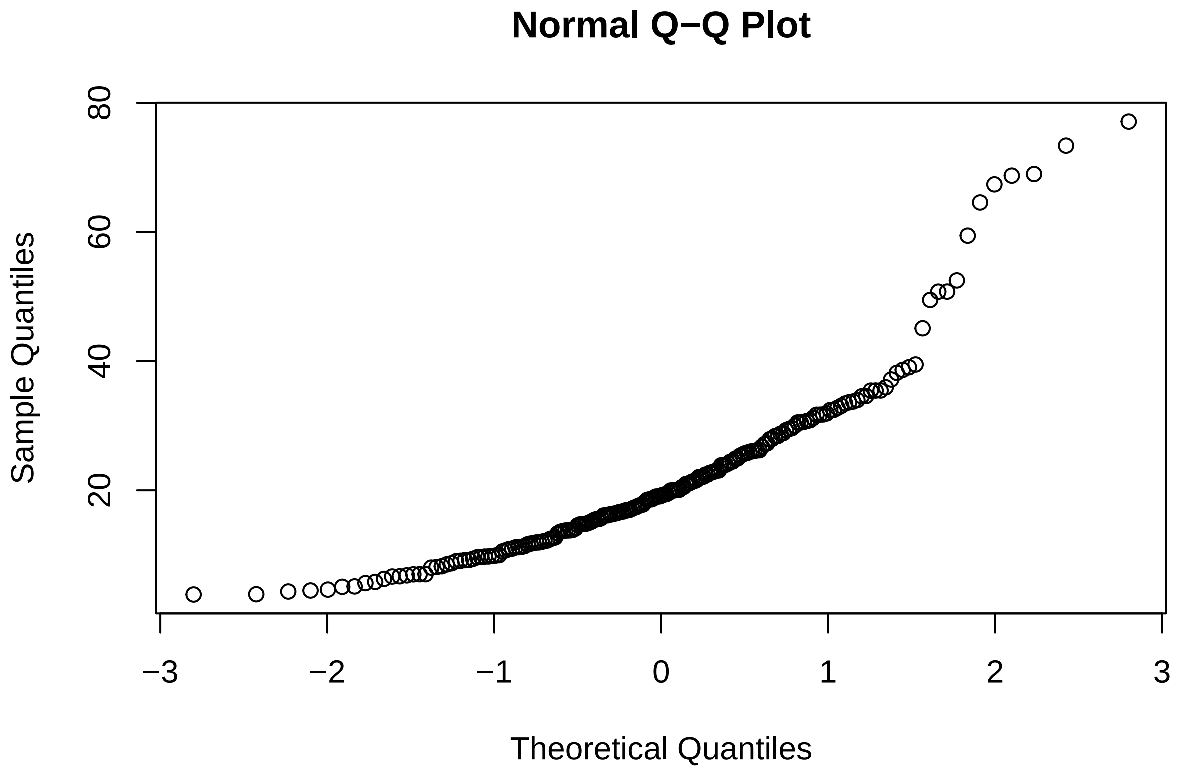
```
## [1] 0.6424031
```

Background

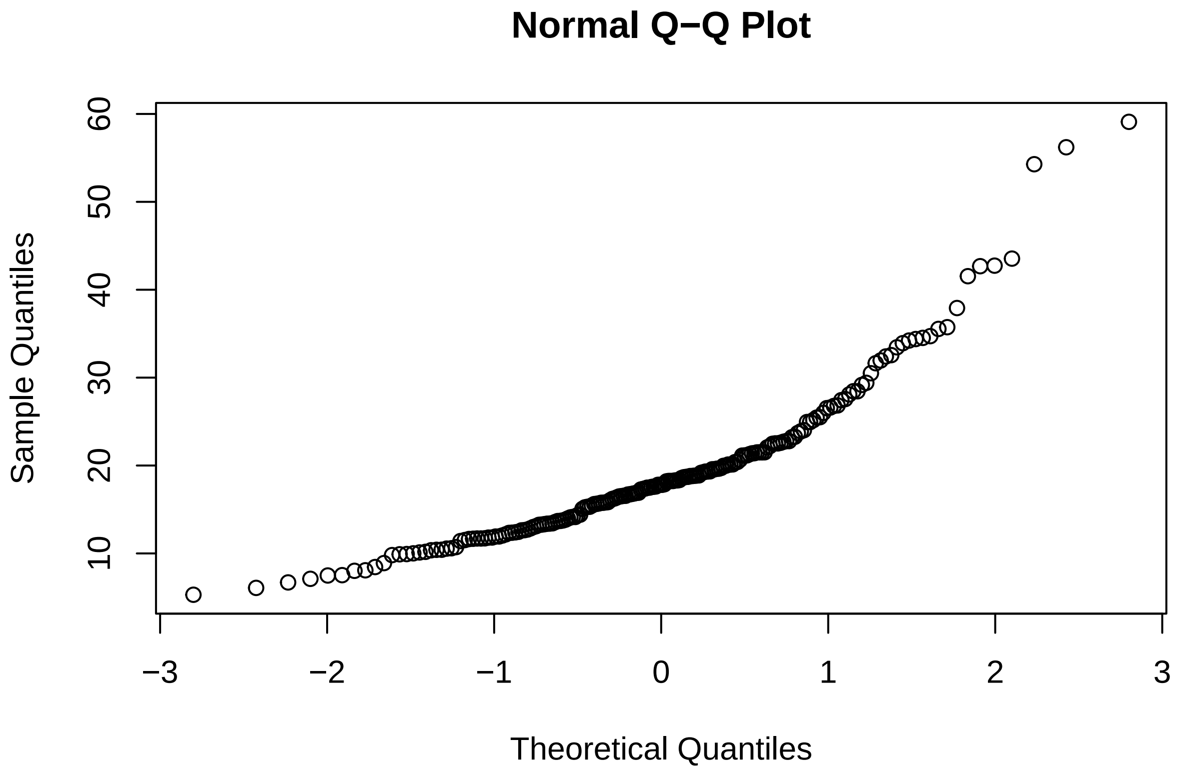
The *Pearson correlation (R)* (`method = 'pearson'`) measures a linear dependence between two variables. It's also known as a **parametric correlation** test because it depends to the distribution of the data: it can be used only when x and y are from *normal distribution*!

If both variables do not come from a bivariate normal distribution, you have to use rank-based correlation coefficients (**non-parametric**), such as the *Kendall rank correlation* (`method = 'kendall'`) or the *Spearman's rho statistic* (`method = 'spearman'`).

```
qqnorm(sample1$st.Pb206U238)
```



```
qqnorm(sample1$st.Pb207U235)
```



From the normality plots, we conclude that both populations may come from normal distributions.

Correlation test Testing the statistical significance of the correlation:

```
cor.test(sample1$st.Pb206U238, sample1$st.Pb207U235, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  sample1$st.Pb206U238 and sample1$st.Pb207U235
## t = 11.675, df = 194, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5519445 0.7179183
## sample estimates:
##       cor
## 0.6424031
```

In the result above :

- `t` is the t-test statistic value (t = 11.675),
- `df` is the degrees of freedom (df = 194),
- `p-value` is the significance level of the t-test (p-value < 2.2e-16).
- `conf.int` is the confidence interval of the correlation coefficient at 95% (conf.int = [0.5519445, 0.7179183]);
- `sample estimates` is the correlation coefficient (cor = 0.6424031).

The p-value of the test is <2.2x10^{-16}, which is less than the significance level alpha = 0.05. We can conclude that the two variables are significantly correlated with a correlation coefficient of 0.64 and p-value of <2.2x10^{-16}.

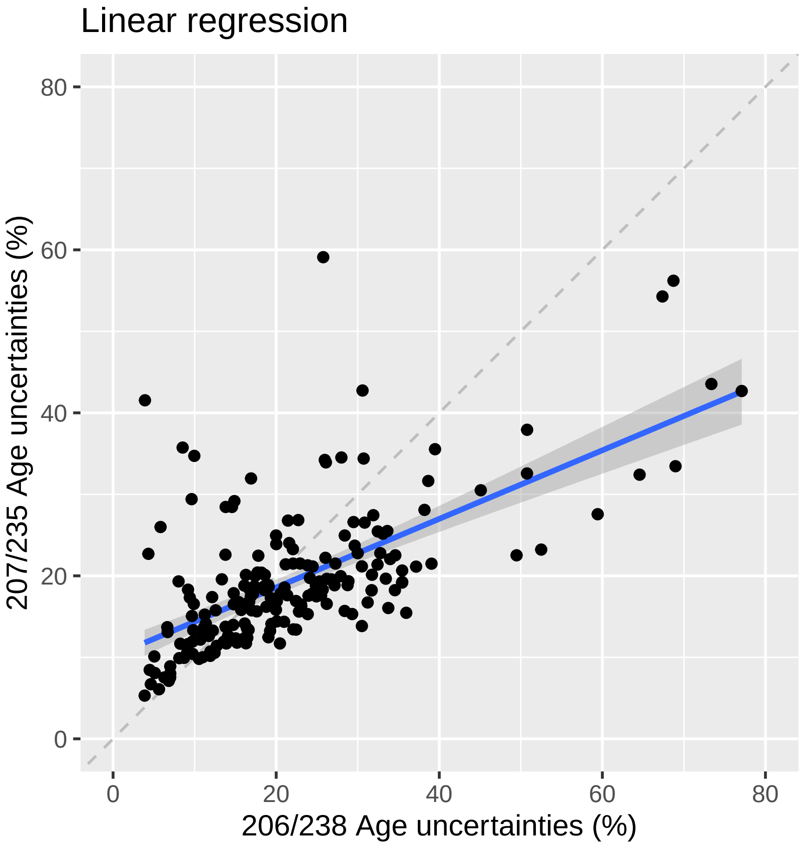
Interpretation of the correlation coefficient Correlation coefficient is comprised between -1 and 1:

- `-1` indicates a strong negative correlation : this means that every time x increases, y decreases (left panel figure)
- `0` means that there is no association between the two variables (x and y) (middle panel figure)
- `1` indicates a strong positive correlation : this means that y increases with x (right panel figure)

Visualization of the correlation

```
# simple R plot
# plot(sample1$st.Pb206U238, sample1$st.Pb207U235)

ggplot(data = sample1, aes(st.Pb206U238, st.Pb207U235)) +
  coord_fixed(xlim = c(0, 80), ylim = c(0, 80)) + # x and y axis have same scaling
  geom_abline(slope = 1, lty = 2, color = "grey") + # draws a diagonal line with slope = 1
  geom_smooth(method = "lm", formula = "y~x") + # 'lm' : linear regression
  geom_point() + # adds points
  labs(
    title = "Linear regression",
    x = "206/238 Age uncertainties (%)",
    y = "207/235 Age uncertainties (%)"
  ) # add description
```



Linear Regression

A summary of the linear regression , incl. R^2 (goodness-of-fit), can be shown via:

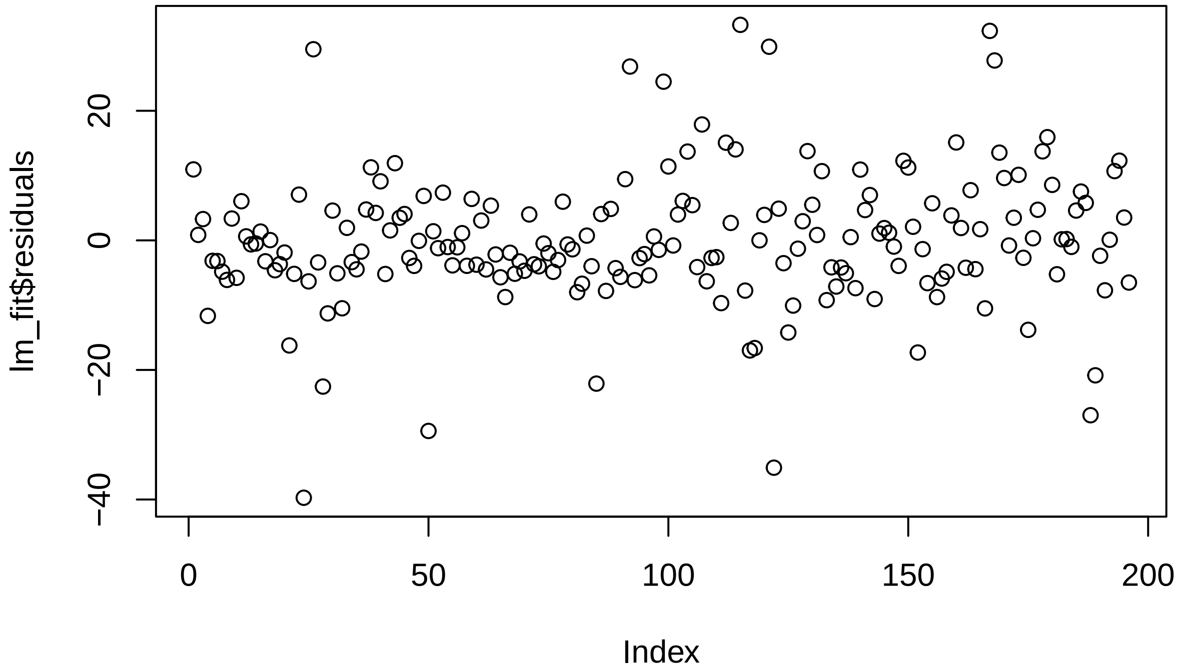
```
lm_fit <- lm(data = sample1, st.Pb206U238 ~ st.Pb207U235)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = st.Pb206U238 ~ st.Pb207U235, data = sample1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.727  -4.842  -0.877   4.732  33.267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.8962     1.7862   1.621    0.107
## st.Pb207U235   0.9808     0.0840  11.675   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.3 on 194 degrees of freedom
## Multiple R-squared:  0.4127, Adjusted R-squared:  0.4097
## F-statistic: 136.3 on 1 and 194 DF,  p-value: < 2.2e-16
```

- R: The correlation between the observed values of the response variable and the predicted values of the response variable made by the model.
- R^2: The proportion of the variance in the response variable that can be explained by the predictor variables in the regression model.

Ideally, residuals should look random. Otherwise there is a hidden pattern that the linear model is not considering. To plot the residuals, use the command `plot(lm_fit$residuals)`.

```
# simple plot
plot(lm_fit$residuals)
```



Useful resources

- **RECOMMENDED**: Hadley Wickham (the master developer of R) briefly demonstrates data analysis with R in a 22 min youtube video: https://www.youtube.com/watch?v=go5Au01Jrvs&t=10s
- Linear Regression: https://www.datacamp.com/community/tutorials/linear-regression-R
- Correlation tests: http://www.sthda.com/english/wiki/correlation-test-between-two-variables-in-r
- http://r-statistics.co/Statistical-Tests-in-R.html
- Tolosana-Delgado & Mueller (2021): "Geostatistics for Compositional Data with R", Springer, Cham
- P. Vermeesch: Lecture notes to "Statistics for geoscientists"
- Customizing ggplot Graphs: Graph defaults are fine for quick data exploration, but when you want to publish your results to a blog, paper, article or poster, you'll probably want to customize the results. Customization can improve the clarity and attractiveness of a graph

Appendix

- **One Sample t-Test**: a parametric test used to test if the mean of a sample from a normal distribution could reasonably be a specific value. `t.test()`

- **Wilcoxon Signed Rank Test**: To test the mean of a sample when normal distribution is not assumed. Wilcoxon signed rank test can be an alternative to t-Test, especially when the data sample is not assumed to follow a normal distribution. It is a non-parametric method used to test if an estimate is different from its true value. `wilcox.test()`

- **Two Sample t-Test and Wilcoxon Rank Sum Test**: Both t.Test and Wilcoxon rank test can be used to compare the mean of 2 samples. The difference is t-Test assumes the samples being tests is drawn from a normal distribution, while, Wilcoxon's rank sum test does not.

- **Shapiro Test**: To test if a sample follows a normal distribution. `shapiro.test(numericVector) # Does myVec follow a normal disbn?`
- The **Kolmogorov–Smirnov test** is used to check whether 2 samples follow the same distribution. `ks.test(x, y) # x and y are two numeric vector`
- **Fisher's F-Test** can be used to check if two samples have same variance. `var.test(x, y)  # Do x and y have the same variance?`
- **Chi Squared Test** can be used to test if two categorical variables are dependent, by means of a contingency table `chisq.test()`
- **Correlation**: To test the linear relationship of two continuous variables `cor.test(x, y)`
- The **Kruskal-Wallis Rank Sum Test** (also Kruskal–Wallis H test, or one-way ANOVA on ranks) is a non-parametric method for testing whether samples originate from the same distribution. It is used for comparing two or more independent samples of equal or different sample sizes. `kruskal.test(x)`
- More Commonly Used Tests `fisher.test(contingencyMatrix, alternative = "greater")  # Fisher's exact test to test independence of rows and columns in contingency table friedman.test()  # Friedman's rank sum non-parametric test`

There are more useful tests available in various other packages.

The package `lawstat` has a good collection. The `outliers` package has a number of test for testing for presence of outliers.

Content | previous course: Data | next course: Geochronology