Linear regression - Coursera

Di Zhen

Apr, 2020

Certification: https://coursera.org/share/04d7ce4c822ef7b7ff7d7370a376b922

Week1 Introduction to Linear Regression

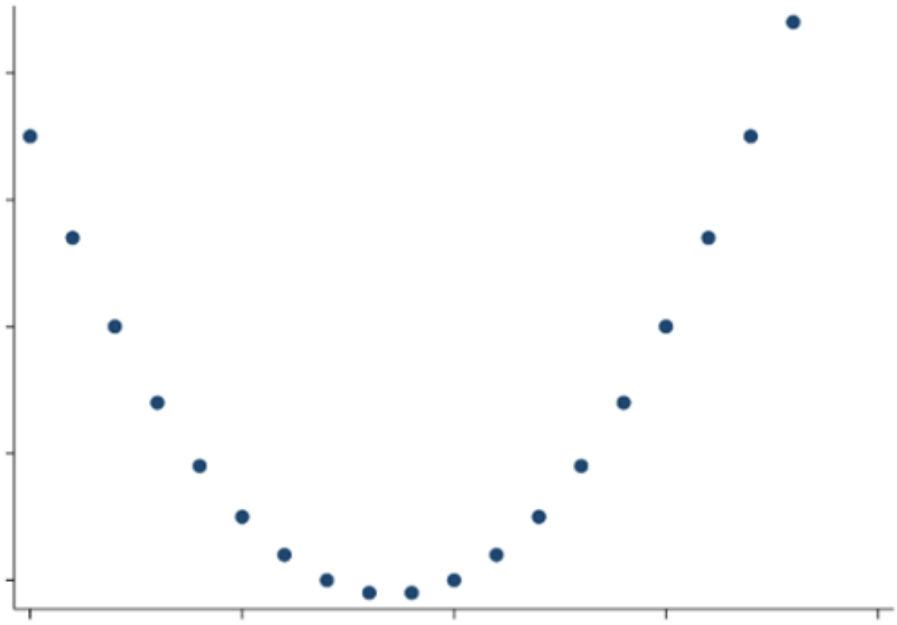
1. Warnings and precautions for Pearson's correlation

- Pearson's correlation coefficient
  - A measure of the strength of an association between two variables. Values range from -1 through to +1.

Value	Strength of Association
±0.7	Strong correlation
±0.5	Moderate correlation
±0.3	Weak correlation
±0.0	No correlation

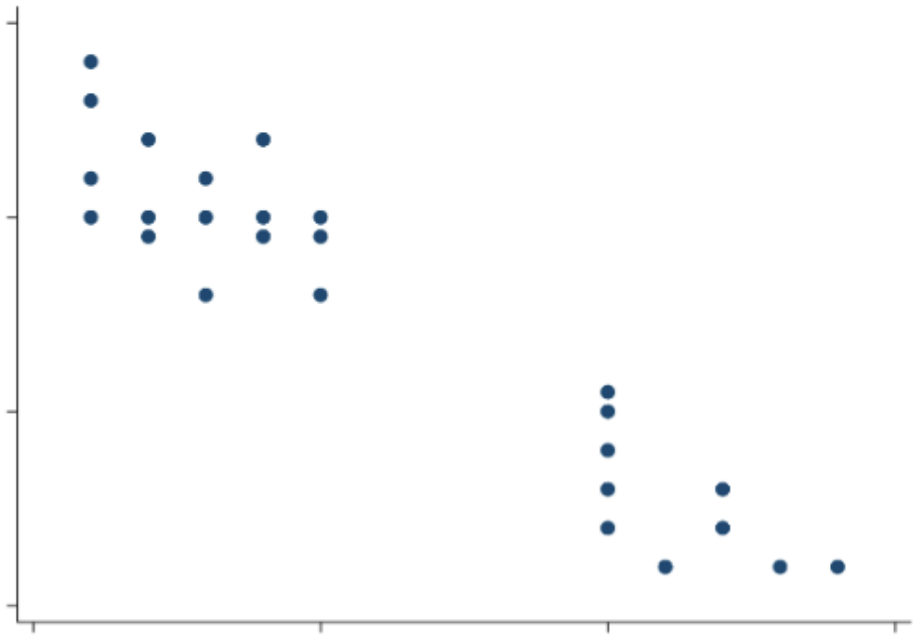
  - It is a measure of the **strength** of the linear association.

- It is **inappropriate** to calculate Pearson's correlation coefficient for the example, where the strong relationship between the variables is not a linear relationship.



Visually inspecting the data on scatterplots will avoid these mistakes. Scatter plots are also useful to spot outlying observations, which can significantly alter the estimate.

- Pearson's correlation coefficient can also be unduly influenced when there are **gaps** in the distribution.



- Summary:
  - Pearson's correlation coefficient can be misleading if these necessary conditions are not satisfied:
    - Both variables are **continuous**;
    - Observations are a **random sample** from the population;
    - Both variables are approximately **normally distributed** in the population.
  - Visual inspection can check:
    - continuous

        - normally distributed
        - linear relationship
    - Test whether the sample correlation (r) could be due to sampling variation by conducting a hypothesis test.
        - H0: no correlation between the variables in the population: ρ = 0

          H1: correlation between the variables in the population: ρ ≠ 0 Where ρ is the population correlation coefficient.
        - This significance test is extremely sensitive to sample size. Therefore, when conducting such tests you should be mindful of your sample size and the impact it is likely to have on your test result.
    - How to calculate the Pearson's correlation coefficient.
        - Consider a situation in which the two variables X and Y satisfy three conditions above. The Pearson's correlation is:

$$cor = \frac{cov(x, y)}{sd(x) \times sd(y)}$$

$$r = \frac{S_{xy}}{S_x S_y}$$

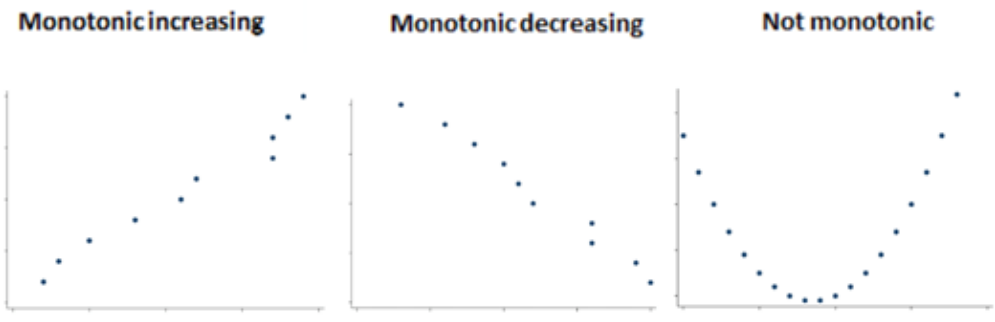
          where cor is denote by r.
        - Covariance is another measure of the strength and direction of the relationship between two variables. covariance is simply the sum of each data point's distance from the mean. Again, it is only useful as a measure of linear relationships. It is calculated as:

$$cov(x, y) = \sum_i \frac{(x_i - \bar{x}) \times (y_i - \bar{y})}{n}$$

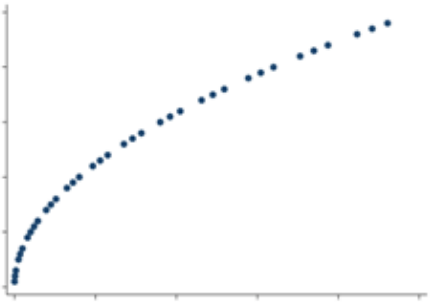
          where $\bar{x}$ and $\bar{y}$ are the sample means of each variable.
        - Therefore, **Correlation** is simply the covariance between the two variables put on a scale from -1 to 1.

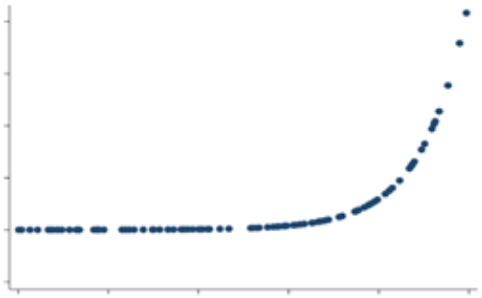
2. Introduction to Spearman correlation

- **Spearman's correlation coefficient** is the **non-parametric** version of the Pearson's correlation coefficient. Spearman's correlation measures the strength of a **monotonic relationship**.
    - **Non-parametric** means that we do not make any assumptions around the form of the data, so we do not assume they follow a specific distribution.
- Monotonic relationship
    - A monotonic relationship is one where the dependent variable (y) never decreases as the independent variable (x) increases or where the dependent variable (y) never increases as the independent variable (x) increases.



- The plots below are both non-linear, and Pearson's correlation is not appropriate. However, they are both monotonic increasing, so the Spearman's correlation is applicable.





- The conditions required for Spearman's correlation are:
  - That there is **monotonic relationship** between the two variables;
  - Both variables are either **continuous or ordinal**;
  - Observations are a **random sample** from the population.
- Difference between Pearson's correlation and Spearman's correlation:
  - Pearson's correlation uses the raw data values to calculate the coefficient; however, Spearman's **ranks** the raw values. Because there is no way to decide which of the tied observations should be ranked first, an **average of the rankings** they would have otherwise occupied is taken.
  - When there are **no tied** ranks the Spearman's correlation is

$$\rho = 1 - \frac{6 \times \sum d_i^2}{n(n^2-1)}$$

    where $d_i = R(x_i) - R(y_i)$, $R(x)$ and $R(y)$ are the ranks and n is the sample size.
  - When there are **tied** observations the Spearman's correlation is:

$$\rho = \frac{\sum_i (R(x_i) - R(\bar{x_i}))(R(y_i) - R(\bar{y_i}))}{\sqrt{\sum_i (R(x_i) - R(\bar{x_i}))^2 \sum_i (R(y_i) - R(\bar{y_i}))^2}}$$

    where $R(\bar{x_i})$ and $R(\bar{y_i})$ are the mean ranks for variable x and y respectively.
- Partial correlation
  - Partial correlation is a measure of the strength of a linear association between two continuous variables that satisfy the relevant conditions for correlation, whilst controlling or adjusting for the effect of one or more other continuous variables

Week2 Linear Regression in R

1. Assessing distributions and calculating the correlation coefficient in R

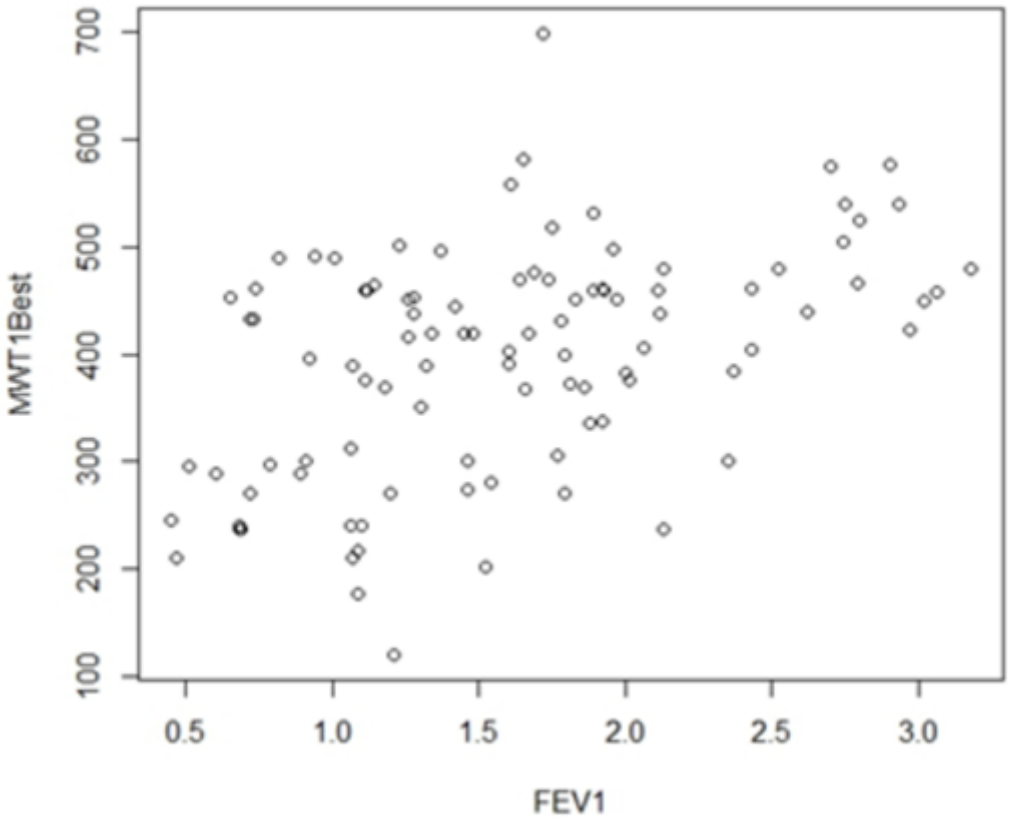
```r
# Histogram
COPD <- read.csv('COPD_student_dataset.csv')
hist(COPD$MWT1Best, main="Histogram of MWT1Best", xlab="MWT1Best", breaks=12)

# Check specific value
subset(COPD, MWT1Best > 650)
subset(COPD, MWT1Best > 600 | MWT1Best < 150)

# Summarize the data
list("Summary" = summary(COPD$MWT1Best), "Mean" = mean(COPD$MWT1Best,
na.rm=TRUE), "Standard Deviation" = sd(COPD$MWT1Best, na.rm=TRUE), "Range" =
range(COPD$MWT1Best, na.rm=TRUE), "Inter-Quartile Range" = IQR(COPD$MWT1Best,
na.rm=TRUE))

# Scatter plot
plot(COPD$FEV1, COPD$MWT1Best, xlab = "FEV1", ylab = "MWT1Best")
```

- If there is unusual value, the choice here is to check the original source from whoever collected it, or leave it as it is. You should never delete an unusual value if the value is a possible one. Deleting unusual values will bias your results and cause you to underestimate the variability in the observations
- **Descriptive statistics** and **two-way plots** can tell you more about the data. Useful statistics to summarize your data are: the mean, the standard deviation, the range, the median, and the inter-quartile range.



- Pearson's correlation coefficient

```r
# Pearson's
cor.test(COPD$FEV1,COPD$MW1Best,use = 'complete.obs',method = 'pearson')

# Spearman's
cor.test(COPD$FEV1,COPD$MW1Best,use = 'complete.obs',method = 'spearman')
```

2. Fit a linear regression in R

- The basic format of a linear regression is:

  Y = α + β1*X + ε

  Where:

  · Y = outcome (i.e. dependent) variable

  · X = predictor (i.e. independent) variable

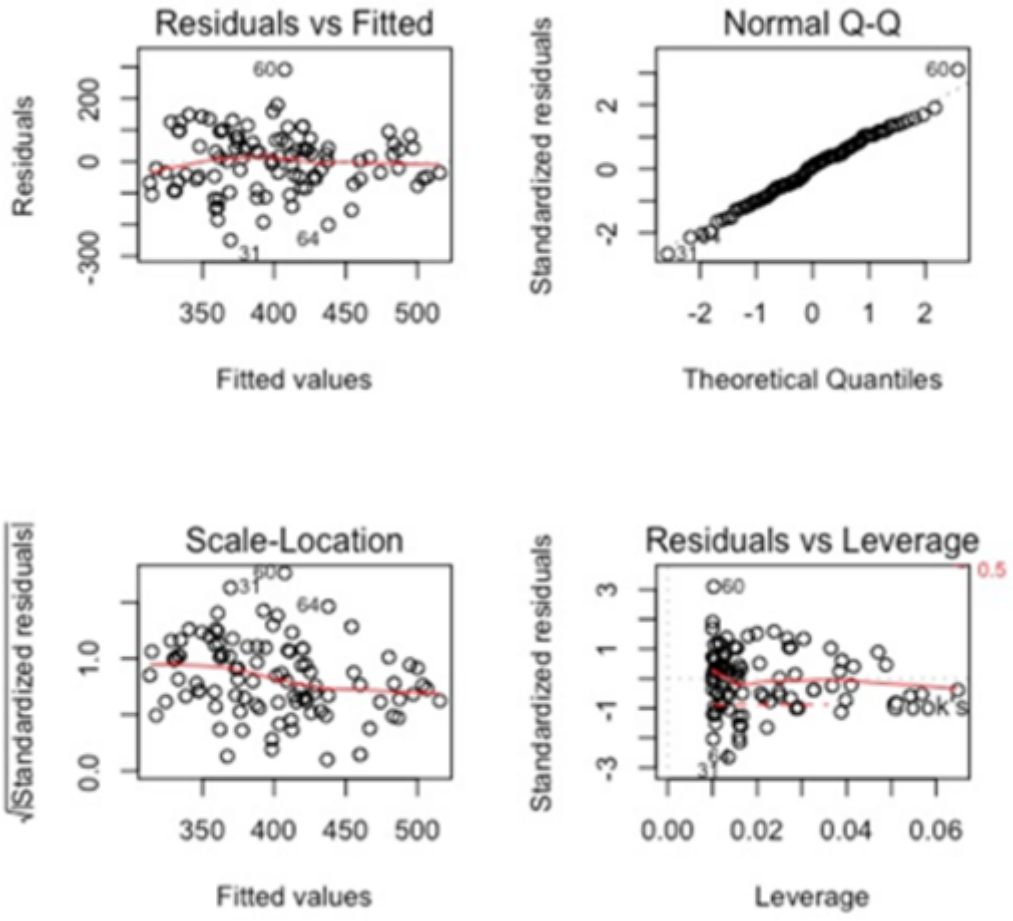
  · α and β are parameters of the regression, with α = intercept (average Y when X=0), and β = slope of the line (change in Y for a 1 unit increase in X). Note: α and β are unit specific, so you'll get different answers if you use distance in metres and in feet.

  · ε is the random variation in Y, i.e. the residuals

```
# fit a linear regression model
MWT1Best_FEV1 <- lm(MWT1Best~FEV1, data = COPD)
summary(MWT1Best_FEV1)

# view 95% confidence intervalas
confint(MWT1Best_FEV1)

# check model assumptions
predictedVals <- predict(MWT1Best_FEV1) # get predicted values for the model
residualVals <- residuals(MWT1Best_FEV1)# get residuals between model and data
par(mfrow=c(2,2)) # set plotting format: 2 by 2
plot(MWT1Best_FEV1) # see residual plots, QQplot
```



- check model assumptions by plot().

    - The first is a **constant variance plot**, which checks for the homogeneity of the variance and the linear relation. If you see no pattern in this graph, then your assumptions are met.
    - The second plot is a **Q-Q plot**, which checks that the residuals follow a normal distribution. The points should fall on a line if the normality assumption is met.
        - QQ plot is a plot of the quartiles of the residuals against the quartiles of a theoretical normal distribution and if the residuals are normal then the observations will lie on a straight line.
    - The third plot allows to detect heterogeneity of the variance.
    - The fourth plot allows for the detection of points that have a large impact on the regression coefficients.
- Assumptions of a linear regression model are that
    - there is linearity between the outcome and predictor variable;
    - that the outcome variable is normally distributed across values of the predictor;
    - that the variance of the outcome variable is constant across values of the predictor variable.
- If these assumptions are satisfied then the **residuals** (distance between the observed values and the fitted regression line) follow a normal distribution with mean zero and constant variance across the predictor values: Residuals ~ $N(0, \sigma^2)$

3. Fit a multiple regression model in R

1. Produce the simple linear regression model between MWT1best and FVC

```
# first simple model
lr1 <- lm(MWT1Best~FVC, data = COPD) # Run the regression, assigning the output
to a new variable lr
summary(lr1) # View the output of the regression
confint(lr1) # View the 95% confidence intervals of the regression

# second simplemodel
lr2 <- lm(MWT1Best~AGE, data = COPD)
summary(lr2)
confint(lr2)
```

MWT1best= α+β∗FVC, where α = 254.95 and β=48.63 and adjusted R2 = 0.19.

MWT1best= α+β∗AGE, where α = 616.45 and β=−3.10, and adjusted R2 = 0.04.

2. Multiple regression model

```
lr3 <- lm(MWT1Best~FVC+AGE, data = COPD)
summary(lr3)
confint(lr3)
```

MWT1best= α+β1∗FVC+ β2∗AGE, where α = 425.38, β1=46.06 and β2 = -2.33.

- Observe: Both **coefficients** have slightly reduced in the multiple regression model. The **p-value** for FVC remains small (<0.001) but the p-value for AGE has increased to 0.059 .
    - Reason: This change is because the coefficients are now adjusted coefficients: α is the estimated walking distance you would expect for people aged 0 years and with a FVC value of 0. β1 is the average increase in walking distance for every one unit increase in

FVC, keeping age held constant. β2 is the average increase in walking distance for every one year increase in age, keeping FVC held constant.

- Comparing the **adjusted-R2 statistics** you can see that the multivariable model now explains 21% of variance in the data compared with 19% for the model with FVC and 4% for the model with AGE,
    - Problem: the added AGE has p-value 0.059 which does not allow you to reject the null hypothesis that the coefficient is 0 at the usual 5% significance threshold.
        - -> you will cover model development and variable selection later in the course and you will see that this should not solely rely on significance testing.
- 95% CI become wide (which means standard error is larger), so it's misleading relationship between x_i and y. (there is no association)
- Your model doesn't seem to show any indication of **collinearity** (remember that this is when there is a strong linear relationship between predictors that causes problems in estimation of model parameters).
    - You can check this by examining the scatterplot or calculating a correlation coefficient such as Pearson's or Spearman's when there are only two predictors. However, this is not possible when the number of predictors increases beyond two
    - In that situation, you will need to produce a **correlation matrix** or calculate the **Variance Inflation Factor (VIF)**.
- Interpret 95% CI:

  This means that there is a 95% chance the true population parameter will lie somewhere in the range from XX to XX.

3. Check correlation. If there is only a weak association between them, the collinearity is not a problem. Then check assumptions of linear regression.

```
# Check Spearman's correlation of AGE and FVC
cor.test(COPD$AGE, COPD$FVC, use="complete.obs", method="spearman")
```

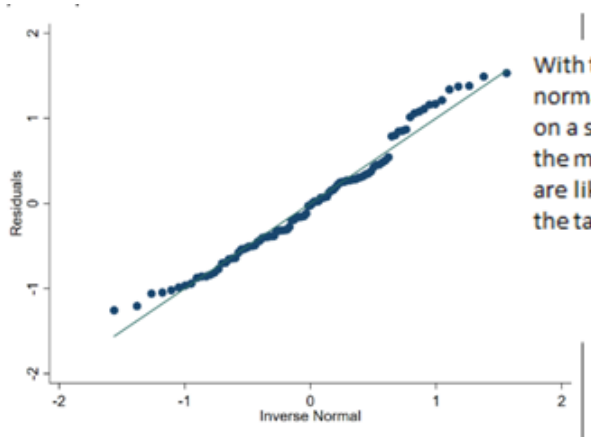
4. How well the model fit the data

- R-squared: how much variability is explained by the model as a proportion of total variability if data.
    - $R^2 = \frac{Variance \ \ explained \ \ by \ \ the \ \ model}{Total \ \ variance}$
    - R^2 is always between 0 and 1
    - If we have one predictor / variable. $R^2 = \rho^2$ (square of correlation coefficient)
    - e.g. $R^2 = 0.15$, regression model is explaining 15% of total variance.
    - e.g. adjusted $R^2 = 0.21$, the regression model explains 21% variability in y observation.
- Each time we add a variable to the model, R^2 will just increase.
    - **Overfitting** is a phenomenon where the model begins to describe the "random error" in the data rather than the relationships between variables
    - So R^2 is not a good measure when we compare models
    - **Adjusted** R^2 penalizes for the number of predictor variables included in the model. It is a better measure to compare models.
- Heteroscedasticity: is the unequal variance of one variable across another variable. Homoscedasticity: is equal variance of one variable across another variable.

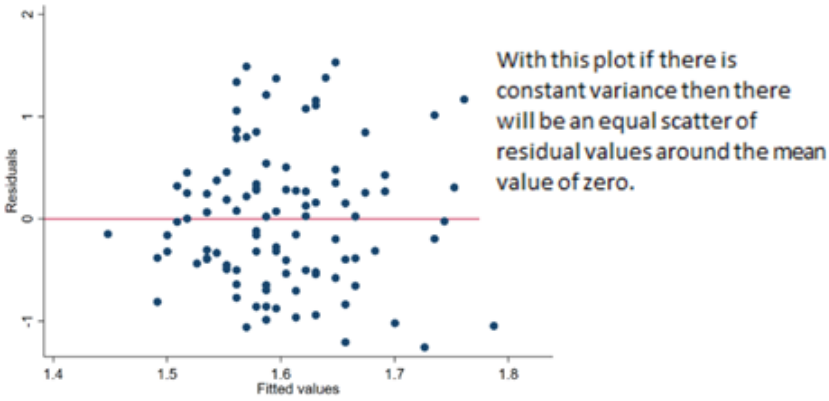
5. Summary

- Correlation

    - Similarity: both measure the strength of association between two variables. Both Pearson's and Spearman's take values between -1 and +1.
    - Differences you need to be aware of before you decide which, if any, is best for your data:
        - Pearson's uses the actual observed values to calculate the correlation, but Spearman's ranks the values and then calculates the coefficient using the ranks.
        - Pearson's measures the strength of a linear association between two variables, whereas Spearman's is not quite as restrictive and measures if there is an association assuming only a monotonic relationship.
        - Pearson's requires that both variables are continuous, whereas Spearman's can be used for continuous and ordinal variables
        - Pearson's requires that both variables are approximately normally distributed in the population, whereas Spearman's makes no such distributional assumptions. Both require that observations are a random sample from the population.
    - Always examine your data in a **scatterplot** before calculating a correlation.
- Linear regression
    - Correlation tells you if you have a strong, moderate or weak association between two variables, but it doesn't describe or quantify that relationship. If you want to quantify the impact of a change in one variable on another, you can use linear regression.
    - Linear regression is simply a way of fitting an optimal straight line to your data; allows you to evaluate relationships and make predictions.
    - One way to fit an optimal straight line ($y = \alpha + \beta x$) to the data is to minimise the sum of squared residuals, also referred to as the residual sum of squares (RSS).
    - Model in R
        - α is the intercept of the straight line which is the value of the outcome where the line cuts the y-axis i.e. when the predictor equals 0.
        - β is the slope or the gradient of the line and it quantifies the relationship between your two variables. It tells you how much on average your outcome variable increases for a one unit increase in your predictor.
    - For your model to be valid, there are 3 key assumptions that need to be satisfied (If these hold, then your residuals are normally distributed with a mean of 0 and a constant variance across the predictor values: **Residuals ~ Normal (0, σ2)**):
        - Linearity between outcome (Y) and predictor (X) variables;
        - The outcome variable is normally distributed across predictor values;
        - The variance of outcome is the same across predictor values.
    -
        - Linearity between outcome (Y) and predictor (X) variables;
        - The outcome variable is normally distributed across predictor values;
        - The variance of outcome is the same across predictor values.
    - Two useful plots:
        - QQ plot



- The scatterplot of the residuals by the fitted regression values



- Multivariable linear regression
  - Model
    - $y = \alpha + \beta1x1 + \beta2x2$
    - Remember that this slightly alters your interpretation of the α and β coefficients.
      - β1 is now the average increase in outcome for every one unit increase in x1 keeping x2 held constant.
      - β2 is the average increase in outcome for every one unit increase in x2 keeping x1 held constant.
  - You can investigate collinearity by examining the correlations among pairs of variables or through **variance inflation factors (VIFs)**.
    - Correlations are limited to only examining pairwise relationships. Correlations don't allow you to identify dependence amongst three or more predictors.
    - The VIF quantifies how much the variances of coefficients is inflated by, so each coefficient will have a VIF. VIFs equal to 1 indicate no collinearity amongst the predictors in the model, values above 4 would prompt further investigation, and anything above 10 would indicate serious issues of collinearity.

Week3 Multiple Regression and Interaction

1. The good practice steps to develop a multivariable linear regression model

1. Inspect your variables using summary statistics and histograms for continuous variables, and tabulations for categorical variables.

This helps you identify and quantify the amount of missing information. It also helps you spot any strange or outlying observations. If you do spot any strange values you will need to decide what to do with them before you move on.

2. Examine the relationship between your candidate predictor variables using cross tabulations for categorical variables and pairwise correlations and scatterplot matrices for continuous variables.

This helps you identify potential associations in your candidate predictor variables that could be problematic if included together in your multivariable model.

3. Fit a simple linear regression model between the outcome and each of your candidate predictor variables.

This allows you to assess the relationship between each of your candidate predictor variables and the outcome variable. Again, this helps to spot errors in either the data or the coding, and it also allows you to anticipate what you might expect to happen when you fit the multivariable model.

Once you've done this you're in a much better position to start thinking about building a multivariable regression model.

2. Practice in R: run a good analysis

1. Inspect the dataset for missing values and outliers

- To examine the datatype and distribution for all of these variables. This can be done using the describe() function from the 'Hmisc' package. This function allows you to examine the different variables, providing the number of values, the range of the values, the number of missing values, the mean, and the different quartiles of values in our variables.
- To examine each variable in more detail using summary statistics and tabulations. These allow you to spot missing data or outliers, which you might need to exclude for the next stages of your analysis.
    - For categorical variables, you can tabulate the data using the CrossTable() function from the 'gmodels' package, then use the sum(is.na()) functions to check for missing values.
    - the summary() command, which will allow you to look at the mean, median, minimum, maximum, 1st and 3rd quartiles, and the number of missing values (NAs)
    - You can use the plot to assess the distribution and identify any outliers. You'll need to make a decision whether this value is an impossible value, maybe due to a coding error, in which case the value may need to be excluded, or whether you think this is it just an unusual value that should be left.

2. Examine the relationship between your candidate predictor variables

- For continuous variables, we will be using pairwise correlations and scatterplot matrices, while we will be using cross tabulations for categorical variables. These will help you identify potential associations in your candidate predictor variables that could be problematic if included together in your multivariable model.
    - To calculate a correlation coefficient using the cor.test() command. To see the pairwise correlation coefficient only for continuous variables, you can use the cor() command.
    - There is an easy way to visually assess the correlation using the pairs() function.
    - To examine associations between categorical variables, you can use cross tabulations. Use the CrossTable() function from the 'gmodels' package.

3. Fit a simple linear regression model

- It's useful to assess the relationship for each of variable in turn with the outcome.
    - Doing this allows an opportunity to spot anything unusual that may due to errors in either the data or coding of the variable, and also allows you to anticipate what you might expect to happen when you fit the multivariable model.
    - It may be at this stage you discover a relationship that, subsequently, disappears in the multiple model, or there may be no relationship at this stage but that one is uncovered later.

3. Summary of data

- It's important that you get to know your data really well before you start modelling.

Examine	Categorical	Continuous
Inspect Variable	Tabulations	Summary statistics Histograms
Between Variables	Cross tabulations	Pairwise Correlations Scatterplot matrix
With the outcome	Lin. Reg. with 1 variable	Lin. Reg. with 1 variable

\1. The first thing is examining variable distributions using summary statistics, tabulations and graphs.

\2. The next stage is examining the relationship between candidate predictors using cross tabulations and correlations.

\3. And finally, get a feel for the relationships between each of the candidate predictors and the outcome by fitting a regression model for each variable in turn.

4. Effect of interaction between two binary predictor variables

MWT1best= α+ β1∗Diabetic+ β2∗AtrialFib+ β3∗Diabetic∗AtrialFib

where:

Diabetic = 0 if diabetes not present, 1 if present

AtrialFib = 0 if atrial fibrillation not present, 1 if present

Diabetic*AtrialFib = 0 if diabetes or atrial fibrillation not present, and 1 if diabetes and atrial fibrillation present.

- R will not be able to create this new variable if the Diabetes and AtrialFib variables are saved as factors! They therefore need to remain/be changed to integers. You can do this using the as.integer() function.

```
COPD$Diabetes <- c(0,1)[as.integer(COPD$Diabetes)]
COPD$AtrialFib <- c(0,1)[as.integer(COPD$AtrialFib)]

# create new variable
DAF <- COPD$Diabetes * COPD$AtrialFib

# fit the model
r2 <-
lm(MWT1Best~factor(Diabetes)+factor(AtrialFib)+factor(Diabetes*AtrialFib),
data=COPD)
summary(r2)
```

Result: MWT1best= 428.1−7.7∗Diabetic−72.0∗AtrialFib−130.1∗(Diabetic∗AtrialFib)1

- Obtain the predicted output directly in R using the prediction() command from the 'prediction' package
- p-value tells whether there is an evidence of an interaction effect between two variables.

  Notes:
- Changing the reference group of a binary variable does not change the regression coefficient: it only changes the sign of the regression coefficient.
- If your model contains a categorical variable, it's important that you check the default group before you interpret the results
- Before including any interactions, it's important to check the coding of a binary variable to make sure the groups are coded as zero and one
- Including an interaction completely changes your interpretation of the regression coefficients.

Week4 Model Building

1. Variable Selection

- Stepwise regression

  To identify he predictor with the greatest influence on the outcome as the criteria for being included into the model.
  - Forward Selection
    - Start with the null model
    - Add variables if they can make a significant improvement
  - Backward Selection
    - Start with the full model
    - Remove least significant variables
  - Stepwise
    - Start with the null model
    - Add variables if they make a significant improvement
- Limitation of stepwise method
  - They yield R-squared values that are badly biased to be high.
  - The F and chi-squared tests quoted next to each variable on the printout do not have the claimed distribution.
  - They yield confidence intervals for effects and predicted values that are falsely narrow; see Altman and Andersen (1989).
  - They yield p-values that do not have the proper meaning, and the proper correction for them is a difficult problem.
  - They give biased regression coefficients that need shrinkage (the coefficients for remaining variables are too large); see Tibshirani (1996).
  - They have severe problems in the presence of collinearity.
  - They are based on methods (e.g., F tests for nested models) that were intended to be used to test prespecified hypotheses.
  - Increasing the sample size does not help very much; see Derksen and Keselman (1992).
  - They allow us to not think about the problem.

- It uses a lot of paper. https://www.stata.com/support/faqs/statistics/stepwise-regression-problems/

(Discussion on CrossValidated https://stats.stackexchange.com/questions/20836/algorithms-for-automatic-model-selection

The technical details are covered in the notes from Frank Harrell's course on regression modelling strategies. Sections 4.3-4.7 are particularly helpful:

http://hbiostat.org/doc/rms.pdf)

2. Too many predictors

- **Overfitting** , which means the model just ends up explaining random error, rather than real relationships. It lacks generalizability.
- **Collinearity** is when there's a strong linear relationship between 2 or more predictors.
- In Harrell's book "Regression Modelling Strategies" he points out that a "regression model is likely to be reliable when the number of (candidate) predictors is less than n/10 or n/20 where n is the total sample size".
- For categorical variables, you need to consider the number of levels in the variable: if there are four levels then this requires three parameters in the model. The number of predictors also counts any interactions you want to include in your model.
- More on: Harrell's book is a great place to start when you are ready: Harrell, FE. Regression modelling strategies: with applications to linear models, logistic regression, and survival analysis, New York, NY: Springer, 2001.

3. Recourses

- There are lots of good and bad examples of model development in the literature. Some examples of how to report model development can be found below:

· Haroon S, Adab P, Riley RD, Fitzmaurice D, Jordan RE. Predicting risk of undiagnosed COPD: development and validation of the TargetCOPD score. European Respiratory Journal 2017; 49(6) 1602191; DOI: 10.1183/13993003.02191-2016. http://erj.ersjournals.com/content/49/6/1602191

· Martínez-Laguna D, Tebé C, Nogués X, Kassim Javaid M, Cooper C, et al. Fracture risk in type 2 diabetic patients: A clinical prediction tool based on a large population-based cohort. PLOS ONE; 2018 13(9): e0203533. https://doi.org/10.1371/journal.pone.0203533

· Selby A, Munro A, Grimshaw KE, et al. Prevalence estimates and risk factors for early childhood wheeze across Europe: the EuroPrevall birth cohort. Thorax 2018;73:1049-1061. https://thorax.bmj.com/content/73/11/1049

- If you are developing a prediction model to either assist in the prognosis or diagnosis of patients, then there is guideline for reporting your work that can be found here:

· http://annals.org/aim/fullarticle/2088549/transparent-reporting-multivariable-prediction-model-individual-prognosis-diagnosis-tripod-tripod#

· Collins GS, Reitsma JB, Altman DG, Moons KG. Transparent Reporting of a multivariable prediction model for Individual Prognosis Or Diagnosis (TRIPOD): The TRIPOD Statement. Ann Intern Med 2015;162:55–63. doi: 10.7326/M14-0697

- Examples of abstracts for multivariable regression models can be found here:

Vioque, J., Weinbrenner, T., Asensio, L., Castelló, A., Young, I., & Fletcher, A. (2007). **Plasma concentrations of carotenoids and vitamin C are better correlated with dietary intake in normal weight than overweight and obese elderly subjects.** British Journal of Nutrition, 97(5), 977-986. doi:10.1017/S0007114507659017. https://www.cambridge.org/core/journals/british-journal-of-nutrition/article/plasma-concentrations-of-carotenoids-and-vitamin-c-are-better-correlated-with-dietary-intake-in-normal-weight-than-overweight-and-obese-elderly-subjects/4296EF7DCA35C04A799A6ACE39015F74

Selby A, Munro A, Grimshaw KE, et al. **Prevalence estimates and risk factors for early childhood wheeze across Europe: the EuroPrevall birth cohort.** Thorax 2018;73:1049-1061. https://thorax.bmj.com/content/73/11/1049

4. Developing a model building strategy

1. Identify known predictors and their interactions
2. Thoroughly examine the data as there may be some variables not sensible or feasible to include
    - missing/ poor quality data
    - Narrow distributions
    - Collinearity

    (It maybe useful to combine several variables to make one variable)
3. Pre-specify interactions of interest
    - Think about what interactions you may want to examine
4. Data Reduction methods.

DO NOT

- Test each variable at the 5% significance level as a way to select variables to include in your model
    - as the relationship at the uni-variable level can appear or disappear at multi-variable level depending on confounding
- Use forward or stepwise selection procedure (possibly the backward selection if you have to)
- Only include known predictors if and only if they're statistically significant at 5% level.
    - If they're known predictors, they should be included in the model, regardless of p-value, they will alter the estimates of the other regression coefficients in the model.

5. Summary

New R script

· Remove previous variables: rm(list=ls())

· Set the working directory: setwd("file_pathway")

· Loading a dataset, labelling it 'COPD': COPD <- read.csv("COPD_student_dataset.csv")

Install a new package

· install.packages("Package name")

· library(Package name)

Viewing the dataset

· Look at the whole dataset: View(COPD)

· Print the first few rows of your dataset: head(COPD)

· See how many rows and columns you have in your dataset: dim(COPD)

· Look at the different variables in the dataset: colnames(COPD)

· Look at all the values in a variable: print(variable)

· To visualise the structure of the data in a variable: str(variable)

· Look at a specific value (x) in a variable: variable[x]

For continuous variables:

· View number of values, missing values, mean and ranges using the describe() function from the 'Hmisc' package.

For categorical variables:

· View number of values, missing values, mean and ranges using the describe() function from the 'Hmisc' package OR Tabulate the data to view the number of values and their frequency using the CrossTable() function from the 'gmodels package. To view missing values, type: sum(is.na(variable)).

· Viewing the categories and distribution of entries in a categorical variable: table(catvariable)

· You can add the argument exclude = NULL in the function parentheses to include missing values in the output.

Running a linear regression

The basic format is:

· modelname <- lm(outcome~predictor, data = dataframe)

· Viewing the regression model output: summary(modelname)

· Viewing the model 95% confidence intervals: confint(modelname)

Drawing a Q-Q plot, constant variance plot, and other diagnostic plots

· Calculate predicted values: predict(modelname)

· Calculate residuals: residuals(modelname)

· Set a plotting format of 4 graphs: par(mfrow=c(2,2))

· View the 4 resulting plots: plot(modelname)

Create a histogram

· The basic format is: hist(variablename)

· If you are getting a variable from the dataset, the $signallowsRtolocatethisvariable.E.g.COPD$MWT1Best

· To change the title of the histogram, use the command: main = "histogram title"

· Don't forget quotation marks when using text!

· To change the x or y axes labels, use the commands: xlab = "x axis label" or ylab = "y axis label"

· Don't forget quotation marks using text!

· To change the number of bins displayed, use the command main = to specify the number of bins you want to see.

· To look at specific values in your variable, you can use the subset() function, using the basic code subset(dataframe, variable > 15) if you want to see values over 15 for that variable. You can add additive rules by including '|', e.g. subset(dataframe, variable1 > 15 | variable2 < 5)

Summary statistics

· Basic summary statistics (incl. minimum, medium, maximum, 1st and 3rd quartiles, and number of blank cells): summary(variablename)

· List of summary statistics, including the basic summary() outcome, standard deviation, range, and inter-quartile range:

· list(summary(variablename), sd(variablename, na.rm = TRUE), range(variablename, na.rm = TRUE), IQR(variablename, na.rm = TRUE))

· Note that the na.rm = TRUE command tells R to remove NA values. Without this, an error message will be displayed.

Correlation

· Scatterplot of two variables: plot(x, y)

· Correlation coefficient: cor(x, y)

· The default method is Pearson, but you can change this to Spearman by adding method = "spearman" in your parentheses. You need to remove missing values, otherwise you will get an error message. To do this, add use = "complete.obs" in your parentheses.

· Correlation test: cor.test(x, y)

· The default method is also Pearson here. You also need to remove missing values to avoid an error message.

Creating a correlation matrix:

· Create a vector with the variables to include in the matrix, e.g. data <- COPD[,c("AGE", "PackHistory", "FEV1")]

· Create the correlation matrix vector, assigning correlation coefficients of the different variables to it, e.g. cor_matrix <- cor(data)

· View the matrix: e.g. cor_matrix to view the output and round(cor_matrix,2) to round this output to 2 decimal points.

· Visualising correlation between variables, i.e. correlation plot: pairs(~ variable1 + variable2 + variable3, data = dataframe)

Multiple linear regression

The basic format is:

· modelname <- lm(outcome~predictor1 + predictor2, data = dataframe)

· Viewing the regression model output: summary(modelname)

· Viewing the model 95% confidence intervals: confint(modelname)

· Examining the VIF using the imcdiagF() function from the 'mctest' package.

Regression with categorical variables

2 ways to do this:

· Check what the variable is saved as, change it to a factor variable if it is not saved as such.

· Check what the variable has been saved as using the class() function

· If it is not saved as a factor, can it using the factor() command, in the following format: variable <- factor(variable)

· Run the regression as normal

· Include factor() before the variable in the regression model. E.g. modelname <- lm(outcome~predictor1 + factor(predictor2), data = dataframe)

Changing the reference category of a variable:

· Use the relevel() function in the following format: variable <- relevel(variable, ref = newreflevel) with the newreflevel being the new reference level, written either as a numeric (1, 2, 3, …) or a character (in which case it needs to be written within apostrophes – "MILD", "SEVERE", …)

Changing data type for a variable

· Check what the variable has been as using the class() function.

· Changing data type:

· To numeric: as.numeric()

· To character: as.character()

· To factor: factor() or as.factor()

· To integer: as.integer()

Creating a new variable on R: e.g. variable 'comorbid'

comorbid is a variable you were asked to create. This variable was to be binary, and indicated the presence of at least one comorbidity ('1') or complete absence of comorbidities ('0') based on the responses to the variables: Diabetes, muscular, hypertension, AtrialFib, and IHD.

· Check that all variables are saved as the correct datatype.

· Create an empty vector of the correct length.

· Here, comorbid will be the same length as the other variables, so:

· comorbid <- length(COPD$Diabetes)

· Assign values to this vector.

· Here, we want comorbid = 1 when Diabetes OR muscular OR hypertension OR AtrialFib OR IHD = 1. So: comorbid[COPD$Diabetes == 1|COPD$muscular == 1 | COPD$hypertension == 1|COPD$AtrialFib == 1 | COPD$IHD == 1] <- 1

· This will assign 1 to the values meeting the set conditions, and NAs to those that are not meeting those conditions.

· We also want comorbid = 0 when ALL above variables = 0. So: comorbid[is.na(cormorbid)] <- 0

· Convert this variable to a factor.

· Optional: add the variable to the dataset, using the following command COPD$comorbid <- cormorbid

Regression with interaction effect

· Use the same format as a multiple linear regression, but include both terms, i.e.

· modelname <- lm(outcome~predictor1 + predictor2 + (predictor1 * predictor2),

· data = dataframe)

· Interpretation of the interaction effect can be simplified using the prediction() function from the ‘prediction’ package.