

Practical Workbook

Dr. Lee Fawcett and Dr. Joe Matthews,
Newcastle University

Lesson 1: Data Summaries

In this lesson we will be using the [Sales](#) data.

To import the data:

1. Download the [Sales.xlsx](#) file and save it to a folder on your computer
2. In RStudio, click Session -> Set Working Directory -> Choose Directory and select the folder where you downloaded the [Sales.xlsx](#) file

For simple spreadsheets, we can just use the [read.csv](#) command (see later practical sessions). However here, because the [Sales.xlsx](#) file has been formatted, we need to use a package in R to import it. Packages contain particular functions and datasets which we can load into R when we need them. A useful package for importing complicated/formatted spreadsheets is the [readxl](#) package.

```
1 # Install the package so we can use it later
2 # Note you only need to install a package once per machine
3 # (i.e. you don't need to run this every time)
4 install.packages("readxl")
5
6 # Load the package into R
7 # You need to do this each time you open RStudio and want to use the package
8 library(readxl)
```

Since the data file we want to import is type [.xlsx](#), we can use the [read.xlsx](#) function to import it. Because of the data quality, here we need to manually tell R the data types to expect so things are formatted correctly.

```
1 sales = read.xlsx("Sales.xlsx", col_types = c("numeric",
2                                               "text",
3                                               "numeric",
4                                               "numeric",
5                                               "text",
6                                               "text",
7                                               "date",
8                                               "date"))
9
10 # Take a look at the data
11 head(sales)
```

Problem Session 1

There are several ways to extract elements from a data frame in R. We can use square brackets [\[a,b\]](#) where [a](#) (the number before the comma) is the row number we want to access, and [b](#) (the number after the comma) is the column we want to access. If we leave one of them blank, R will output the entire row/column.

```

1 # Get the 2nd element of row 10
2 sales[10, 2]
3 # Get everything in row 4
4 sales[4, ]
5 # Get everything from column 3
6 sales[, 3]

```

We can obtain multiple rows/columns at once by including vectors inside the square brackets:

```

1 # Extract rows 3 and 5
2 sales[c(3, 5), ]
3 # Extract columns 2 and 4
4 sales[, c(2, 4)]

```

If our rows and columns are named, we can extract them by name instead of number (this is very helpful if we have lots of rows or columns and don't want to count to see the row/column number we need):

```

1 sales[, c("Rank", "Company")]

```

When we're dealing with data frames we can also access columns by name using \$:

```

1 sales$Company

```

Using square brackets we can also extract rows which meet a specified condition (note the ' marks that are required around a name with spaces – this indicates to R that everything inside is part of a single name):

```

1 # Extract products with a positive changes in sales since Q3 2014
2 sales[sales$'Change from Q4 2013' > 0, ]

```

Equivalently we can use the `subset` command:

```

1 subset(sales, 'Change from Q4 2013' > 0)
2 subset(sales, 'Change from Q4 2013' < 0)

```

If we wish to subset based on an element is exactly equal to something we use a double equals ==

```

1 subset(sales, 'Change from Q4 2013' == 0)

```

We can subset by character strings (i.e. elements in our data which are words rather than numbers) in the same way:

```

1 subset(sales, Company == "AstraZeneca")

```

If we want to include partial matches (i.e. where a sequence of letters or words is included in an element, but may not be the full thing) we can use the `grep` command. `grep` takes two main inputs: the text we're looking to find, and the vector to look in.

```
1 # Only entries where the medical use is listed as HIV exactly
2 subset(sales, 'Medical use' == "HIV")
3 # Entries where the medical use includes HIV (and possibly other text)
4 sales[grep("HIV", sales$'Medical use'), ]
```

By default `grep` is case sensitive, and so will only return a row if the exact combination of upper and lower case letters is matched. If we would like `grep` to ignore the letter case, we can specify `ignore.case = T`.

```
1 sales[grep("Depression", sales$'Medical use'), ]
2 sales[grep("Depression", sales$'Medical use', ignore.case = T), ]
```

We can combine multiple statements in our subsetting, using the `&` symbol if we want entries where both our statements are satisfied, and `|` when we just need one or the other to be satisfied:

```
1 # Products approved before 2000 with a rank in the top 10
2 sales[sales$'First approval date' < "2000-01-01" & sales$Rank <= 10, ]
3 # Products produced by Pfizer or AstraZeneca
4 sales[sales$Company == "Pfizer" | sales$Company == "AstraZeneca", ]
```

We can see how many rows satisfy a condition using the `sum` command, or what proportion of rows satisfy the condition using the `mean` command:

```
1 sum(sales$'Sales Q1 2014 ('000 USD)' < 500000)
2 mean(sales$'Sales Q1 2014 ('000 USD)' < 500000)
```

Questions: Querying dataframes

1. How many Pfizer products are ranked in the top 50?
2. When was the product with brand name Herceptin first approved?
3. How many products in the data treat diabetes?
4. How much in sales in Q1 2014 was made by the top performing product made by Bayer?

Problem Session 2

Useful commands for producing numerical summaries of data:

```
1 x = sales$`Change from Q4 2013`  
2 # The mean of a sample  
3 mean(x)  
4 # The standard deviation of a sample  
5 sd(x)  
6 # The interquartile range  
7 IQR(x)  
8 # The largest and smallest value  
9 max(x)  
10 min(x)  
11 # A quantile of a sample, e.g. the 20% quantile  
12 quantile(x, 0.2)
```

Questions: Data summaries

1. What is the mean amount in sales in Q1 2014 made by the top 10 performing products?
2. Which product has had the worst (most negative) change in sales since Q4 2013?
3. What is the range of changes from Q4 2013 for products that treat depression?
4. On average who made more in sales in Q4 2014: GlaxoSmithKline or Johnson & Johnson?

Problem Session 3

Questions: Graphical summaries

1. Produce a histogram and boxplot of the sales in Q1 2014. Based on these plots, which measures of average and spread should we use to summarise this variable?
2. Produce a scatter plot of sales in Q1 2014 against change from Q4 2013. Based on this plot, does there appear to be a relationship between how well the products performed in Q1 and how they performed in the previous quarter? Calculate the correlation – does this agree with your conclusions from the scatter plot?
3. Create a new data frame just containing data for arthritis treatments. Produce a box plot of their sales in Q1 2014. What can we say about this data?
4. Produce histograms of sales in Q1 2014 for treatments first approved before 1st January 2000, and those first produced after. How do the sales figures compare?

Lesson 2: Exploratory Multivariate Data Analysis

Problem Session 1

Question: A hidden outlier?

Suppose that you're the production supervisor at a company making the next big blockbuster drug. The fermentation process used to make the active ingredient has been running smoothly until, suddenly, alarms are sounding, and you need to decide what to do!

Your production relies on observations on two main variables – temperature (°C) and acidity (pH) – both of which are recorded once per hour.

Load the `experiment` data into RStudio. The data on acidity are stored in the first column; data on temperature are stored in the second column.

- (a) Produce time series plots for both acidity and temperature. For example:

```
1 plot(ts(experiment[,1]), main="Plot of acidity values", xlab="Time", ylab="pH")
```

Can you detect anything unusual from your time series plots?

- (b) Now produce a scatterplot of acidity against temperature, and comment.

Problem Session 2

Question: Scatterplot matrices

Let's consider the construction of simple scatterplot matrices in R.

- (a) First, import the `heptathlon` dataset that we studied during the last session, and take a look around the dataset:

```
1 #Look at the head of the dataset:
2 head(heptathlon)
3
4 #Extract performances for the 100 metres hurdles:
5 heptathlon$H100M
```

- (b) Use R's `apply` command to produce histograms of performances on all 7 variables. Use `par(mfrow=c(2,4))` to partition your plotting window!
- (c) Use the `pairs` command to re-produce the scatterplot matrix from the lecture.
- (i) What is the nature of the association between performances in the 100 metres hurdles and the 800 metres? What about the relationship between performances in the 100 metres hurdles and the long jump?
 - (ii) How are performances in the shot put and the javelin throw associated?

Question: Correlation matrices

Let's consider the construction of correlation matrices in R. As demonstrated in the last session, execute the following code in R:

```
1 cor(heptathlon)
```

- (a) Can you explain how the correlation matrix supports your answers in part (c) of the last question?
- (b) Why do all the cells down the diagonal of the matrix contain 1.00000?
- (c) Which pair of variables seem to have the strongest association? What is the *direction* of this association?
- (d) For which pairs of variables might you say there is *no* association? Why should you be careful here?

Problem Session 3

Question: Principal Components Analysis

The primary focus of the lecture was on the use of Principal Components Analysis (PCA) as a way of exploring multivariate datasets. Recall from the lecture that, for easier interpretation of the principal components, we should make sure the direction of each variable has the same interpretation. In the lecture, we re-scaled data on the running events to make high values correspond to better performances. The following code reads the heptathlon dataset into R (you would have already used this in an earlier question), and then re-scales the running variables so that – for all variables – larger values correspond to better performances:

```
1 #First, copy the original dataset into a new dataframe:
2 > heptathlon2 = heptathlon
3
4 #Now re-scale, as we saw in the lecture:
5 > heptathlon2$H100M = max(heptathlon2$H100M) - heptathlon2$H100M
6 > heptathlon2$R200M = max(heptathlon2$R200M) - heptathlon2$R200M
7 > heptathlon2$R800M = max(heptathlon2$R800M) - heptathlon2$R800M
```

Now try the following code to re-produce the analysis from the lecture – making sure you understand the output! Compare to the slides used in the lecture if necessary.

```
1 > (pca = prcomp(x=heptathlon2, scale=TRUE))
```

In the final part of the lecture, we will produce a *score plot*. Use the following code to produce the score plot for the heptathlon dataset, then return to the lecture to find out what it shows and what it tells us!

```
1 > plot(pca$x[,1], pca$x[,2], xlab="First PC", ylab="Second PC")
2 > text(pca$x[,1], pca$x[,2], labels=rownames(heptathlon2))
```

Question: Principal Components Analysis

This is an additional question for you to try on Principal Components Analysis. Feel free to skip this for now and just return to the lecture.

Load the [MVSolvent](#) data into R.

You should see properties for a number of solvents that feature in the solvent selection tool used by chemical development scientists working on the design of manufacturing processes for active pharmaceutical ingredients. Perform a PCA on this dataset, and use a score plot to detect any clustering.

Lesson 3: Clustering

In this practical we will be using the `Reaction` data. We can take a look at the data using the `head` command:

```
1 head(reaction)
```

We see that the first column contains the reaction names. This will cause problems for us later when we try to calculate distances between the rows. Let's remove this column of reaction names, and set them to be the row names instead (so we can still see which reaction is which, but R won't treat the names as part of the data when calculating distances):

```
1 # Create a data frame without the reaction names
2 reac = reaction[,-1]
3 # Set the reaction names as row names
4 row.names(reac) = reaction[,1]
```

We can see the effect this has had using the `head` command again:

```
1 head(reac)
```

Problem Session 1

Question: Distance metrics

1. Obtain the distance matrix for the `Reaction` data using Euclidean distance.
2. Convert it to matrix form using the `as.matrix` command.
3. Produce a plot of the Euclidean distances from reaction 10 "Bromo N-alkylation". Which reaction has the shortest distance from reaction 10? Which reaction has the largest distance from reaction 10?
4. Repeat the above procedure to obtain a plot of the Manhattan distances from reaction 10. How has the change in distance metric affected our results?

Problem Session 2

Question: *k*-means clustering

1. Carry out a *k*-means clustering of the `Reaction` data with $k = 6$ clusters.
2. Produce a plot of the first and second principal components of the `Reaction` data, with each point coloured according to the cluster they were assigned in part 1. Does this seem like a sensible clustering? Why/why not?
3. Produce a plot of SSW against k for values of k from 1 to 10. Based on this plot, which value(s) of k would be a sensible choice?

4. Repeat part 2 for your chosen value of k . Does this look like a better cluster partition?

Problem Session 3

Question: Hierarchical clustering

1. Carry out a hierarchical clustering of the [Reaction](#) data with Euclidean distance and complete linkage. Plot the dendrogram. What does this suggest to be a sensible number of clusters?
2. Cut the dendrogram to produce your chosen number of clusters and visualise the cluster partition on a principal component plot. Does it seem to be a sensible clustering?
3. Repeat Questions 1 and 2 with Manhattan distance instead of Euclidean. How does this affect the results?
4. Repeat the clustering with Euclidean distance and single and then average linkage. How does this affect the results? Which linkage method is best for this data?

Lesson 4: Probability and Probability Models

Problem Session 1

Question: Introductory ideas

Consider a spinner with the following quadrants:



- (a) Suppose X : Colour landed on after spin. Complete the probability mass function below.

x	Yellow	Blue	Green	Red
$P(X = x)$	1/4			

- (b) Explain why the table of results above represents a valid probability mass function.
- (c) Find:
- (i) $P(X = \text{Blue or } X = \text{Green})$
 - (ii) $P(X = \text{Yellow or } X = \text{Red})$
- (d) The spinner is spun twice. What is the probability that we see both a Blue and a Yellow?

Problem Session 2

Question: Binomial and Poisson probabilities

Let's consider the construction of simple probability distributions in R. We'll start by re-visiting the two examples from the lecture:

1. The number of tropical storms observed per year
2. The number of rats developing tumours after exposure to *Ochratoxin A*

Recall that in example 1 the number of tropical storms observed per year was assumed Poisson with rate $\lambda = 1.92$, where λ was estimated from the mean of the sample. Try out the following code, making sure you follow the comments provided.

```

1 #Set up a vector for X, the number of tropical storms:
2 > (X = seq(0, 7, 1))
3 [1] 0 1 2 3 4 5 6 7
4
5 #Now apply the dpois command to X to find the PMF:
6 > dpois(X, 1.92)
7 [1] 0.146606962 0.281485367 0.270225953 0.172944610 0.083013413 0.031877150
8 [7] 0.010200688 0.002797903
9
10 #And the ppois command to find the CDF:
11 > ppois(X, 1.92)
12 [1] 0.1466070 0.4280923 0.6983183 0.8712629 0.9542763 0.9861535 0.9963541
13 [8] 0.9991520

```

(a) Using the R output, find:

- (i) $P(\text{Exactly 3 tropical storms})$
- (ii) $P(\text{Exactly 6 tropical storms})$
- (iii) $P(\text{At most 4 tropical storms})$
- (iv) $P(\text{Fewer than 4 tropical storms})$
- (v) $P(\text{At least 4 tropical storms})$

Recall that in example 2 the number of rats developing tumours was assumed Binomial with $n = 5$ and $p = 0.35$.

- (b) Use the code above for example 1 to obtain the PMF and CDF for the number of rats developing tumours, as shown during the lecture. *Hint: Try R's built-in commands `dbinom(X, n, p)` and `pbinom(X, n, p)`, to replace the commands used for the Poisson distribution above.*
- (c) Suppose we will roll a fair, six-sided die, 4 times, and we are interested in X : The number of sixes we will roll. Notice this could be modelled as a Binomial distribution:

- We have $n = 4$ fixed trials
- Each trial has two possible outcomes – “success” (get a 6) or “failure” (don’t get a 6)
- We should be able to assume independence between the trials – for example, getting a six on one trial has no bearing over whether we get a six in any other trial
- The probability of success is $p = 1/6$
- Therefore, we can say $X \sim \text{Bin}(4, 1/6)$

Use R in the same way you did in the last example to find the full probability mass function for X . By hand, calculate the expected value and the variance of X .

Question: More Binomial and Poisson probabilities

- (a) 20% of tablets from a batch of film-coated tablets are estimated not to fully dissolve after the allotted time period. For a sample of 10 randomly chosen tablets from the batch, use R to find the full probability distribution for X : Number of tablets that will not fully dissolve after the prescribed time.
- (b) Levels of a genotoxic impurity are known to increase when tablets of Theojex are stored after their manufacture. For a sample of 150 tablets, the average level of the genotoxic impurity is 15 ppm relative to the active ingredient. Use R to find the probability that a randomly selected tablet has exactly 10, 15, 20 and 30 ppm of the impurity relative to the active ingredient.

Problem Session 3

Question: The Normal distribution

Load the female `haematocrit` measurements, as used in the lecture, into R.

- (a) Use appropriate graphical and numerical techniques to summarise the dataset, and to help to confirm an assumption of Normality – with $\mu = 42.15$ and $\sigma = 3.44$.
- (b) Use the following R code to find the probability that a randomly chosen female student will have a haematocrit value of less than 45%:

```
1 #Recall, the mean of our sample is 42.15 and the standard deviation is 3.44:
2 > pnorm(45, 42.15, 3.44)
3 [1] 0.796303
```

- (c) Now, using similar code, find the probability that a randomly chosen female student will have a haematocrit value (i) less than 48%; (ii) less than 53%.
- (d) What is the probability of a randomly chosen female student having a haematocrit measurement of *more than* 45%?
- (e) Suppose we are told that male haematocrit values are Normally distributed, with $\mu = 45$ and $\sigma = 4$. To produce synthetic data for 126 male students, type the following code (note the switch from the `p` prefix to the `r` prefix – this switches from probability generation to random generation!):

```
1 pcv_male = rnorm(126, 45, 4)
```

Explore the simulated dataset by producing a histogram. Also find the mean and standard deviation of your simulated sample. What is the probability that a male student will have a haematocrit value less than 48%? How does this compare to that for a female student?

Question: Normal Q-Q plots

- (a) Sticking with the female Haematocrit values, let's produce a Q-Q plot to formally assess Normality in the data:

```
1 qqnorm(pcv)
2 qqline(pcv)
```

How would you interpret this plot?

- (b) The *Exponential distribution* is a probability distribution for continuous data that are left-skewed (rather than symmetric, as with Normally-distributed data). This distribution has a single rate parameter λ controlling the extent of asymmetry in the distribution. Let's look at what a normal Q-Q plot looks like for exponentially-distributed data!

```
1 #Simulate a dataset of 500 values from an exponential distribution with rate 2:
2 X = rexp(1000, 2)
3 #Now produce a Normal Q-Q plot:
4 qqnorm(X)
5 qqline(X)
```

Lesson 5: Inference

Problem Session 1

Here we will perform some basic statistical inference on the `Process` data. This can be imported into R using the `read.csv` command. We can attach this data and get a glimpse of the variables using the `attach` and `head` command respectively:

```
1 attach(process)
2 head(process)
```

Question: Confidence intervals

1. Produce a box plot of the amount of impurity A, split by the process variant. What initial comments can we make about the data?
2. Produce a 95% confidence interval for the amount of impurity A obtained from process variant C.
3. Produce a 99% confidence interval for the amount of impurity A obtained from process variant C. How does this differ from the interval obtained in the last part? How can we explain/interpret this?
4. Produce a 99% confidence interval for the amount of impurity A obtained from process variant A. Compare this with the 99% interval from process variant C – how can we interpret this?

Problem Session 2

Question: One-sample hypothesis test

1. Carry out a one sample t -test on the `assay.after.distillation.wt` to test the following hypotheses:

$$H_0 : \mu = 50$$

$$H_1 : \mu \neq 50$$

What is the p -value and is it significant? What can we conclude about our investigation?

2. Repeat this test but this time using a one tailed alternative

$$H_0 : \mu = 50$$

$$H_1 : \mu > 50$$

How has the p -value changed? Can we explain this?

3. Are the assumptions of this t -test valid?
4. Suppose we were interested solely in processes taking place in China in 2017. Extract the corresponding `assay.after.distillation.wt` measurements, and perform an appropriate test of

$$H_0 : \mu = 50$$

$$H_1 : \mu \neq 50$$

Problem Session 3

We wish to investigate whether the amount of impurity A produced by a process significantly differs based on the process variant.

Question: Two-sample hypothesis test

1. Produce a box plot of the amount of impurity A, split by the process variant. (This is a repeat of the first question from problem session 1)
2. Carry out an appropriate two-sample t -test comparing the amount of impurity A from process variant A, and from process variant C (Note: The easiest way to do this may be to create a new data frame containing only the variant A and variant C samples). What are our hypotheses? What is the p -value? What can we conclude from this test?
3. What would be the effect on the p -value if we adopted the one-tailed alternative

$$H_0 : \mu_A = \mu_C$$

$$H_1 : \mu_A > \mu_C$$

What about if we used

$$H_0 : \mu_A = \mu_C$$

$$H_1 : \mu_A < \mu_C$$

4. What happens to the p -value if we use a non-parametric test instead of a t -test? Repeat the two-tailed example as a non-parametric test to confirm your answer (Note: if you add `$p.value` to the end of a test command, R will print out the p -value in full) i.e.

```
1 # R will print out the p-value in full
2 t.test(x, y)$p.value
```

Lesson 6: Analysis of Variance

Problem Session 1

Question: One-way ANOVA

The following data are from an experiment to compare the effects of four catalysts on the yield of a chemical process:

Catalyst	Observations				
A	4.85	4.89	5.60	5.08	3.42
B	9.43	6.77	5.39	7.39	
C	4.67	5.49	5.96	3.71	5.51
D	5.50	4.90	6.00	5.81	4.68

Type the data manually into R in the following way:

```
1 yield = c(4.85, 4.89, 5.60, 5.08, ..., 4.68)
2 catalyst = c(rep("A", 5), rep("B", 4), rep("C", 5), rep("D", 5))
```

- Using any techniques you have learnt in this course so far, check the assumption that the four samples are Normally distributed.
- Also, informally check the underpinning assumption of equal variances in the four populations from which these samples have been drawn.
- Write down your null hypothesis.
- Now perform a one-way Analysis of Variance using the following code, and interpret the output.

```
1 > fit = aov(yield ~ catalyst)
2 > summary(fit)
3 > (tukeytest = TukeyHSD(fit))
4 > plot(tukeytest)
```

Question: Randomised blocks ANOVA

Recall the suncream example from the lecture:

In an experiment to determine the relative effectiveness of four brands of sun cream (Miksa *et al.*, 2016), six volunteers each tried four creams (A, B, C and D) on various parts of their legs. They were then subjected to ultra-violet radiation and an estimate of the degree of burning was made (higher figures indicate greater burning).

The results are given below, along with some summaries:

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Total	mean	s.d.
A	8.6	8.8	7.9	6.4	6.8	8.1	46.6	7.77	0.969
B	5.9	6.2	6.6	6.6	7.4	6.3	39.0	6.50	0.514
C	6.7	9.0	8.4	8.6	6.9	7.2	46.8	7.80	0.982
D	8.2	7.4	8.4	6.9	7.5	7.5	45.9	7.65	0.554

Read the [suncream](#) data used in the lecture into R.

- (a) Repeat the analysis from the lecture, checking for differences in the performances of the four suncreams.
- (b) Add an extra term into the model to check the significance of a blocking factor (the 'subject' could be a blocking factor here), and perform the randomised blocks ANOVA in R. What do you conclude?

Problem Session 2

Question: Factorial experiments

Run the following code in R to re-produce the analysis on the [rats](#) data we have just examined in the lecture. The R code is available separately, so you can just copy-and-paste this into your R terminal. Make sure you understand what's going on each step of the way!

```
1 #Make sure R knows that the source, and level, of protein are both factors:
2 > rats$Source = as.factor(rats$Source)
3 > rats$Level = as.factor(rats$Level)
4
5 #Produce some summary plots – checking for Normality along the way!
6 > par(mfrow=c(1,2))
7 #Boxplots for the data by source of protein:
8 > boxplot(rats$Gains~rats$Source)
9 #And now by level of protein:
10 > boxplot(rats$Gains~rats$Level)
11
12 #How about Normal Q-Q plots from last week? Easy!
13 > par(mfrow=c(2,3))
14 > qqnorm(rats$Gains[rats$Source==1], main="Source = Beef")
15 > qqline(rats$Gains[rats$Source==1])
16 > qqnorm(rats$Gains[rats$Source==2], main="Source = Cereal")
17 > qqline(rats$Gains[rats$Source==2])
18 > qqnorm(rats$Gains[rats$Source==3], main="Source = Pork")
19 > qqline(rats$Gains[rats$Source==3])
20
21 > qqnorm(rats$Gains[rats$Level==1], main="Level = High")
22 > qqline(rats$Gains[rats$Level==1])
23 > qqnorm(rats$Gains[rats$Level==2], main="Level = Low")
24 > qqline(rats$Gains[rats$Level==2])
25
26 #Let's now produce some numerical summaries – we can use any function we like, but let's
   check the assumption of equal variances:
27 > aggregate(rats$Gains, list(rats$Source), FUN = sd)
28 > aggregate(rats$Gains, list(rats$Level), FUN = sd)
29
30 #Now let's do the ANOVA!
31 > fit = aov(rats$Gains~rats$Source+rats$Level+rats$Source*rats$Level)
32 > summary(fit)
```

```

1 #And finally , an interaction plot:
2 > par(mfrow=c(1,1))
3 > interaction.plot(x.factor = rats$Source, #x-axis variable
4                   trace.factor = rats$Level, #variable for lines
5                   response = rats$Gains, #y-axis variable
6                   fun = mean, #metric to plot
7                   ylab = "Weight gain (g)",
8                   xlab = "Source of protein",
9                   col = c("pink", "blue"),
10                  lty = 1, #line type
11                  lwd = 2, #line width
12                  trace.label = "Level of protein")

```

Question: Factorial experiments

Read the [RRStudy](#) data into R.

Note there are several response variables:

- Impurity RRT
- Impurity Rs
- Product RRT
- Product Rs
- Product T

And several potential 'Input variables':

- Column (two levels: A/B)
- Instrument (two levels: X/Y)
- mMNH4OAC (three levels: 9.6/12/14.4)
- Flowrate (mL/min; three levels: 0.5/0.6/0.7)
- Temperature (°C; three levels: 45/50/55)
- Wavelength (nm; three levels: 241/244/247)
- Gradient time (min; three levels: 13.5/15/16.5)

The data relates to an analytical method that uses liquid chromatography to separate a pharmaceutically-related material ("Product") and a contaminant ("Impurity"). Important characteristics of the method are the responses shown: the resolution of the Impurity and Product peaks (Impurity Rs and Product Rs), the retention time of the

contaminant relative to the Product (Impurity RRT) and the amount of tailing on the Product peak (Product T). (Product RRT is not important as it is by definition unity).

The reliability of the analytical method for routine use has been assessed through a robustness and ruggedness study. Robustness has been tested by assessing the impact on the responses when the continuous input variables are deliberately changed by small amounts. The ruggedness is tested by assessing the impact on the responses when the column and instrument are changed to items that are nominally equivalent (e.g. a different column but of the same type and part number as the original). The robustness and ruggedness study does not vary the input variables one at a time. They are systematically changed as part of an experimental design.

- (a) Use ANOVA techniques from the lecture to assess whether or not average **Impurity RRT** measurements differ across the different levels of each input variable.
[Hint: You can use exactly the same code as we used for the rats experiment, but the formula used inside the `av` function will include all input variables together. And don't forget to make sure that R knows your input variables are factors!]
 - (i) From your ANOVA, you should find that only two input variables are significant. For each of these, use the *Tukey Highest Significant Difference* multiple comparisons test to identify where the differences lie.
 - (ii) Check for an interaction between the two variables that are significant.
 - (iii) Use normal Q-Q plots to assess Normality in the Impurity RRT data, across different levels of the two significant variables. Also informally check the assumption of equal variances.
- (b) Repeat part (a), in turn, for each of the other response variables.

Lesson 7: Linear Regression

Here we will carry out a regression analysis on the `Process` data. This can be imported into R using the `read.csv` command. We can attach this data and get a glimpse of the variables using the `attach` and `head` command respectively:

```
1 attach(process)
2 head(process)
```

Problem Session 1

To fit a regression model in R, we use the `lm()` command, for example to fit a regression model on `impurity.B..` with `impurity.A..` as the predictor, we would run:

```
1 m = lm(impurity.B.. ~ impurity.A..)
```

We can then see the model results using the `summary` command:

```
1 summary(m)
```

We can then visualise the resulting model with the `abline` command:

```
1 plot(impurity.A.., impurity.B..)
2 abline(m)
```

Question: Simple linear regression

1. Produce a scatter plot of `assay.after.distillation.wt` against `impurity.C..`. From this plot what can we say about the relationship between the two variables?
2. Fit a linear regression model with `assay.after.distillation.wt` as the dependant/response variable, and `impurity.C..` as the independent/predictor variable. What is the fitted regression equation? Does this support our initial impressions from the scatter plot? Add the fitted regression line to the plot.
3. Predict the `assay.after.distillation.wt` for a process resulting in 0.25 impurity C. What is a potential issue with making this prediction from our data?

Problem Session 2

Question: Simple linear regression

1. Fit a simple linear regression model on `assay.after.distillation.wt` which uses `impurity.A..` as the predictor, and a model which uses `impurity.B..` as the predictor.
2. Using the output of these 3 regression models (including the `impurity.C..` model from the first problem session), which impurity measure is the best at modelling `assay.after.distillation.wt`?
3. For your chosen best model, check to ensure the regression assumptions are satisfied and if there are any high leverage points in the data.

Problem Session 3

To fit a linear regression with multiple predictors, we just use a `+` symbol to separate the predictors in the `lm` function. E.g. if we wanted to use `impurity.A..` and `impurity.C..` to predict `impurity.B..` then we would run:

```
1 model = lm(impurity.B.. ~ impurity.A.. + impurity.C..)
```

To access the `vif` function (used to check for multicollinearity), we first need to (install and) load the `car` package.

Question: Multiple linear regression

1. Fit a multiple linear regression model on `assay.after.distillation.wt` using all other variables in the process dataset as predictors.
2. Carry out backwards stepwise elimination to remove unnecessary predictors from the model.
3. Verify the regression assumptions are satisfied for your final chosen model (including checks for multicollinearity).
4. We receive information on an extra reaction:
 - Year of Manufacture: 2018
 - Process Variant: Variant A
 - Manufacturing Location: India
 - Impurity A: 0.3
 - Impurity B: 0.25
 - Impurity C: 0.15
 - Solution Colour After Distillation: Off White

The `assay.after.distillation.wt` for this reaction was 60. Using a 95% prediction interval, comment on whether this suggests something unexpected has happened with this reaction.

Lesson 8: Further Topics in Regression

Problem Session 1

Question: Quadratic regression

The effects of reaction temperature ($X_1^\circ\text{C}$) and mixing rate (X_2 revolutions per minute) on the yield of biodiesel (Y wt. %) was investigated, resulting in the datafile [bio](#). Read this file into R.

- (a) Investigate the dependence between Y and each of X_1 and X_2 using scatter-plot and correlation matrices. Also use these diagnostics to assess any multi-collinearity.
- (b) Now try fitting the following multiple regression model in R:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \epsilon, \quad \epsilon \sim N(0, \sigma^2).$$

The following code might help:

```
1 # Attach the column headings for convenient access to the data!
2 > attach(bio)
3
4 #Create new vectors for a potential quadratic regression on both predictors:
5 > Tempsq = Temperature^2
6 > Mixsq = MixRate^2
7
8 #Now fit the full model and examine the output:
9 > full.model = lm(Yield~Temperature+Tempsq+MixRate+Mixsq)
10 > summary(full.model)
```

- (c) Proceed with a backwards elimination procedure from the full model in part (b) until you have only significant predictors remaining. Store your fitted model output in `final.model`. You should find that you're left with the following fitted model:

$$Y = -108.092 + 7.001X_1 - 0.062X_1^2 + \epsilon, \quad \epsilon \sim N(0, \sigma^2),$$

and with an adjusted R^2 of 70.9%.

- (d) What is the expected yield when the temperature is 58°C ?
- (e) Now try the following code, and make sure you understand what is going on:

```
1 > par(mfrow=c(2,3))
2 > plot(Temperature, Yield, pch=19, xlab="Temperature", ylab="Yield", main="")
3 > curve(-108.092+7.001*x-0.062*x^2, from=min(Temperature), to=max(Temperature), lty=2,
4 > add=TRUE)
4 > plot(final.model)
```

Problem Session 2

Question: Binary logistic regression

Tabletability is the capacity of a powdered material to be transformed into a tablet of specified strength under the effect of compaction pressure. Since pharmaceutical tablets usually contain multiple ingredients, the ability to predict tabletability of a powder mixture from individual components is of both fundamental and practical importance.

For a sample of 150 tablets, the pressure used to produce the tablet (X , the “weight fraction”, measured on a scale of $0 \rightarrow 1$) was recorded against the binary variable:

$$Y = \begin{cases} 1 & \text{If the tablet's tensile strength is ideal} \\ 0 & \text{If the tablet's tensile strength is not ideal.} \end{cases}$$

You can find the data in the file `tablet.tensile`, which you should read into R.

Follow the material from the lecture to perform a binary logistic regression on the data. The following code will help:

```
1 > attach(tablet.tensile)
2 > fit = glm(ideal ~ weight.fraction, family = binomial)
3 > summary(fit)
```

Make sure you can:

- (i) Check the significance of the predictor (independent) variable
- (ii) Write down your logistic regression equation, and use it to predict whether or not the following tablets would have an ideal tensile strength:

Weight fraction	0.25	0.42	0.51	0.96
-----------------	------	------	------	------

- (iii) Overlay the fitted logistic regression line on a plot of Y against X

Problem Session 3

Question: Log-linear regression

In a large-scale clinical trial of a new chemotherapy drug which is administered intravenously, the number of patients suffering from extravasation (leakage of the drug from the vein into surrounding subcutaneous tissue) is recorded across a range of delivery rates (drops per minute). Read the file `cytotoxic` into R.

Follow the direction of the lecture to perform a suitable log-linear regression. Use code similar to that shown in the previous question, but make sure you switch the model `family` to `poisson`. Make sure you can:

- (i) Check the significance of the predictor (independent) variable
- (ii) Write down your log-linear regression equation, and use it to estimate the number of patients who would suffer extravasation with delivery rates of 17 and 25 drops per minute. Why should we not use our fitted model to predict extravasation frequency for a delivery rate of 1 drop per second?
- (iii) Should we switch to a negative binomial regression model here?

Lesson 9: Classification

Here we will carry out a classification analysis on the `Process` data. This can be imported into R using the `read.csv` command. We can attach this data and get a glimpse of the variables using the `attach` and `head` command respectively:

```
1 attach(process)
2 head(process)
```

Problem Session 1

Recall that to run a logistic regression we use the `glm` command, specifying that `family = "binomial"`. To access the `roc` function (used to calculate AUC values), we first need to (install and) load the `pROC` package.

Question: Logistic regression

1. Carry out a logistic regression on whether the distillation process results in a pale yellow colour, using all other variables in the data as predictors. To do this we will need to create a binary variable, which takes the value 1 if the solution is pale yellow, and 0 otherwise. We can do this using the `as.numeric` command, which takes a TRUE/FALSE statement as an input, and outputs a 1 if the statement is true, and 0 if the statement is FALSE.

```
1 py = as.numeric(solution.colour.after.distillation == "pale yellow")
```

From the resulting model output, which variables affect the likelihood of a pale yellow solution?

2. Obtain the resulting confusion matrix. What is the sensitivity, specificity and misclassification rate for this classifier?
3. Repeat the analysis but this time only using `assay.after.distillation.wt` as a predictor in the logistic regression model. How does the performance compare?
4. Calculate AUC values for the full and reduced classifier. Which performs best overall?

Problem Session 2

In this session we're going to look at using the impurity measures: `impurity.A..`, `impurity.B..` and `impurity.C..` to predict `process.variant`. To make things a bit easier, it's recommended to create a new data frame containing just the variables we'll need for this analysis. We can extract columns from a data frame in R using square brackets:

```
1 var.imp = process[ , c("process.variant",  
2                       "impurity.A..",  
3                       "impurity.B..",  
4                       "impurity.C..")]
```

To access the LDA/QDA functions, we first need to load the `MASS` package.

Question: Discriminant analysis

1. Carry out linear discriminant analysis to predict the process variant using the three measures of impurity as predictors. Obtain the confusion matrix. How does this classifier perform?
2. Repeat the above analysis using quadratic discriminant analysis. Based on their misclassification rates, which classifier performs best?
3. Use 10-fold cross validation to obtain misclassification rates for LDA and QDA. Has this changed our conclusions?

Problem Session 3

In this session we'll be looking at implementing a k -nearest neighbours algorithm in R. To access the `knn` function we need to (install and) load the `class` package.

Question: Nearest neighbour analysis

1. Using the $k \approx \sqrt{n}$ rule, what would be a sensible initial value of k for a k -nearest neighbours analysis of this data?
2. Using your chosen value of k , run a KNN analysis to predict the process variant using all the three measures of impurity using 10-fold cross validation and report the CV misclassification rate.
3. Repeat the KNN analysis for values of k close to your chosen value from the previous part and produce a plot of misclassification rate against k . Which value of k produced the best classifier?
4. How does the optimal KNN classifier compare with the LDA/QDA classifiers from the previous problem session? Why do you think this is?

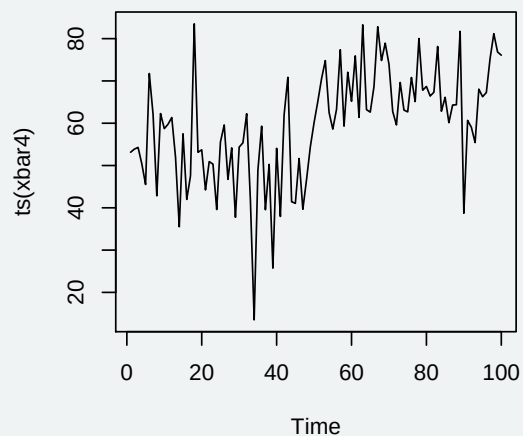
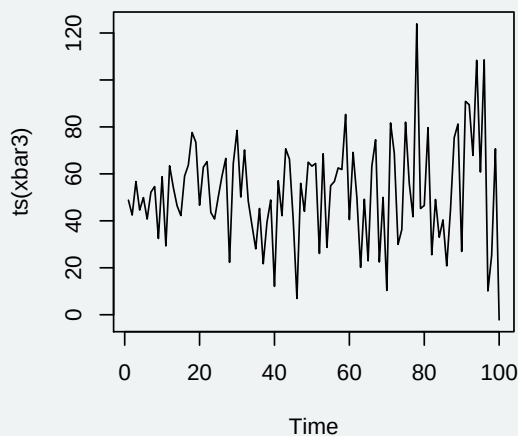
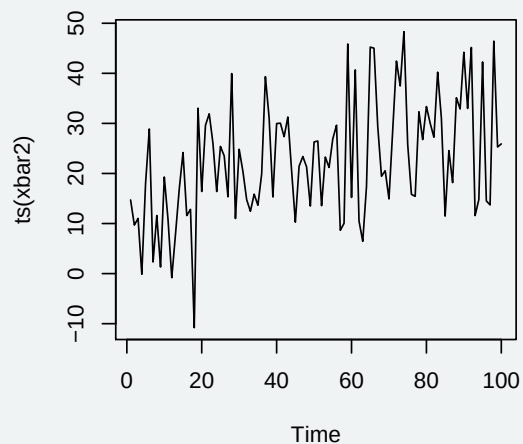
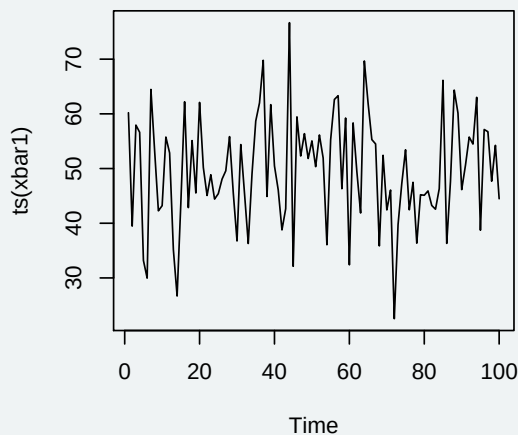
Lesson 10: Statistical Process Control

Problem Session 1

Question: Process variation

Look at the following $\bar{x}$ -charts. Which charts do you think indicate that the associated processes are under control?

For those charts that might indicate an out-of-control process, do you think this has been caused by (i) level shift; (ii) instability; (iii) trend, or (iv) cycle?



Problem Session 2

Question: Control limits and false alarms

- (a) Calculate the 2.5–sigma control limits for the tablet weight example used in the lecture, assuming a process mean and standard deviation of $\mu = 500$ mg and $\sigma = 15$ mg (respectively). Recall that the samples means were obtained from samples of size 20.
- (b) Assuming the 2.5–sigma control limits you found in part (a), what is the probability that on any sample the process will give a *false alarm*?

Question: Control limits and false alarms

A firm manufactures notebook computers. For one component, the company draws samples of size 10 every half an hour. The company makes 4000 of these components every hour. The control limits of the $\bar{x}$ –chart are set at 3 standard errors from the mean.

On average, how many units will be produced until the control chart gives a *false alarm*?

Problem Session 3

Question: Control charts when population parameters are unknown

Load the `tablet2` data used in the lecture, in which we have the weights of four tablets taken from the production line every hour, over a 25-hour period (updated to 31 hours with the inclusion of six new samples). Recall that the nominal tablet weight is 500mg. Produce $\bar{X}$, S and R charts, assuming both the population mean and variance are unknown. The following code will be useful:

```
1 install.packages("qcc")
2 library(qcc)
3 qcc(tablet2, type="xbar")
4 qcc(tablet2, type="S")
5 qcc(tablet2, type="R")
```

Question: I-MR Charts

I-MR (Individual-Moving Range) charts are used when data are continuous and not collected in subgroups; in other words, we have a single observation being made at every sample time-point. Two separate charts are produced, together providing a complete picture of process behaviour. Essentially a time series plot of the individual

observations, the *I*-chart tracks our observations through time; the *MR*-chart monitors the variability of the process by displaying, at each time-point, the absolute difference between the current observation and that at the previous time-point.

1. Always interpret the *MR*-chart first. The control limits on the *I*-chart are functions of the Moving Range, and so if this chart is out-of-control then the control limits on the *I*-chart will be inaccurate: any lack of control in the *I*-chart may be due to unstable variation, not actual changes in the process centre. If the *MR*-chart is in control, you can be sure that any out-of-control *I*-chart is due to changes in the process centre.
2. Now interpret the *I*-chart in the same way we did with the $\bar{X}$ -chart in the lecture.

The two-step procedure outlined above, for interpretation of *I-MR* charts, is exactly as discussed in the lecture when interpreting *S* and $\bar{X}$ -charts together.

Fifty-three batches of an API have been manufactured to date. Although all of the batches meet the customer's specification, some of the assay values are close to the specification limits, increasing the probability of failure in the future. Tightening the natural control limits on the assay measurement of an API, so they are much tighter than the customer's specification, has been targeted as a means of improving the manufacturing process. UCL and LCL were estimated on the basis of the historical data and used as a baseline for monitoring of the future batches. Each and every data point that goes beyond the control limits in future batches is to be investigated and assignable causes addressed, as part of Phase I of the improvement initiative, leading to narrowing of the internal control limits.

Load the [API](#) data into R, and use the following code to produce an *I-MR* chart. Interpret your charts.

```
1 qcc(API, type="xbar.one")
2 #The following code is required to generate the MR-plot:
3 range.cal = matrix(cbind(API[1:length(API)-1], API[2:length(API)]), ncol=2)
4
5 #And then the I-plot:
6 qcc(range.cal, type="R")
```