

Environmental Informatics \ Time Series Analysis

Topic Notes and Exemplars

These don't include all the exercises, Just a couple of exemplars

Exercises

Week 1 | Material Due: 17 July 2017

Regulations relating to Environmental Hypotheses Testing

Things that require measurement's, ostensibly by law, would include fish populations, ocean acidity, CO₂ levels, temperature, rainfall etc.ss

Finding which legislative instrument provides for this is difficult, a cursory glance through *Westlaw*, *LexisNexis*, *Google* and *Austlii* does not provide anything obvious.

Summary of Temperature Data

A table of data with rows as observations and columns as variables is a data frame.

First few Lines

```
> # Print out the Data Frame
> head(Temperature)
# A tibble: 6 x 3
  Day `Max Temperature` `Min Temperature`
  <int>          <dbl>          <dbl>
1     1           38.1           20.7
2     2           32.4           17.9
3     3           34.5           18.8
4     4           20.7           14.6
5     5           21.5           15.8
6     6           23.1           15.8
```

Structure of the Data Set

```
> str(Temperature)
Classes 'tbl_df', 'tbl' and 'data.frame': 365 obs. of  3 variables:
 $ Day          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Max Temperature: num  38.1 32.4 34.5 20.7 21.5 23.1 29.7 36.6 36.1
20.6 ...
 $ Min Temperature: num  20.7 17.9 18.8 14.6 15.8 15.8 15.8 17.4 21.8
20 ...
```

Summary of the Data Frame

```
> summary(Temperature)
   Day      Max Temperature  Min Temperature
Min.   : 1      Min.   : 9.70   Min.   : 2.10
1st Qu.: 92      1st Qu.:15.50   1st Qu.: 8.30
Median :183      Median :19.50   Median :11.20
Mean   :183      Mean   :20.51   Mean   :11.52
3rd Qu.:274      3rd Qu.:23.70   3rd Qu.:14.40
Max.   :365      Max.   :41.80   Max.   :25.00
```

Correlation of Minimum and Maximum Temperature

```
> max_temp <- Temperature$`Max Temperature`  
> min_temp <- Temperature$`Min Temperature`  
> cor(max_temp, min_temp)  
[1] 0.7498076  
>  
> cor(min_temp, max_temp)  
[1] 0.7498076
```

The value provided by the `cor` function is the default *Pearson* method¹ which is a *linear correlation coefficient* such that:²

$$r = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2} \times \sqrt{n(\sum y^2) - (\sum y)^2}}$$

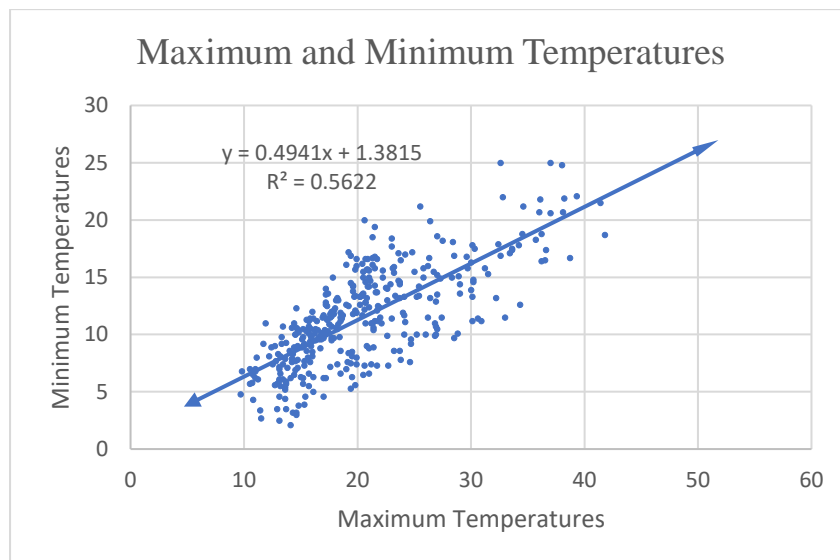
The value of r measures the strength and direction of a linear relationship by comparing the variation, whereby a value of 1 would occur if all the data were to lie exactly on a straight line, a strong correlation is usually $r > \pm 0.8$.

The Coefficient of determination is r^2 :

$$r^2 = 0.7498076^2 = 0.562211$$

The Coefficient of determination is the percentage of variation that is explained by the linear function, so in this case 56% of the variation between maximum and minimum temperatures can be explained by a linear function between those two variables, the linear function can be found by using an RSS method, explained by “\Simple\ Linear\ Regression.pdf”.

This is the same method used with Excel:



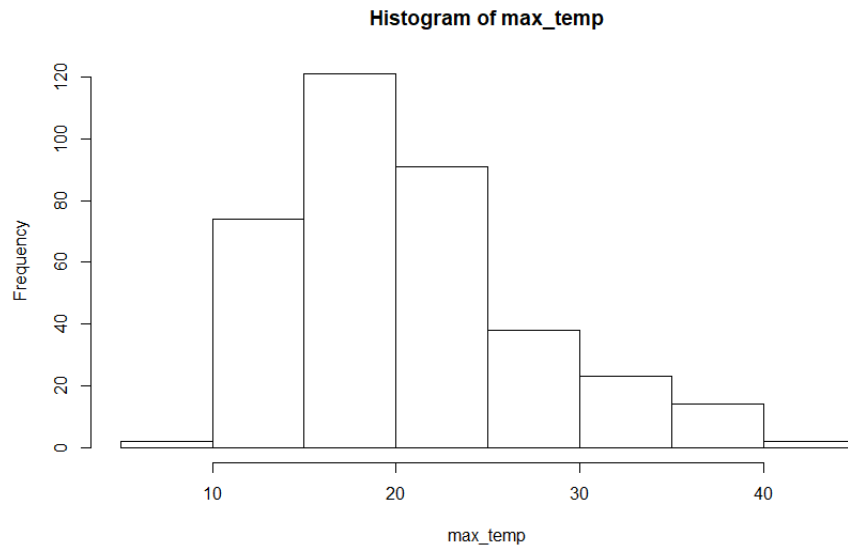
¹ Rdocumentation.org. (2017). *cor function* / R Documentation. [online] Available at: <https://www.rdocumentation.org/packages/stats/versions/3.4.1/topics/cor> [Accessed 21 Jul. 2017].

² Roberts, D. (2017). *Statistics 2 - Correlation Coefficient and Coefficient of Determination*. [online] Mathbits.com. Available at: <https://mathbits.com/MathBits/TISection/Statistics2/correlation.htm> [Accessed 21 Jul. 2017].

1.5 Re-Produce Figures³

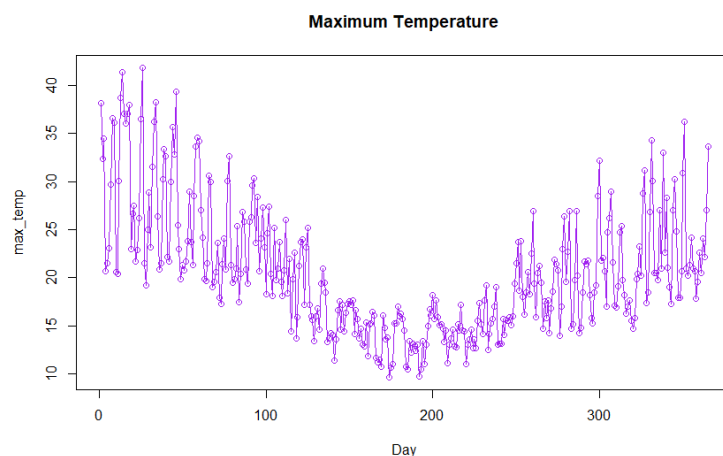
Histogram

```
> hist(max_temp)
```



Line Plot

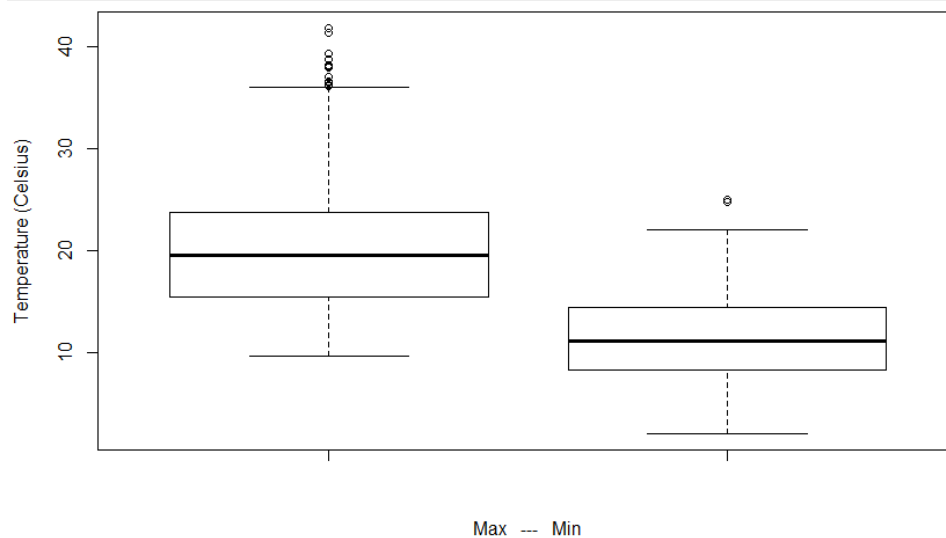
```
plot(
+     Temperature$Day,
+     max_temp,
+     type="o",
+     xlab="Day", ylab="max_temp", main="Maximum Temperature",
+     col="purple"
+ )
```



³ Use this cite as a man page:
https://www.tutorialspoint.com/r/r_line_graphs.htm

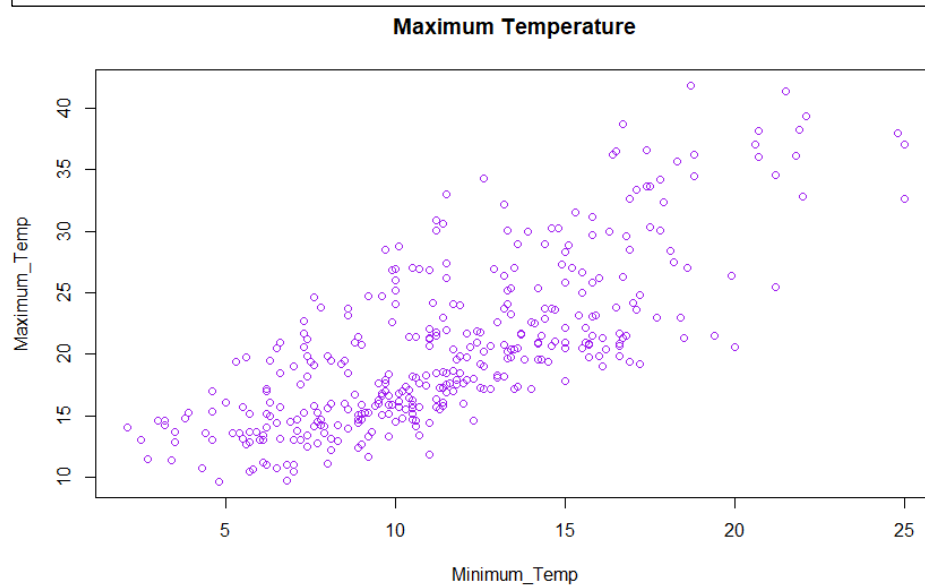
Box Plot

```
plot(
+     Temperature$Day,
+     max_temp,
+     type="o",
+     xlab="Day", ylab="max_temp", main="Maximum Temperature",
+     col="purple"
+ )
```



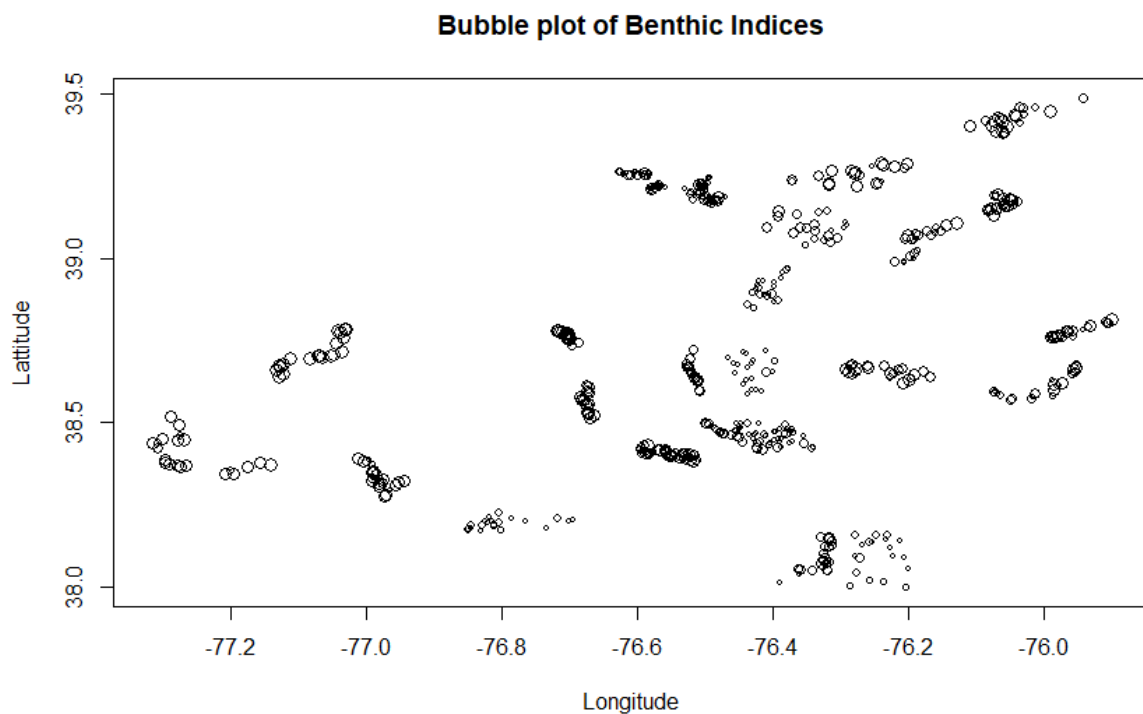
Scatter Plot

```
> plot(
+     min_temp,
+     max_temp,
+     type="p",
+     xlab="Minimum_Temp", ylab="Maximum_Temp", main="Maximum Temperature",
+     col="purple"
+ )
```



Bubble Plot

```
> x_bubble <- Benthic.df$Longitude
> y_bubble <- Benthic.df$Latitude
>
> plot(x_bubble, y_bubble,
+      type="n",
+      main="Bubble plot of Benthic Indices",
+      xlab="Longitude", ylab="Latitude"
+    )
>
> symbols(x_bubble, y_bubble,
+        circles = sqrt(Benthic.df$Index),
+        add=T,
+        inches=0.05
+    )
```



Parameter Estimation and Hypothesis Testing

Week 2 Material, Due 24 July 2017

Exercises 2.1

Measuring the average size of a type of fish in a river.

Every fish in the river would represent the population of fish.

Samples will be taken by measuring a collection of fish at various locations.

The random variable is the length of that type of fish.

This experiment would be normally distributed.

A Note on R and Standard Normal Tables of distribution

It's easy to forget that a standard normal distribution has a function of:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

And hence the probability, would be the area under the curve:

$$P(a < x < b) = \int_a^b \left[\frac{1}{\sqrt{2\pi\sigma^2}} \times e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)} \right] dx$$

The whole point of using R or tables of a Standard Normal distribution is to totally bypass that function and just use, in the case of a table, pre-allocated solutions for a standard distribution of $\mu = 0$, $\sigma = 1$.

Normal Distribution in R

The following commands allow R to work with Normal Distributions:

dnorm.....Outputs the value (height) of the curve (frequency), given input of an x-value

pnorm.....Outputs the area under the curve (probability), given input of an x-value

qnorm.....Outputs the x-value (relevant measurement), given input of probability (area)

rnormGenerates a random normal distribution.

Exercises 2.2

Empirically

Empirically, a 95% confidence interval will span two standard deviations from the mean, hence 5% of the data under the distribution will be less than 10 ppb or greater than 30 ppb.

Thus 2.5% of the time the well will naturally have more than 30 ppb of the chemical.

In R

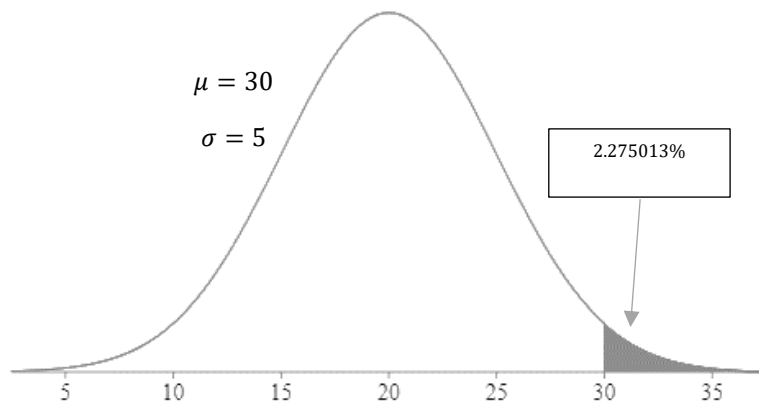
If we were to do this in R though, the question would become:

Determine probability of a sample-value $x > 30$, given that the population is normally distributed with a mean value of $\mu = 20$ and standard deviation of $\sigma = 5$ ($X \sim N(\mu = 20, \sigma = 5)$)

In *R* the `pnorm` command does this:

```
> z <- (30-20)/5  
> pnorm(z, mean=0, sd=1, lower.tail=FALSE, log.p=FALSE)  
[1] 0.02275013
```

Thus 2.275% of the time the chemical levels will be above 30 ppb in the well.



Exercise 2.3

These answers can all be confirmed via *StatTrek's* calculator.⁴

Part A;

To find the value on the normal distribution given a percentile we would use the *qnorm* command in R:

```
> qnorm(0.05, mean=0.5, sd=0.8, lower.tail=FALSE, log.p=FALSE)
[1] 1.815883
```

Thus a concentration of 1.815883 represents the upper 95th percentile of concentration at the background site.

Part B;⁵

So the question becomes:

What is the probability that the sample mean of \bar{x} from the sample of 1 is greater than 1.815883?

Let:

- \bar{x} be the sample mean of the concentration
- $n = 1$ be the number of samples taken
- σ_{SE} be the standard deviation of all possible means from a sample of size n ; equivalently known as the standard error of the sample mean:
 $SE(\bar{x}) = SEM = SE = \sigma_{SE}$
- x be the TcCB concentration at the background site.
- $\sigma = 0.8$ be the standard deviation of the concentration of TcCB at the background site.
- $\mu = 0.5$ be the mean value of concentration of TcCB at the background site

NB: This is not a *Student's t-distribution* as we already know the population mean μ and the population standard deviation σ , this is calculating the probability for a sample mean given a population.

Which Statistical Test

- A distribution of all mean values, of all samples drawn from a population tend to be normally distributed, this is provided by the *Central Limit Theorem*⁶;
 - if the population distribution was *non-normal*, a sampling distribution of the sample mean \bar{x} would be *approximately* normally distributed for large samples by the Central Limit Theorem ($n \geq 30$)
 - However if the population has a *normal* distribution, the sampling distribution of \bar{x} will be *exactly* normal no matter what the sample size is.

⁴ <http://stattrek.com/online-calculator/normal.aspx>

⁵ Mendenhall, W. and Beaver, R. (2013). *Introduction to probability and statistics*. Boston, MA: CL-Wadsworth, pp.254-258.

⁶ Mendenhall, W. and Beaver, R. (2013). *Introduction to probability and statistics*. Boston, MA: CL-Wadsworth, pp.251.

- As the concentration value of TcCB follows a normal distribution, the sample means are also normally distributed. Thus we can form a normal distribution of samples, observe that σ_{SE} is used and not σ :
 - σ is used when finding an area under the probability distribution of x , that is the distribution of actual values centred about the population mean value μ .
 - σ_{SE} is used when finding the area under the curve for a sampling distribution of \bar{x} , that is the distribution of all sample mean values centred about the mean value of sample means $\mu_{\bar{x}}$:
 - The average mean value $\mu_{\bar{x}}$ of a normal sampling distribution, is actually the population mean μ , so $\mu_{\bar{x}} = \mu$
 - This is provided by the *Central Limit Theorem*.
- This is not a *Student's t-distribution* as we already know the population mean μ and the population standard deviation σ , this is calculating the probability for a sample mean given a population.
 - If the population standard deviation was unknown, then a t-distribution may be used.

Solving Via a Standard Normal table of distribution

1. Find the corresponding Z-Value for the sample mean:

$$z = \frac{\bar{x} - \mu}{\sigma_{SE}}$$

- a. The value of σ_{SE} :

$$\begin{aligned}\sigma_{SE} &= \frac{\sigma}{\sqrt{n}} \\ &= \frac{0.8}{\sqrt{1}} \\ &= 0.8\end{aligned}$$

$$\begin{aligned}z &= \frac{\bar{x} - \mu}{\sigma_{SE}} \\ &= \frac{1.815883 - 0.5}{0.8} \\ &= 1.64485375\end{aligned}$$

2. Find the probability above this value on a *Standard Normal Distribution*:

$$\begin{aligned}P(\bar{x} > 1.815883) &= P(z > 1.64485375) \\ &\approx P(z > 1.64) \\ &= 0.0505\end{aligned}$$

Thus the probability that a sample will exceed the 95th percentile of the background site is 5%.

Solving in R

```
####Begin
####Assign Variables####
sd  <- 0.8           #Std. Deviation of Background Site TCCB
mu  <- 0.5           #Mean of Background Site TCCB

n   <- 1             #Sample Size for part B question
SE  <- s.d._2.3/sqrt(n_2.3b) #Standard Error of the Sample Mean

prcnt <- 95          #Relevant Percentile
alpha <- 1-prcnt_23/100 #Relevant alpha value

####Find the value of the 95th Percentile

TCCB_95th_backup <- qnorm(alpha, mean=mu, sd, lower.tail=FALSE, log.p=FALSE)

####Find the probability

answer <- pnorm(1.815883, mean=0.5, sd = 0.8, lower.tail = FALSE, log.p = FALSE)

#Probability of sample exceeding 95th percentile of backup site:
percent(answer)
[1] "5%"
####End
```

Thus there is a 5% chance that the sample will exceed the 95th percentile of the background site.

Part C;

This question relates to a Binomial Distribution;

A statistical experiment will follow a binomial distribution where:⁷

- The experiment consists of n repeated trials (in this case 10 samples).
- Each Trial can result in either success or failure (in this case above or below the 95% threshold)
- The probability of success, denoted by P , is the same on every trial (In this case 5.05% from part B;)
- Each trial or sample is independent

Detailed Formula

$$P(x = k) = \binom{n}{k} p^k q^{n-k}$$
$$= \frac{n!}{k! (n - k)!} \times p^k q^{n-k}$$

Where:

n Is the number of trials

k Is the number of successes

p Is the probability of Success

q Is the probability of failure

Find the Solution

1. State the Equation:

$$P(x = k) = \frac{n!}{k! (n - k)!} \times p^k q^{n-k}$$

2. Substitute the values:

$$\begin{aligned} P(x \geq 2) &= 1 - P(x < 2) \\ &= 1 - P(x < 2) - P(x < 1) - P(x < 0) \\ &= 1 - \frac{10!}{2! (10 - 2)!} \times 0.05^2 \times 0.95^{10-2} - \frac{10!}{1! (10 - 1)!} \times 0.05^1 \times 0.95^{10-1} - \frac{10!}{0! (10 - 0)!} \times 0.05^0 \times 0.95^{10-0} \\ &= 1 - 0.0746 - 0.315 - 0.5987 \\ &= 0.012 \end{aligned}$$

3. Conclusion:

Thus there is a 1.2% probability of finding 2 samples above the 95th percentile concentration of TcCB from a batch of 10 samples taken from a remediated site, given that the background site has TcCB concentrations distributed normally with $\sigma = 0.8$ and $\mu = 0.5$.

⁷ <http://stattrek.com/probability-distributions/binomial.aspx>

Solving in R

To solve this in R we need to use the `dbinom` command:

```
###Begin
```

```
#We need to use a binomial distribution to determine this probability
```

```
####Variables
```

```
p <- answer_23b #The probability success, i.e. that a Sample will be above the 95th percentile
```

```
q <- 1-p_23c    #The probability of failure, as above.
```

```
k <- 2          #The relevant number of successes, the relevant number of samples above the 95th percentile
```

```
n <- 10         #The number of samples taken
```

```
###Find the probability
```

```
answer <- pbinom(k_23c, n_23c, p_23c, lower.tail=FALSE, log.p=FALSE)
```

```
##The probability, of 2 samples of the 10, being above the 95th percentile is:
```

```
percent(answer)
```

```
[1] "1.15%"
```

```
#End
```

Thus there is a 1.15% probability of finding 2 samples above the 95th percentile from a batch of 10. Observe that this probability suffers from less rounding error and is hence more accurate.

Exercise 2.4

List the detailed formulas for calculating confidence intervals

Normal Distribution

The Normal Distribution is a bell curve distribution of continuous data, a population may be normally distributed with a population mean of μ and population standard deviation of σ . The Central Limit Theorem provides that the sums, means and sampling statistics taken from random samples of measurements will tend to have a normal distribution⁵, thus many continuous values measured from large populations, such as height, weight, etc. will tend to be normally distributed.

The Student's t distribution

The Central Limit Theorem provides that the sampling distribution of a statistic (e.g. a mean value), will follow a normal distribution, hence, as long as the population mean μ and the population standard deviation σ is known a Standard Normal Distribution can be used with a z-score to calculate the probability of finding a sample with that statistic (e.g. mean).⁸

However, if:

1. the population statistics σ and μ are unknown, OR
2. A population is not normally distributed, AND
 - a. The sample size is not sufficiently large (Usually $n > 30$)

Then the Normal distribution could not be used, hence the t distribution is relied upon:

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

Where s is the sample standard deviation and \bar{x} is the sample mean. Relative to the sample size taken, the t distribution comes in many different forms, determined by the *degrees of freedom*.

The t distribution can be used in any of the following situations:

1. The population distribution is normal
2. The sample size is sufficiently large (Usually $n \geq 30$)

⁸ 5Mendenhall, W. and Beaver, R. (2013). Introduction to probability and statistics. Boston, MA: CL-Wadsworth, pp.251.

Confidence Intervals

Normal Distribution

A $(1 - \alpha)\%$ confidence interval of a normal population is provided by:

$$\mu \pm \sigma \times Z_{\alpha/2}$$

Where:

μ is the population mean

σ is the population standard deviation

Z_{α} is the x -value on a standard normal distribution such that the area under the curve preceeding it is equivalent to the α value.

α is the probability of incorrectly rejecting the null hypothesis, i.e. the probability of incorrectly assuming that something happened when it didn't.

Student's t -Distribution

A $(1 - \alpha)\%$ confidence interval of a normal population is provided by:

$$\bar{x} \pm \left(\frac{s}{\sqrt{n}} \right) \times t_{\alpha/2, d.f.}$$

Where:

\bar{x} is the sample mean

s is the sample standard deviation

n is the sample size

$d.f.$ is the degrees of freedom of the t -distribution: $d.f. = n - 1$

$t_{\alpha, d.f.}$ is the x -value on a t -distribution (which is normal) of given degrees of freedom, which is, such that the area under the curve is equivalent to α .

Assuming:

- The samples are random
- The observations are independent of each other
- The sample was sufficiently large ($n \geq 30$)

Quality Control: Prediction Intervals, Tolerance Intervals and Control Charts

Week 3 Material | Topic 3

Introductory Material

Types of Intervals

- A **confidence interval** provides an interval for the true mean value at a specified probability $(1 - \alpha)$.
- A **prediction interval** provides an interval for that will contain k future values with a specified probability $(1 - \alpha)$.
- A **tolerance interval** provides an interval that contains a proportion of all future observations, this interval is referred to as the coverage
 - A tolerance interval uses, for history sake, β , to represent the coverage, but is not referring to the probability of a type II error (Incorrectly rejecting the alternative hypothesis)

A confidence interval should not be used to forecast future observations as the interval for the mean is much narrower than prediction intervals, giving an exaggerated accuracy for the forecast.⁹

| Distribution | Function name | Description |
|---------------|-----------------------------|--|
| Normal | predIntNorm | Construct a <i>prediction interval</i> for the next k observations or next k means from a normal distribution |
| | predIntNormK | Compute the value of K for a prediction interval for a normal distribution |
| | predIntNormSimultaneous | Construct a <i>simultaneous prediction interval</i> for the next r sampling occasions based on a normal distribution |
| | predIntNormSimultaneousK | Compute the value of K for a simultaneous prediction interval for the next r sampling occasions based on a normal distribution |
| Lognormal | predIntLnorm | Construct a <i>prediction interval</i> based on a lognormal distribution |
| | predIntLnormAlt | |
| | predIntLnormSimultaneous | Construct a <i>simultaneous prediction interval</i> based on a lognormal distribution |
| | predIntLnormAltSimultaneous | |
| Gamma | predIntGamma | Construct a <i>prediction interval</i> based on a gamma distribution |
| | predIntGammaAlt | |
| | predIntGammaSimultaneous | Construct a <i>simultaneous prediction interval</i> based on a gamma distribution |
| | predIntGammaSimultaneousAlt | |
| Poisson | predIntPois | Construct a prediction interval for the next k observations or sums from a Poisson distribution |
| Nonparametric | predIntNpar | Construct a nonparametric prediction interval for the next k of m observations |

⁹ Robjhyndman.com. (2017). *The difference between prediction intervals and confidence intervals* | Rob J Hyndman. [online] Available at: <https://robjhyndman.com/hyndsight/intervals/> [Accessed 27 Jul. 2017].

Standard Error vs Standard Deviation¹⁰

Suppose that we were to draw every possible sample of size n , from a population of size N and computed the mean for each one of those samples.

Mean

The distribution of those sample means would be such that the most common sample mean would be the population mean (i.e. the mean of the sample means is the population mean):

$$\mu_{\bar{x}} = \mu$$

Standard Deviation

The standard deviation of those sample means would be:

$$\sigma_{\bar{x}} = \left[\frac{\sigma}{\sqrt{n}} \right] \times \sqrt{\frac{N-n}{N-1}}$$

However, where N is much larger than n (e.g. 20 fold) it can be observed that the following is a good estimate where N is not known or not knowable:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Where σ is unknown, a t distribution is used such that s can be used to provide an estimate for σ and that uncertainty can be rolled into the confidence interval.

Use in an Interval

As μ and $\frac{\sigma}{\sqrt{n}}$ represent the parameters of the sampling distribution, when using a sampling distribution to predict the probable population mean from a sample, it is appropriate to use those values, such as in a control chart.

¹⁰ Stattrek.com. (2017). *Sampling Distribution*. [online] Available at: <http://stattrek.com/sampling/sampling-distribution.aspx> [Accessed 26 Jul. 2017].

Confidence Interval

A $(1 - \alpha) \times 100\%$ Confidence interval is provided by:

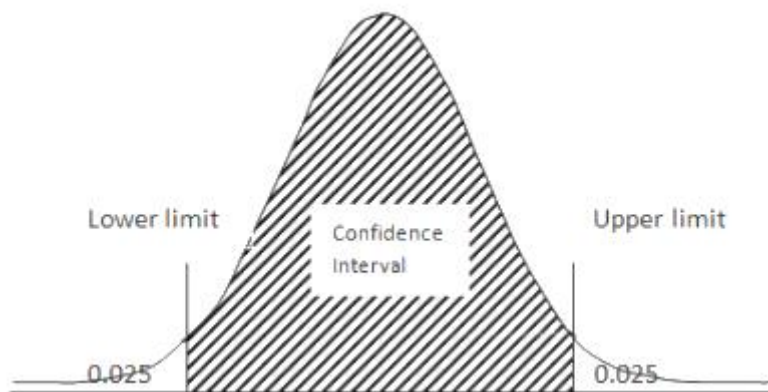
$$\mu \pm \sigma \times z_{\frac{\alpha}{2}}$$

This provides an interval for the true mean value such that there is only an $(\alpha \times 100)\%$ probability of incorrectly rejecting the null hypothesis

Solving Confidence Intervals in R

The `t.test` command will draw up a confidence interval, or of course the actual formula can be used, which is included in the Wk. 2 work:

```
t.test(  
  data_values,  
  alternative = "two.sided",  
  mu = 0,  
  conf.level = 0.95  
)
```



Prediction Intervals

A $(1 - \alpha) \times 100\%$ Probability interval is provided by:¹¹

$$\bar{x} \pm K \times s$$

A $(1 - \alpha) \times 100\%$ Probability interval for $k = 1$ is provided by:¹²

$$\bar{x} \pm t_{1-\frac{\alpha}{2}, n-1} \times s \sqrt{1 + \frac{1}{n}}$$

This predicts the value of the next sample value (X_{n+1}), given a sample set $\{X_1, X_2, X_3 \dots X_n\}$. A prediction interval is an interval for the variable itself, rather than a parameter of the distribution (e.g. μ or σ) as contrasted with a confidence interval.

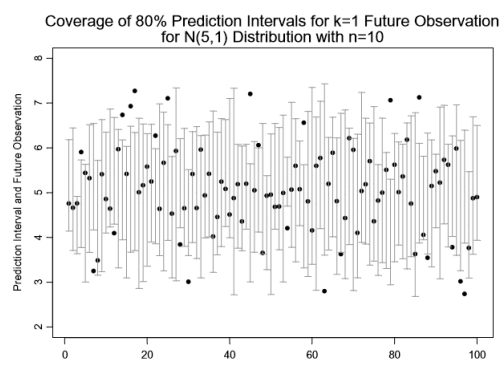
An (for example) 80% prediction interval will not contain 80% of future observations, rather, for a number of successive samples from the same population, the interval will contain a future observation 80% of the time.¹³

The Mathematics of a prediction interval

At equation [6.22] *Environmental Statistics with S-Plus*¹⁴ provides the mathematics behind a confidence interval, the mathematics in horrendous and unusable, e.g. solving:

$$1 - \alpha = \int_0^\infty \left[\int_{-\infty}^\infty \Phi \left(\frac{x + y \sqrt{\frac{1}{\frac{n}{m} + 1}}}{\sqrt{1 - \frac{1}{\frac{n}{m} + 1}}} \right) \times \phi(y) \right] dy \times \frac{(s\sqrt{n-1})^{v-1} \times e^{\left(-\frac{(s\sqrt{n-1})^2}{2}\right)}}{2^{\frac{(v-1)}{2}-1} \times \Gamma\left(\frac{v-1}{2}\right)} \times \sqrt{n-1} ds$$

Hence **R** is used to solve probability Intervals



¹¹ Millard, S. and Neerchal, N. (2001). *Environmental statistics with S-Plus*. Boca Raton: CRC Press.

¹² Propharmagroup.com. (2017). *Prediction Interval - Statistical Intervals Pt. 2 | ProPharma Group*. [online] Available at: <http://www.propharmagroup.com/blog/understanding-statistical-intervals-part-2-prediction-intervals> [Accessed 27 Jul. 2017] and Millard, S. and Neerchal, N. (2001). *Environmental statistics with S-Plus*. Boca Raton: CRC Press [6.15].

¹³ Propharmagroup.com. (2017). *Prediction Interval - Statistical Intervals Pt. 2 | ProPharma Group*. [online] Available at: <http://www.propharmagroup.com/blog/understanding-statistical-intervals-part-2-prediction-intervals> [Accessed 27 Jul. 2017].

¹⁴ Millard, S. and Neerchal, N. (2001). *Environmental statistics with S-Plus*. Boca Raton: CRC Press.

Solving Prediction Intervals in R

The *EnvStats* package is used for this (although the generic `predict` command could be tweaked for use).

The `predInt...` family of commands can be used to solve prediction intervals and is provided by the *EnvStats* book:¹⁵

| Distribution | Function name | Description |
|---------------|--|--|
| Normal | <code>predIntNorm</code> | Construct a <i>prediction interval</i> for the next k observations or next k means from a normal distribution |
| | <code>predIntNormK</code> | Compute the value of K for a prediction interval for a normal distribution |
| | <code>predIntNormSimultaneous</code> | Construct a <i>simultaneous prediction interval</i> for the next r sampling occasions based on a normal distribution |
| | <code>predIntNormSimultaneousK</code> | Compute the value of K for a simultaneous prediction interval for the next r sampling occasions based on a normal distribution |
| Lognormal | <code>predIntLnorm</code> | Construct a <i>prediction interval</i> based on a lognormal distribution |
| | <code>predIntLnormAlt</code> | |
| | <code>predIntLnormSimultaneous</code> | Construct a <i>simultaneous prediction interval</i> based on a lognormal distribution |
| | <code>predIntLnormAltSimultaneous</code> | |
| Gamma | <code>predIntGamma</code> | Construct a <i>prediction interval</i> based on a gamma distribution |
| | <code>predIntGammaAlt</code> | |
| | <code>predIntGammaSimultaneous</code> | Construct a <i>simultaneous prediction interval</i> based on a gamma distribution |
| | <code>predIntGammaSimultaneousAlt</code> | |
| Poisson | <code>predIntPois</code> | Construct a prediction interval for the next k observations or sums from a Poisson distribution |
| Nonparametric | <code>predIntNpar</code> | Construct a nonparametric prediction interval for the next k of m observations |

¹⁵ Millard, S. (2013). *EnvStats: An R Package for Environmental Statistics*. Springer. P. 114 (p. 130 of 305); see also < <https://rdrr.io/cran/EnvStats/man/FcnsByCatPredInts.html> >

The commands have help pages which can be reached under the list of commands at *R-Studio's* help under the heading: "*EnvStats Functions for Prediction Intervals*" which can be found by searching "*FcnsByCatPredInts*":

```
Install.packages (EnvStats)
Library (EnvStats)

predIntNorm(
  x,
  n.mean = 1,
  k = 1,
  method = "Bonferroni",
  pi.type = "two-sided",
  conf.level = 0.95
)
```

Where:

| | |
|------------------------------------|---|
| <code>x</code> | A vector containing all the observations |
| <code>n.mean = 1</code> | The sample size of future predictions, usually just 1, i.e. single observations, not various averages |
| <code>k = 1</code> | The number of future observations that are going to be predicted |
| <code>method = "Bonferroni"</code> | This can be the approximate method "Bonferroni" or the "exact" method as discussed above |
| <code>pi.type = "two-sided"</code> | This can be "upper", "two-sided" or "lower" |
| <code>conf.level = 0.95</code> | The 1- α value relative to the probability of containing future predictions |

Observe that *EnvStats* uses "two-sided", where as the `t.test` command uses "two.sided".

Tolerance Intervals

The mathematics for a tolerance interval is equally horrendous and can be found somewhat discussed in chapter 6 of *Environmental Statistics with S-Plus*¹⁶

Solving Tolerance Intervals in R

The Tolerance Interval can be calculated using commands included with *EnvStats*, the family of commands is provided in the book:¹⁷

| Distribution | Function name | Description |
|---------------|----------------|---|
| Gamma | tolIntGamma | Construct a tolerance interval for a gamma distribution |
| Normal | tolIntGammaAlt | |
| | tolIntNorm | Construct a tolerance interval for a normal distribution |
| | tolIntNormK | Compute the value of K for a tolerance interval for a normal distribution |
| Lognormal | tolIntLnorm | Construct a tolerance interval for a lognormal distribution |
| Poisson | tolIntLnormAlt | |
| | tolIntPois | Construct a tolerance interval for a Poisson distribution |
| Nonparametric | tolIntNpar | Construct a nonparametric tolerance interval |

If a tolerance interval is based on background data, sample data from a site can be compared to the tolerance interval, if any data falls outside you can declare the site contaminated.

There are help pages dedicated to these commands, for instance the “*Tolerance Interval for a Normal Distribution*” page can be found by searching “tolIntNorm”:¹⁸

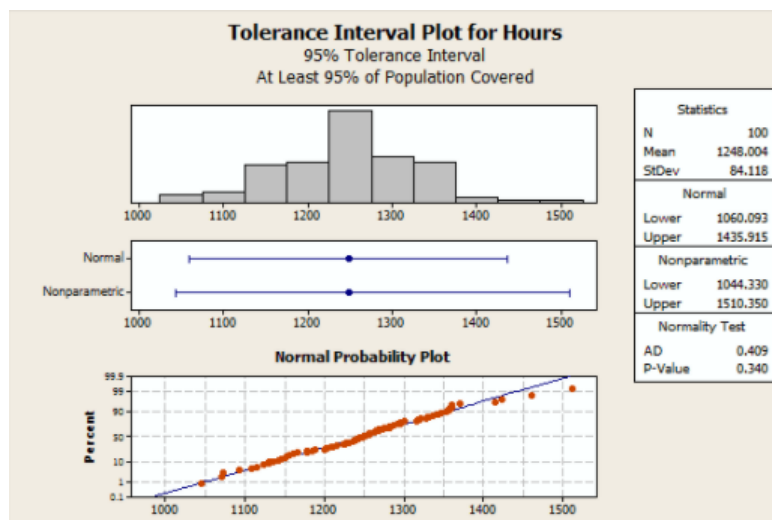


Figure 1 - Representing a tolerance interval (This is from Minitab)

¹⁶ Millard, S. and Neerchal, N. (2001). *Environmental statistics with S-Plus*. Boca Raton: CRC Press [6.21].

¹⁷ Millard, S. (2013). *EnvStats: An R Package for Environmental Statistics*. Springer. P. 141 (p. 157 of 305).

¹⁸ See also, Rdocumentation.org. (2017). *tolIntNorm function* / *R Documentation*. [online] Available at: <https://www.rdocumentation.org/packages/EnvStats/versions/2.1.0/topics/tolIntNorm> [Accessed 30 Jul. 2017].

```

Install.packages (EnvStats)
Library (EnvStats)

tolIntNorm(
    x,
    coverage = 0.95,
    cov.type = "content",
    ti.type = "two-sided",
    conf.level = 0.95,
    method = "exact"
)

```

Where:

| | |
|--------------------------------|---|
| <code>x</code> | A vector containing all the observations |
| <code>coverage</code> | The proportion of all future observations covered. |
| <code>cov.type</code> | "content" is an interval that contains at least the relevant proportion (β) of the population of future observations at a confidence level specified (α) |
| <code>ti.type</code> | "expectation" is such that the average coverage of the interval is provided by $1-\beta$ |
| <code>conf.level = 0.95</code> | This can be "upper", "two-sided" or "lower" |
| <code>method</code> | The 1- α value relative to the probability of containing future predictions |
| | This is only relevant for a two-sided test, and can be "exact" or "wald.wolfowitz" |

Compliance-to-Background Comparisons: Construct a tolerance interval based on background data, then compare data from a compliance well or site to the tolerance interval. If any compliance data are outside of the tolerance interval, then declare contamination is present.

Compliance-to-Fixed Standard Comparisons: Construct a tolerance interval based on compliance data, then compare the tolerance limit to a fixed standard (e.g., GWPS). If the tolerance limit is greater (less) than the fixed standard, declare contamination is present.

Control Charts

A control Chart is a visual display that shows how a process changes over time. Data is plotted in time order with upper and lower limits of expected variation in data.

Types of Control Charts

Control charts can plot the:

- sample mean (\bar{x} Chart)
- The sample range (R Chart)
- Or the sample proportion (p Chart)

\bar{x} Charts

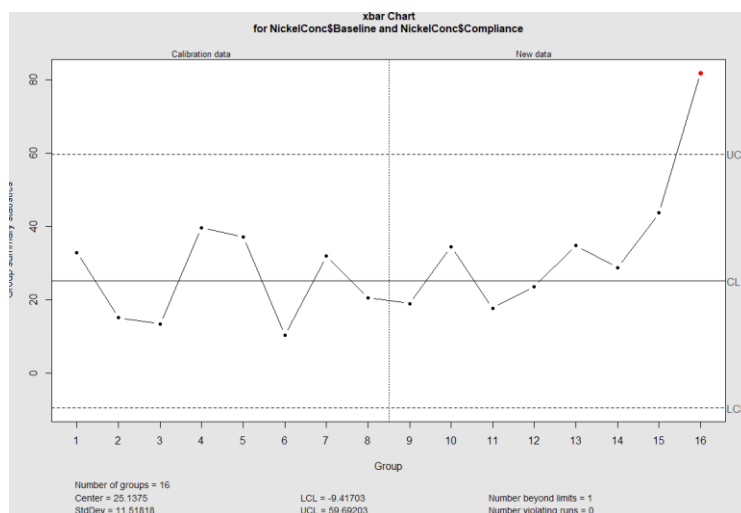
\bar{x} charts are what we are considering at this time, they are discussed at page 730 of the *Applied Statistics* textbook.¹⁹

In an \bar{x} control chart, the sample mean is plotted as an *upper control limit (UCL)* and *lower control limit (LCL)* is set, usually the UCL is set at $\pm 3\sigma$ because 99.73% of all sample means will fall within those limits on a normal distribution:

$$UCL = \mu + 3 \frac{\sigma}{\sqrt{n}}$$

$$LCL = \mu - 3 \frac{\sigma}{\sqrt{n}}$$

The standard error ($\frac{\sigma}{\sqrt{n}}$) is being used rather than the population standard deviation (σ) to form this interval because we are concerned with the distribution of means in a sampling distribution rather than the distribution of a variable.



¹⁹ Doane, D. and Seward, L. (2013). *Applied statistics in business and economics* 4th ed. New York, NY: McGraw-Hill/Irwin, ch. 17.4, p. 730.

Using R to generate Control Charts

Control Charts can be created using the “*quality control chart*”: `qcc` package and command:

```
install.packages(qcc)
library(qcc)

qcc (
  data,
  type="xbar",
  std.dev=sd(data),
  newdata = ...,
  confidence.level=0.95
)
```

Where:

| | |
|-------------------------------|---|
| <code>data</code> | A vector containing all the background observations, used from which to solve upper and lower control limits |
| <code>type</code> | What the control chart is plotting, e.g. sample mean (xbar), range (R), std. dev. (S) etc. |
| <code>sizes</code> | Specifies the sample sizes associated with each group (default is 1) |
| <code>std.dev</code> | The standard deviation of the process, so if the population standard deviation is known use that otherwise just use the <code>sd()</code> command |
| <code>newdata</code> | The data that's going to be plotted but is not going to be used to calculate descriptive statistics |
| <code>nsigmas</code> | How many std deviations from the mean (e.g. 2 sigmas is a 95% conf. level, 3 sigmas is a 99.7% conf. level) |
| <code>confidence.level</code> | The desired confidence level, this overrides the <code>nsigmas</code> and is probably more desirable to use because it saves working backwards |

Cumulative Summation Charts (CUSUM)

These charts use a running sum to form the upper and lower bounds and plot some such value.

These are good for plotting a gradual trend in data because they are more sensitive to a slow trend.

This can be observed, generate two sets of random data in R with a slightly higher mean in one and observe that the CUSUM will detect a change when the xbar chart does not.

Using R to generate Control Charts

Control Charts can be created using the `cusum` command:

```
cusum(  
  data,  
  std.dev=sd(data),  
  decision.interval = 4,  
  se.shift = 1,  
  newdata = ...,  
)
```

Where the variables are almost the same as an xbar chart, but the `decision.interval` variable refers to the number of standard errors of the summary statistics is required before the process is deemed 'out of control' (usually 4 or 5).

`Se.shift` refers to the amount of shift to detect in the process, probably just leave this as 1.

Exercise 3.2

Example 3.1

The table below shows arsenic concentrations (ppb) collected quarterly at two groundwater monitoring wells (data in “Arsenic.csv”).

| Well | Year | Observed Arsenic (ppb) | | | |
|------------|------|------------------------|------|------|------|
| Background | 1 | 12.6 | 30.8 | 52.0 | 28.1 |
| | 2 | 33.3 | 44.0 | 3.0 | 12.8 |
| | 3 | 58.1 | 12.6 | 17.6 | 25.3 |
| Compliance | 4 | 48.0 | 30.3 | 42.5 | 15.0 |
| | 5 | 47.6 | 3.8 | 2.6 | 51.9 |

Construct a prediction interval for the next 4 observations.

Solution

Using a Prediction Interval in R to test compare the data

The following code in R will provides that the next four observations (for year 4 or year 5), have a 95% probability of being less than 72.9:

```
#Use the predIntNorm command as found in the Help docs

Prediction_Interval <- predIntNorm(
  Arsenic$Background,
  n.mean = 1,
  k = 4,
  method = "exact",
  pi.type = "upper",
  conf.level = 0.95
)

#What is the Upper Value of the Prediction Interval?
UPL <- Prediction_Interval$interval$limits[2]
LPL <- 0

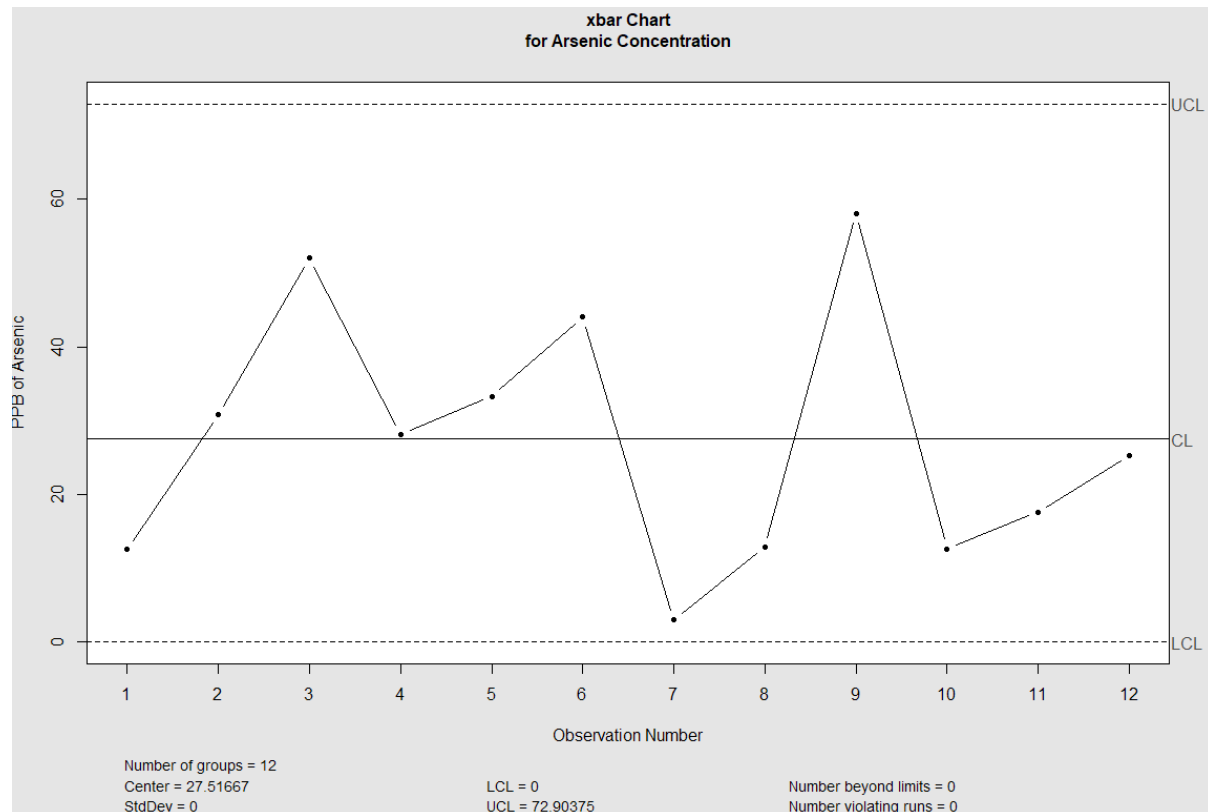
#Has the Upper Value of the prediction interval, established by the
Background data been exceeded by the Compliance sample?
if(all(Arsenic$Compliance[1:8]<UPL)){
  print("The 4 Values from either year fall within the prediction
interval, there is no sign of contamination")
}else{print("The prediction interval has been exceeded, there may be
signs of contamination")}
```

None of the data falls below that value (which the code tests for) and it cannot be concluded, at a 95% probability, that arsenic levels are higher.

Plot the Prediction Interval

The prediction Interval can be plotted using `qcc` with the following code:

```
qcc(Arsenic$Background, type="xbar", sizes=2, limits=c(LPL, UPL), xlab =  
"Observation Number", ylab = "PPB of Arsenic", data.name = "Arsenic  
Concentration")
```



And it can hence be seen that the observed values of arsenic never exceed the upper limit taken from a 95% prediction interval given the nature of the background data.

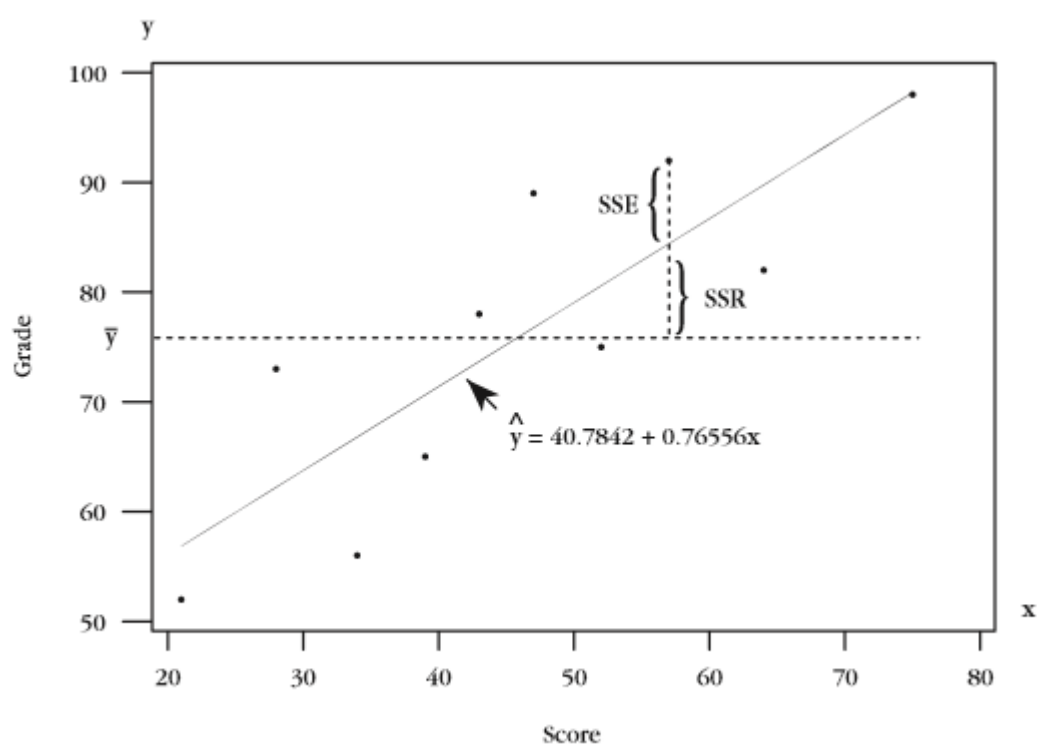
Correlation and Linear Regression

Estimators used in Linear Regression

| | |
|---|--|
| $s = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^n [(x_i - \bar{x})^2]}$ | The sample standard deviation. |
| $S_{xx} = \sum_{i=1}^n [(x_i - \bar{x})^2] = \sum_{i=1}^n [(x_i)^2] - \frac{(\sum_{i=1}^n [x_i])^2}{n}$ | The sum of squared differences from data point to mean for the set of x values |
| $S_{yy} = \sum_{i=1}^n [(y_i - \bar{y})^2] = \sum_{i=1}^n [(y_i)^2] - \frac{(\sum_{i=1}^n [y_i])^2}{n}$ | The sum of squared differences from data point to mean for the set of y values |
| $S_{xy} = \sum_{i=1}^n [(x_i - \bar{x})(y_i - \bar{y})]$ | Is the square value formed via the difference from the means $(x_i - \bar{x})$ and $(y_i - \bar{y})$. |
| $s_{xy} = S_{xy} \times \frac{1}{n-1}$ | Is the covariance. |
| SS = S_{yy} | Is the sum of all squared differences from data point to mean, measured vertically along the y axis. (Equivalently $SST \equiv SSE$). |
| $SSR = \frac{(S_{xy})^2}{S_{xx}}$ | Is the Sum of Squared Residuals, the summed square difference from observation to model. |
| SSE = SS - SSR | Is the Sum of all squared differences from the mean value to a data point that can be attributed to random error |
| MSE = $\frac{SSE}{n-2}$ | Is the average squared distance from a data point to the mean value that can be attributed to error. |

| <u>Source of Variation</u> | <u>Sum of Squares</u> | <u>df</u> | <u>Mean Square</u> | <u>F-stat</u> |
|---|---|-------------|-------------------------------|------------------------------|
| <u>Sum of Squared differences due to Regression (Explained variation of Data as per the Model)</u> | $SSR = \sum_{i=1}^n [(\hat{y}_i - \bar{y})^2]$ | k | $MSR = \frac{SSR}{k}$ | $F_{calc} = \frac{MSR}{MSE}$ |
| <u>Residual or Error (Random Variation that occurs distinct of the model)</u> | $SSE = \sum_{i=1}^n [(y_i - \hat{y})^2]$ | $n - k - 1$ | $MSE = \frac{SSE}{n - k - 1}$ | |
| <u>Total)</u> | $SST = SSE + SSR$ $= s^2(n - 1)$ $= \sum_{i=1}^n [(y_i - \bar{y})^2]$ | $n - 1$ | | |

The Sum of Squares can be visualised for a *Simple Linear Regression* like so:²⁰



²⁰ Mendenhall, W., Beaver, R. and Beaver, B. (2013). *Introduction to probability and statistics*. 14th ed. Boston, MA: Brooks/Cole, p.489 (p. 513 of 752).

Measuring the utility of the Regression

Slope test

Given the line $y = \alpha + \beta x$ we can use b as a sample value for the actual value of β .

If $\beta = 0$ then there must be no linear correlation between the variables.

If $\beta = 0$ is refuted at some confidence level, then there is a β value such that a linear correlation may fit.

The point of this test is that a linear correlation may definitely exist, but there is a shit tonne of random error, so a linear correlation may exist at a high significance level but have a really low coefficient of determination.

TEST OF HYPOTHESIS CONCERNING THE SLOPE OF A LINE

1. Null hypothesis: $H_0 : \beta = \beta_0$
2. Alternative hypothesis:

One-Tailed Test

$$H_a : \beta > \beta_0 \\ \text{(or } \beta < \beta_0 \text{)}$$

Two-Tailed Test

$$H_a : \beta \neq \beta_0$$

3. Test statistic: $t = \frac{b - \beta_0}{\sqrt{\text{MSE}/S_{xx}}}$

When the assumptions given in Section 12.2 are satisfied, the test statistic will have a Student's t distribution with $(n - 2)$ degrees of freedom.

Measuring the strength of the relationship

Correlation Coefficient²¹

The correlation Coefficient r measures the strength of the linear relationship:

$$r = \frac{\frac{S_{xy}}{n-1}}{S_x \times S_y} = \frac{S_{xy}}{S_x \times S_y}$$

It is a ratio comparing the sum of the multiple of the deviations of x and y to the multiple of the standard deviations of x and y .

Coefficient of Determination²²

This value is the proportion of total variation explained by the linear model:

$$\frac{SSR}{SS} = \frac{(S_{xy})^2}{S_{xx} \times S_{yy}} = \left(\frac{S_{xy}}{S_x \times S_y} \right)^2 = r^2$$

Correlation Coefficient in R

The correlation coefficient (r) between two sets of data (X and Y) can be calculated in R like so:

1. Calculate the correlation coefficient (r)
 - a. `Cor(X, Y)`
2. Perform a hypothesis test: $H_0: r = 0$
 - a. `Cor.test(X, Y)`
 - b. There are a few different correlation coefficients, *Pearson's product moment correlation coefficient* is a common one and the one used here.
 - c. This test can be used to confirm whether or not there is a significant linear association between variables.
3. Calculate the covariance ($\frac{S_{xy}}{n-1} = s_{xy}$)
 - a. `Cov(X, Y)`

²¹ Mendenhall, W., Beaver, R. and Beaver, B. (2013) *Introduction to probability and statistics*. p. 103 (127 of 752)

²² Mendenhall, W., Beaver, R. and Beaver, B. (2013) *Introduction to probability and statistics*. p. 498 (522 of 752)

Forming a Linear Regression

Refer to page p. 485 (509 of 752) of the Biometry Textbook.²³

In R it can be useful to get an idea of an object or data frame, in which case the `summary(...)` command can be useful, to get an idea of a data frame or vector the `str(...)` command can be useful.

To fit a linear regression to two sets of variables X and Y.

```
lm(Y ~ X)
```

To plot those variables:

```
plot(X, Y, col="purple", main="Title/Heading")
```

To draw a line through the current plot

```
abline(lm(Y~X), col="blue")
```

or with intercept values a and b :

```
abline(a=,b=, col="blue")
```

The following will provide the p-value associated to the t-statistic corresponding to the intercept and slope respectively:

```
summary(lm(Y ~ X))$coefficients[7]
```

```
summary(lm(Y ~ X))$coefficients[8]
```

²³ Mendenhall, W., Beaver, R. and Beaver, B. (2013) *Introduction to probability and statistics*. p. 498 (522 of 752)

Forecasting predictions from a linear model

A confidence interval of the mean value of the y -value given an x -value (say 22.3) for a linear regression can be created in **R** via the `predict` command:

```
predict(lm(Y~X), data.frame(X=22.3), interval = "confidence")
```

A prediction interval for a y -value given an x -value (say 34.8) for a linear regression can be created in **R** via the `predict` command:

```
predict(lm(Y~X), data.frame(X=34.8), interval = "predict")
```

Note

This can get particularly confusing observe the following points:

1. Variable Names have spaces in them, avoid that
2. Variable Names are really long or ambiguous, avoid that
3. Try to include the axis in the variable name.
4. Don't use the `attach` command, don't use the `dfname$column`, just assign them variables like `"maxtemp_x"`.

Plotting A prediction

A plot can be formed of a prediction using the following **R** code:

```
Mytestdist <- rnorm(300,0,1)
hist(Mytestdist, prob=TRUE, lwd=2) #Prob chooses probability of frequency for the
histogram.
#curve(dnorm(x, mean=mean(Mytestdist), sd=sd(Mytestdist)), add=TRUE) #Draws the
actual density function
lines(density(Mytestdist), col='purple', lwd=2) #Draws the observed density
function
abline(v=-1.96, col='red', lwd=3)
abline(v=1.96, col='red', lwd=3)
```

The Mathematics

A $(1 - \alpha)100\%$ confidence interval for the mean value of \hat{y} :²⁴

$$\hat{y} \pm t_{\alpha/2, n-2} \sqrt{MSE \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{\sum (x_i - \bar{x})^2}}}$$

I don't know about finding a prediction interval.

²⁴ PennState: Eberly College of Science. (2017). *A Confidence Interval for the Mean of Y*. [online] Available at: <https://onlinecourses.science.psu.edu/stat414/node/297> [Accessed 9 Aug. 2017].

Regression Diagnostics

A residual is the error from an observed point to the predicted point, so if e is the error or residual, \hat{y} is the fitted point (i.e. the line) and y is the observed point:

$$e = y - \hat{y}$$

Usually i denotes the number of the observation:

$$e_i = y_i - \hat{y}_i$$

In the lecture notes the standardised residual was used, this value is:²⁵

$$std(e_i) = \frac{e_i}{s_e}$$

Where:

$std(e_i)$ Is the standard residual

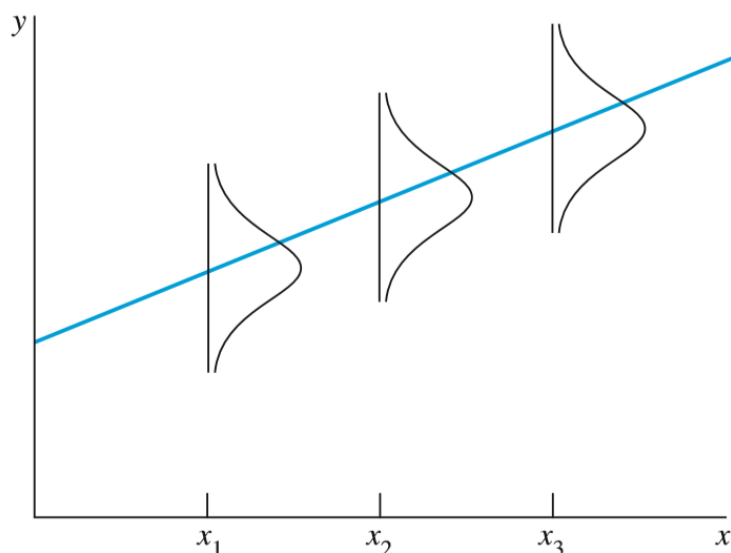
e_i Is the residual for the i^{th} data point

s_e Is the standard deviation of the residual.

What we are testing

An assumption made when forming a linear regression is that the error (ε) in the model ($y = \beta_1 \cdot x + \beta_0 + \varepsilon$) is normally distributed (and hence has a constant variation).

If this assumption can be rejected, then the linear regression may not be an appropriate model:²⁶



²⁵ http://www.stat.ucla.edu/~nchristo/introeconometrics/introecon_compute_sres_hat.pdf

²⁶ Mendenhall, W., Beaver, R. and Beaver, B. (2013). Introduction to probability and statistics. 14th ed. Boston, MA: Brooks/Cole, p.485 (p. 509 of 752).

The Easy Way

All four of the Regression diagnostics will be generated and have lines drawn through them by simply running the following:

```
plot(lm(Y~X))
```

Individually Creating the Residual Diagnostics

The actual method to create the diagnostics is:

Residuals (i.e. the random error) should follow a normal distribution with a mean value of 0, this assumption can be tested by various plots:

1. Plotting residuals (e) against fitted values (\hat{y})
 - a. There should be no pattern
2. Plotting the square root of the absolute value of residuals $\sqrt{|e|}$ against the fitted values (\hat{y})
 - a. Same as above but tests more so the assumption of constant variance of the residual
3. Normal Q-Q plot (residuals plotted against fitted values)
 - a. Tests the assumption that the errors come from a normal distribution
4. Plotting residuals against Cook's Distance
 - a. Cook's distance is a measure of how much a data point affects the slope of the regression, it is the leverage of that data point.

These plots are such that the residual is always put along the y -axis.

Creating Residual Plots in R

1. Residuals can be plotted in R like any other variable
 - a. Residuals of a linear model can be found as such:
 - i. `resid(lm(Y ~ X))`
 - b. The y-values of the linear model are referred to as the fitted values and can be retrieved with:
 - i. `fitted(lm(Y~X))`
 - c. A plot with the residuals on the y-axis and the y-values from the linear model on the x-axis can be formed like so:
 - i. `plot(fitted(lm(Y~X)), resid(lm(Y ~ X)))`
 - d. The residuals should be normally distributed about 0, hence a line can be drawn in the plot like so:
 - i. `abline(0,0)`
2. A plot of the square root of residuals could be achieved as above where the root of the residuals could be found like so:
 - i. `sqrt(abs(resid(lm(Y ~ X))))`
 - ii. The standard residual, if needed can be calculated:
 1. `rstandard(lm(Y~X))`
3. A normal Q-Q plot may can be generated with the `qqnorm` function like so:
 - a. With the residuals corresponding to the Y-axis relative to a Standard Normal Distribution:
 - i. `qqplot(resid(lm(Y~X)))`
 - b. A line can also be added to the current plot with:
 - i. `qqline(resid(lm(Y~X)))`
4. Leverage refers to the amount a data point affects the slope of the regression, we are using *Cook's Distance* to calculate the slope, this can be calculated as such:
 - a. `cooks.distance(lm(Y~X))`

Exercise 4.1; Example 4.1

Exercise

Figure 1.7 from lecture 1 appears to have a strong correlation between the two random variables, test this hypothesis.

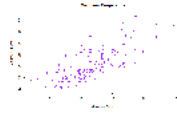


Figure 1: Scatter Plot Comparing Maximum and Minimum Temperatures, Figure 1.7 from from the Lecture 1 Materials,

Solution

To test whether they are correlated values the *Pearson's product moment correlation coefficient* (r) can be tested.

$$r = \frac{s_{xy}}{s_x s_y} \quad (1)$$

In R the test can be utilised as such:

```
#Perform the Test
cor.test('Max Temperature', 'Min Temperature')

> cor.test('Max Temperature', 'Min Temperature')

Results of Hypothesis Test
-----

Null Hypothesis:          correlation = 0

Alternative Hypothesis:    True correlation is not equal to 0

Test Name:                 Pearson's product-moment correlation

Estimated Parameter(s):    cor = 0.7498076

Data:                      Max Temperature and Min Temperature

Test Statistic:            t = 21.59091

Test Statistic Parameter:  df = 363

P-value:                   4.318158e-67

95% Confidence Interval:    LCL = 0.7011211
                           UCL = 0.7915352
```

Given the low p-value, it can be determined at a high confidence level that there is a linear association between the variables.

Rolling into a Single Command

```
> if(
+correlation_test <- cor.test('Max Temperature', 'Min Temperature')$p.value < 1-sig_level/100
+ ){
+ print("There is a significant linear association between the variables")
+ }

[1] "There is a significant linear association between the variables"
```

Wk. 5 Notes – Multiple Linear Regression

Wk. 5 Material (14 August) | Due Wk. 6 Prac (21 August)

Contents

| | |
|---|----|
| Ordinary Least Squares Regression..... | 41 |
| Creating a Multiple Linear Regression Model in R | 41 |
| Forecasting Predictions from the linear model | 42 |
| Note | 42 |
| Plotting A prediction | 42 |
| Analysis of Variance <i>F</i> -Test | 43 |
| To determine the <i>F</i> -Statistic in R | 43 |
| Categorical (Discrete) Variables..... | 45 |
| Residual Diagnostics | 46 |
| Variable Screening | 46 |
| Backward Elimination Method | 46 |
| Forward Elimination..... | 46 |
| AIC..... | 46 |
| Testing the Summed Square Formulas | 47 |
| Observations made playing in R..... | 48 |
| Sample Size and Confidence Interval observation..... | 48 |
| Larger Attempt at it. | 49 |
| Code from Investigation..... | 51 |

Ordinary Least Squares Regression

This is a process of minimizing the error between observations and a multiple linear model.

There are some limitations to the *Ordinary Least Squares* method:

- More observations than x -variables
- Only one y value can be modelled

Ordinary least squares requires the errors ($\varepsilon_1, \varepsilon_2, \varepsilon_3, \dots \varepsilon_n$) be *independent of one another and distributed identically* (i.i.d.).

Creating a Multiple Linear Regression Model in R

A multiple linear regression for as many predictors as desired can be created via the command:

```
Y <- Response Variable
X1 <- Predictor #1
X2 <- Predictor #2
X3 <- Predictor #2

Lm (Y~X1+X2+x3...)
```

Forecasting Predictions from the linear model

A confidence interval for the mean value of the y-value given for predictor x-values (say $X1 = 10$, $X2 = 20$, $X3 = 30$) for a linear regression can be created in **R** via the `predict` command:

```
predict(lm(Y~X1+X2+X3...), data.frame(X1=10, X2=20, X3=30...),  
interval = 'confidence', level=0.95)
```

A prediction interval for a single observation or y-value given predictor x-values (say $X1 = 10$, $X2 = 20$, $X3 = 30$) for a linear regression can be created in **R** via the `predict` command:

```
predict(lm(Y~X1+X2+X3...), data.frame(X1=10, X2=20, X3=30...),  
interval = 'predict', level=0.95)
```

I'm not totally sure how to create a prediction interval for multiple observations from a linear regression right now.

Note

This can get particularly confusing, observe the following points:

5. Variable Names that have spaces in them can break the command, avoid that
6. Variable Names are really long or ambiguous can lead to problems, avoid that
7. Try to include the axis (e.g. y, x1, x2 etc.) in the variable name.
 - a. Number the predictor values so you don't get confused later
 - b. The order of the predictor values in the data frame doesn't matter because they are labelled (hence why a data frame is used rather than a mere vector).
 - c. The data frame can ONLY have **1** row.
8. Don't use the `attach` command, don't use the the format: `dfname$column`, just assign them variables like "maxtemp_x1".

Plotting A prediction

A plot can be formed of a prediction using the following **R** code:

```
Mytestdist <- rnorm(300,0,1)  
hist(Mytestdist, prob=TRUE, lwd=2) #Prob chooses probability of frequency for the  
histogram.  
#curve(dnorm(x, mean=mean(Mytestdist), sd=sd(Mytestdist)), add=TRUE) #Draws the  
actual density function  
lines(density(Mytestdist), col='purple', lwd=2) #Draws the observed density  
function  
abline(v=-1.96, col='red', lwd=3)  
abline(v=1.96, col='red', lwd=3)
```

Analysis of Variance F-Test

In *simple linear Regression* the significance of the regression could be determined by testing whether the slope β was more than 0 via a *t-statistic*, the *t-statistic* can only compare two values and for *Simple Linear Regression* is equivalent to an *F-statistic*.

In Multiple Linear Regression the *t-statistic* is not equal to the *F-statistic*, and to determine whether atleast one of the slopes is significant an *Analysis of Variation (ANOVA)* testing strategy must be employed to get an *F – Statistic*.

For a Multiple Linear Regression of the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \dots \beta_k x_{k+1}$$

A hypothesis test of significance would be:

$$H_0: \beta_1 = \beta_2 = \beta_3 \dots \beta_k = 0$$

$$H_a: \text{Atleast one slope } \beta \text{ value is non-zero}$$

| <u>Source of Variation</u> | <u>Sum of Squares</u> | <u>df</u> | <u>Mean Square</u> | <u>F-stat</u> |
|---|--|-------------|-------------------------------|------------------------------|
| <u>Sum of Squared differences due to Regression (Explained variation of Data as per the Model)</u> | $SSR = \sum_{i=1}^n [(\hat{y}_i - \bar{y})^2]$ | k | $MSR = \frac{SSR}{k}$ | $F_{Calc} = \frac{MSR}{MSE}$ |
| <u>Residual or Error (Random Variation that occurs distinct of the model)</u> | $SSE = \sum_{i=1}^n [(y_i - \hat{y})^2]$ | $n - k - 1$ | $MSE = \frac{SSE}{n - k - 1}$ | |
| <u>Total)</u> | $\begin{aligned} SST &= SSE + SSR \\ &= s^2(n - 1) \\ &= \sum_{i=1}^n [(y_i - \bar{y})^2] \end{aligned}$ | $n - 1$ | | |

To determine the F-Statistic in **R**

The following code will output the F-Statistic in R:

```
Y <- Response Variable
X1 <- Predictor #1
X2 <- Predictor #2
X3 <- Predictor #2

summary.lm <- summary(lm(Y~X1+X2+X3...))
summary.lm$fstatistic[1]
```

The relevant variables in linear regression (simple and multiple) are described below)

| | |
|---|--|
| $s = \sqrt{\frac{1}{n-1} \times \sum_{i=1}^n [(x_i - \bar{x})^2]}$ | The sample standard deviation. |
| $S_{xx} = \sum_{i=1}^n [(x_i - \bar{x})^2] = \sum_{i=1}^n [(x_i)^2] - \frac{(\sum_{i=1}^n [x_i])^2}{n}$ | The sum of squared differences from data point to mean for the set of x values |
| $S_{yy} = \sum_{i=1}^n [(y_i - \bar{y})^2] = \sum_{i=1}^n [(y_i)^2] - \frac{(\sum_{i=1}^n [y_i])^2}{n}$ | The sum of squared differences from data point to mean for the set of y values |
| $S_{xy} = \sum_{i=1}^n [(x_i - \bar{x})(y_i - \bar{y})]$ | Is the square value formed via the difference from the means $(x_i - \bar{x})$ and $(y_i - \bar{y})$. |
| $s_{xy} = S_{xy} \times \frac{1}{n-1}$ | Is the covariance. |
| SS = S_{yy} | Is the sum of all squared differences from data point to mean, measured vertically along the y axis. (Equivalently $SST \equiv SSE$). |
| $SSR = \frac{(S_{xy})^2}{S_{xx}}$ | Is the Sum of Squared Residuals, the summed square difference from observation to model. |
| SSE = SS - SSR | Is the Sum of all squared differences from the mean value to a data point that can be attributed to random error |
| MSE = $\frac{SSE}{n-2}$ | Is the average squared distance from a data point to the mean value that can be attributed to error. |

Categorical (Discrete) Variables

In order to deal with discrete variables **R** uses a data type called factors, to create factors the `factor()` command is used, within it a vector containing factor levels must be enclosed, e.g.:

```
factor(c("Male", "Female"))
```

Would create a factor containing the factor levels (i.e. categories) of Male/Female.

When performing a multiple linear regression with categorical data, the categorical data will be treated as a variable that can be 1 or 0, basically, choosing between different categorical variables is choosing a different intercept (adding a constant value).

This is NOT even close to being as accurate as a linear regression separately on the data within that category, so, unless there is a good reason (e.g. the different categories would have a trend with the same rate but a different intercept perhaps the ambient temperature would fit this description), don't combine a regression with categorical data.

e.g.:

$$Response = -74 - 3.109 \cdot X_{wind} - 1.875 \cdot X_{temp} - 14.76 \cdot JUNE - 8.749 \cdot JULY - 4.197 \cdot AUG$$

Residual Diagnostics

Recall that a residual is the difference between the observed value (y) and the fitted value (i.e. the regression value) (\hat{y}):

$$e_i = y - \hat{y}$$

Diagnostic models for a linear regression can be created by using the `plot()` command on a linear model like so:

```
Y <- Response Variable
X1 <- Predictor #1
X2 <- Predictor #2
X3 <- Predictor #2

lm(Y~X1+X2+x3...)           #Create the Linear Model

plot( lm(Y~X1+X2+x3...) )    #Create Plots for the Regression Diagnostics
```

For a more detailed explanation of the residual diagnostics, refer to the Wk. 4 Notes.

Variable Screening

In an effort to simplify the equation a statistically insignificant coefficient (i.e. $t < t_{crit}$) may be removed.

This is useful where a host of possible predictors are available and relevant ones need to be found, different methods include *Bayesian information criterion (BIC)*, *Akaike information criterion (AIC)*, R^2 etc.²⁷

Backward Elimination Method

1. Perform a linear regression
2. Remove the coefficient with the lowest p-value IF it is greater than the significance level (e.g. 5%)
3. Perform another linear regression and repeat step 2 until there are no coefficients to remove.

Forward Elimination

1. Perform a linear regression
2. Include only the coefficients with the highest p-value.

AIC

The AIC Method can be used in R with the `step()` command.

²⁷ Lindquist, M. (2017). *Variable Selection*. [online] Columbia University Department of Statistics. Available at: <http://www.stat.columbia.edu/~martin/W2024/R10.pdf> [Accessed 11 Aug. 2017].m

Testing the Summed Square Formulas

The SSR, SST and SSE formulas can be tested and confirmed inductively in R by using the code:

```
#If we take SSR and SSE as they are described in my notes, will they add
to SST?

regression_error <- rnorm(n = 3000, mean=0, sd=300)
x_regression <- seq(1:30000)
y_regression <- x_regression*3+1+regression_error
line_regression <- lm(y_regression~x_regression)

plot(x = x_regression, y = y_regression)
abline(lm(y_regression~x_regression))

intercept <- line_regression$coefficients[1]
slope <- line_regression$coefficients[2]

anova_regression <- anova(line_regression)
SSR_R <- anova_regression$`Sum Sq`[1]
SSE_R <- anova_regression$`Sum Sq`[2]

SSR= sum(((x_regression*slope+intercept)-mean(y_regression))^2 )
if(round(SSR,3)==round(SSR_R,3)){print("SSR is Correct")}else(print("SSR
is Wrong"))

SSE=sum((y_regression-(x_regression*slope+intercept))^2 )
if(round(SSE,3)==round(SSE_R,3)){print("SSE is Correct")}else(print("SSE
is Wrong"))

SST_form=sum((y_regression-mean(y_regression))^2)
SST_add=SSE+SSR

if(round(SST_form,1)==round(SST_add,1)){print("The Formulas are the
same")}else(print("They are different"))
```

Observations made playing in R

If the random error that might affect a regression is known how much would that affect the r^2 value?

e.g. measuring a person's waist size would have a standard deviation of 1-5 inches.

To find out I used R to create a linear model such that the response variable was 3 the predictor variable plus generated random error and ran that in a loop collecting the r^2 value

Using R to produce sample sizes of 1000 the following was observed:

| Sample Size | Standard Deviation | $\frac{n}{\sigma}$ | R^2 (99.99% conf. Interval) | Distribution of R^2 values |
|-------------|--------------------|--------------------|-------------------------------|-------------------------------|
| 200 | 200 | 1 | 0.430-0.432 | Normal |
| 10 | 10 | 1 | 0.996-0.997 | Normal |
| 50 | 10 | 5 | 0.99671-0.99678 | Normal |
| 100 | 10 | 10 | 0.996 | Normal Left Skewed |
| 1000 | 10 | 100 | 0.999 | Normal |
| 10 | 50 | 0.2 | 0.12 | Chi |
| 50 | 50 | 1 | 0.12-0.14 | Normal |
| 100 | 50 | 2 | 0.751-0.75511 | Normal |
| 1000 | 50 | 20 | 0.996 | Normal |
| 10 | 100 | 0.1 | 0.11-0.12 | Chi (like $y = \frac{1}{x}$) |
| 50 | 100 | 0.5 | 0.16-0.17 | Normal Right Skewed |
| 100 | 100 | 1 | 0.43-0.43 | Normal Left Skewed |
| 1000 | 100 | 10 | 0.98 | Normal Slightly Left Skewed |
| 10 | 1000 | 0.001 | 0.10-0.11 | Chi |
| 50 | 1000 | 0.005 | 0.020-0.024 | Chi |
| 100 | 1000 | 0.1 | 0.01 | Chi |
| 1000 | 1000 | 1 | 0.0428-0.043 | Normal |

So for measuring say waist size/body weight, 30 observations would be required given the standard deviation of the error was 4 (kg or inch), to be able to form a linear regression with a high $r^2 = 97\%$ value assuming the variation only results from random error given that standard deviation.

10 samples would only provide $r^2 = 0.84$

Sample Size and Confidence Interval observation

It was observed that the distribution of r^2 values that resulted from different regressions of random x/y sets was usually normally distributed.

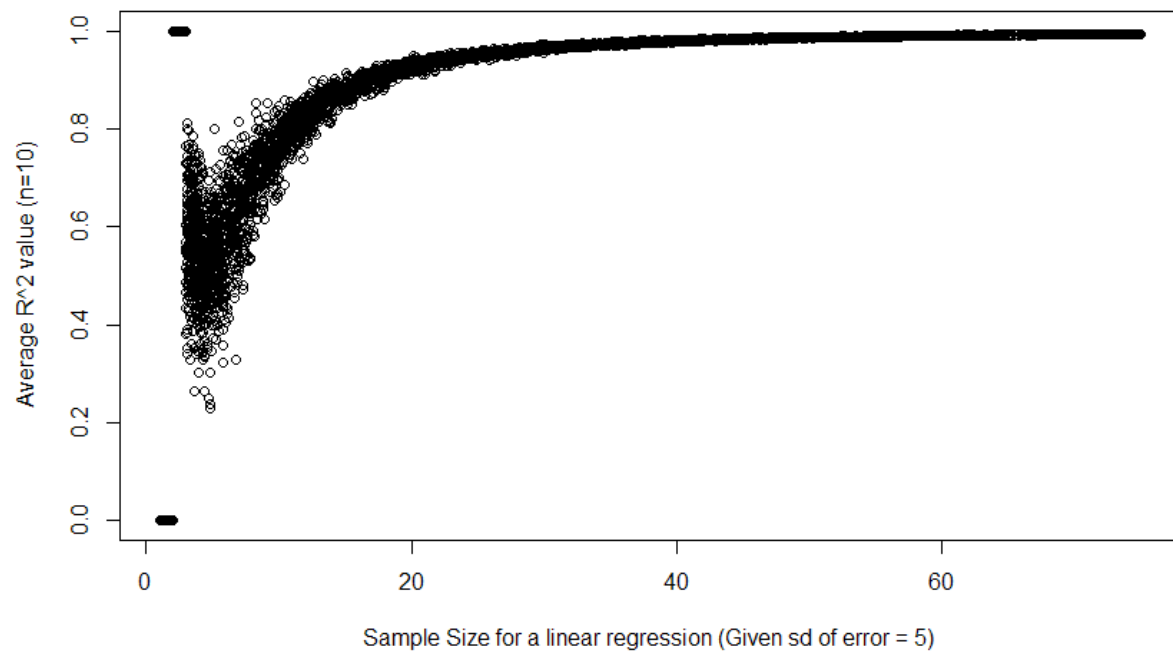
If the sample size is a fraction of the standard distribution, the distribution of R^2 values isn't just a normal distribution with a mean of 0 without negative values, it actually takes on a chi distribution.

At 50 samples the 95% confidence interval was 0.41-0.45

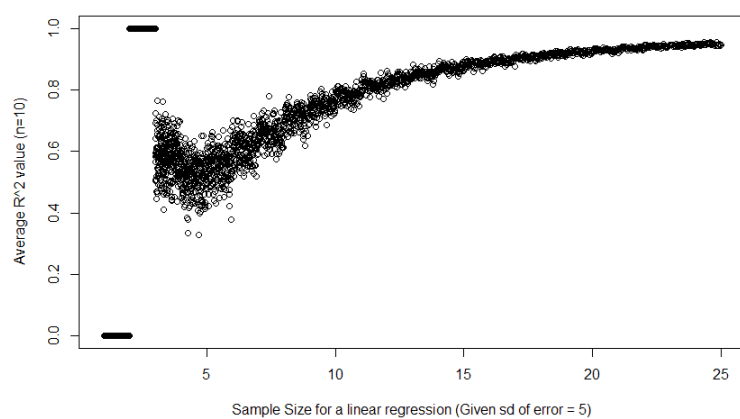
At 45,000 samples the 95% confidence interval was 0.431-0.431

Larger Attempt at it.

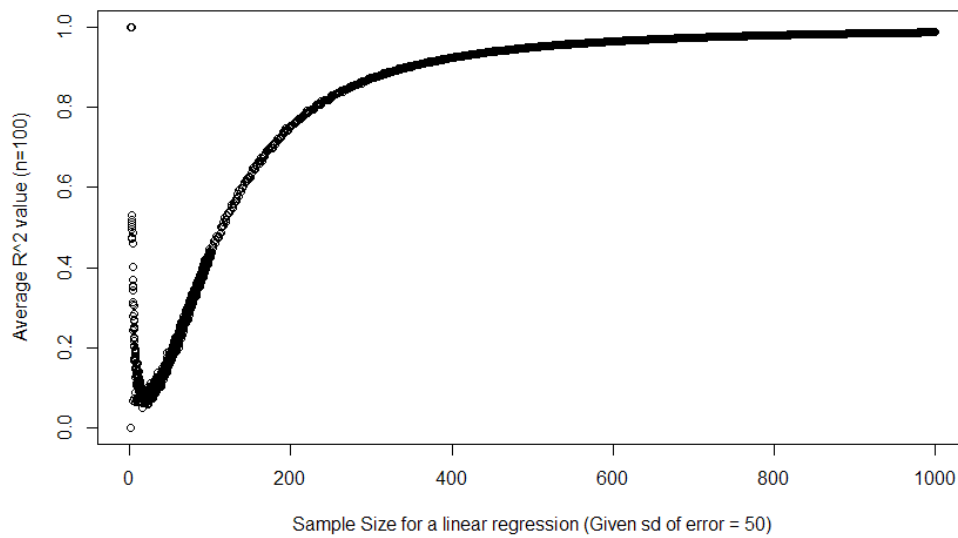
With an error standard deviation of 5, having R generate 10 regressions, collect the average R^2 value, increase the sample size and then repeat, we can get the following graphs:



Investigate 0-75 and set an n -value proportional to the sample size, there is no change after 200, so for standard deviation of 5, nothing changes after 200.



For a Standard Deviation of 100(NOT 50):



OK, so what I've gotta do, is I've gotta make a 3-dimensional model of the distribution of standard deviation and sample sizes relative to R^2 values.

Code from Investigation

```
#OK, so I want to create a multiple linear regression, that will predict, from the
sample size and the std dev.
# what the r^2 value will be???

#Experimental design, create a data frame of vector so Excel can be used for
polynomial fitting
#I want to compare 1000 to 1000

n <- 1
means_sample <- c()
sample_size <- c()
max_n <- 1000
while(n<max_n){

R2vals <- c()
#n <- 50
sd <- 5

  while(length(R2vals)<10){

    regressionerror <- rnorm(n = n, mean = 0, sd = sd)
    xtest <- seq(1:n)
    ytest <- xtest*3+regressionerror

    sumlm <- summary(lm(ytest~xtest))
    r2 <- sumlm$r.squared
    R2vals <- c(R2vals, r2)

    #print(length(R2vals))
    #hist(R2vals)

  }

  if((n*100/n) %% 5 == 0){    print(n*100/max_n)  }

  mean_R2 <- mean(R2vals)
  n_changing <- n

  means_sample <- c(means_sample, mean_R2)
  sample_size <- c(sample_size, n)

  if(n<100){n <- n+0.1}else{n <- n+1}

}

plot(x = sample_size,y =means_sample, xlab = "Sample Size for a linear regression
(Given sd of error = 5)", ylab = "Average R^2 value (n=10)" )
```

Exercise 5.1 (example 5.1)

With the supplied data set, create a multiple linear regression model for the level of atmospheric ozone using the predictors, solar, wind and temperature.

Solution

Multiple Linear Regression

First create the linear model in **R**, which provides the *multiple linear regression* is:

$$y_{\text{ozone}} = -64.342 + 0.060 \cdot X_{\text{solar}} - 3.33 \cdot X_{\text{wind}} + 1.65 \cdot X_{\text{temp}}$$

```
Ozone_multiple.lm <-  
lm(response_ozone_y~predict_solar_x+predict_wind_x+predict_temp_x)  
  
#Summarise the Linear Model  
summary(Ozone_multiple.lm)  
...  
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept)   -64.34208    23.05472   -2.791   0.00623 **  
predict_solar_x    0.05982     0.02319    2.580   0.01124 *  
predict_wind_x   -3.33359     0.65441   -5.094  1.52e-06 ***  
predict_temp_x    1.65209     0.25353    6.516  2.42e-09 ***
```

Significance

The F-statistic of 54.83 corresponds to a p-value less than 0.01, hence the linear regression is highly significant.

```
summary_ozone.lm <- summary(Ozone_multiple.lm)  
summary_ozone.lm$fstatistic  
      value  
54.83366
```

Forecasting

For values of solar, wind and temp a confidence interval for the mean ozone value and a prediction interval for the observed ozone level can be formed.

| Solar | Wind | Temperature |
|-------|------|-------------|
| 180 | 10 | 60 |

Confidence Interval

A 95% confidence interval for the mean value of the population ozone level, given the predictors, is:

[30.69, 40.35]

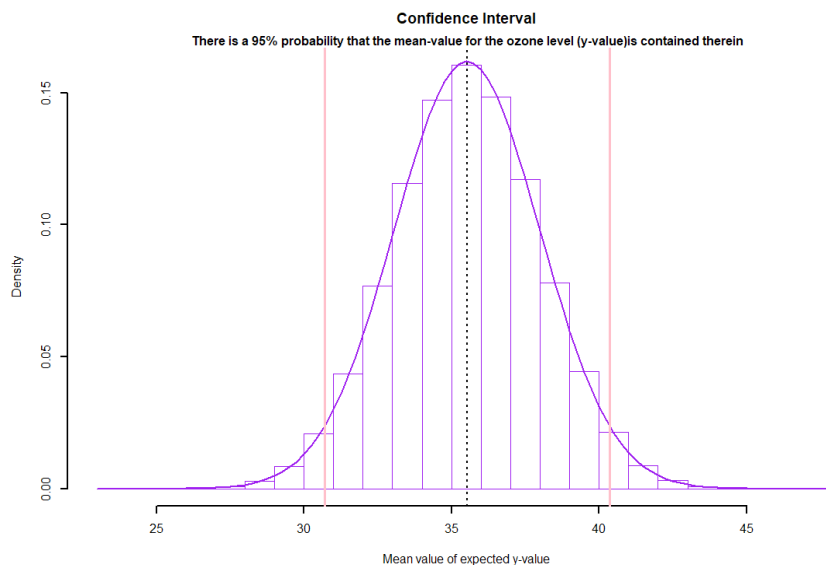
This can be solved in **R** thusly:

```
# Forecast Data from the Linear Model -----
#Forecast Values
fcastval <- data.frame(predict_solar_x1=184, predict_temp_x3=78, predict_wind_x2=12)
alphaint <- 0.95

# Confidence Interval -----
predict(Ozone_multiple.lm, fcastval, level=alphaint, interval='confidence')

fit      lwr      upr
1 35.52506 30.69638 40.35374
```

Graphical depiction of Interval



Prediction Interval

A 95% prediction interval for the value of the ozone level, given the predictors, is:

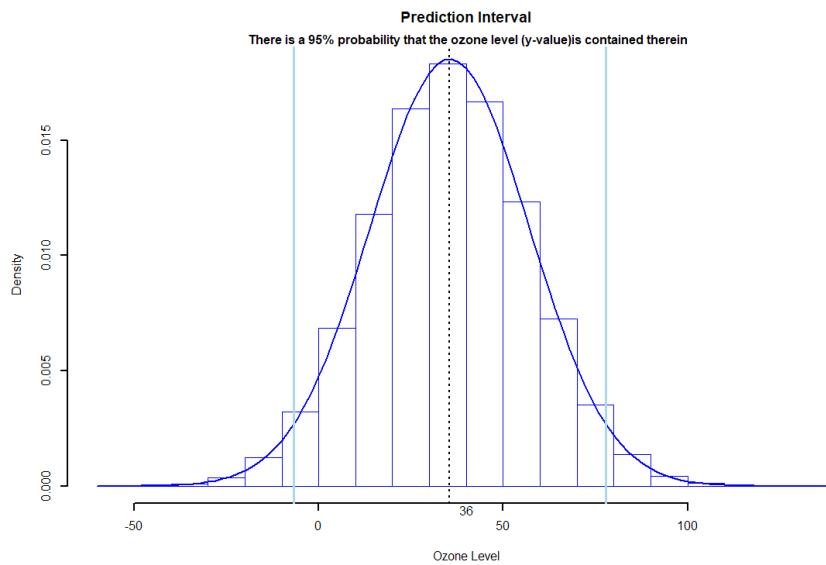
[−6.74, 77.79]

This can be solved in **R** thusly:

```
# Prediction Interval -----
predict(Ozone_multiple.lm, fcastval, level=alpha, interval='predict')

      fit      lwr      upr
1 35.52506 -6.740045 77.79017
```

Graphical depiction of Interval



The following Script will create the Histograms in **R**:

Observe that a corresponding Normal distribution can be generated with the same mean μ value as the linear model and by solving a standard deviation that corresponds to the critical prediction points

```
alphaint <- 0.95

upr_pred <- predict(ozone_mult.lm,
                    fcastvals,
                    level=alphaint,
                    interval='predict')[3]
lwr_pred <- predict(ozone_mult.lm,
                    fcastvals,
                    level=alphaint,
                    interval='predict')[2]

# Plot the Predidence Interval -----
#The histogram of possible values, corresponds to a
#normal distribution along the y-axis
# This normal distribution will have a mean value of
#the fitted y-value The standard deviation will
#correspond to a normal distribution where the Z-value occurs at 41.9 rather
than 1.96

mean_pred_y <- predict(ozone_mult.lm,
                      fcastvals,
                      level=alphaint,
                      interval='predict')[1]
sd_pred_y <- -(upr_pred-mean_pred_y)/qnorm(0.025, mean=0, sd=1)

possiblyvals_pred <- rnorm(n=100000, mean=mean_pred_y, sd=sd_pred_y)
hist(possiblyvals_pred, prob=TRUE, lwd=2,
     main = "Prediction Interval",
     xlab = "Ozone Level", border = "blue"
    ) #Prob chooses probability of frequency for the histogram.
mtext(" There is a 95% probability that
      the ozone level (y-value)is contained therein", font = 2)
curve(dnorm(x, mean=mean(mean_pred_y),
            sd=sd(possiblyvals_pred)),
      add=TRUE, col="blue", lwd=2
    ) #Draws the actual density function
#lines(density(possiblyvals_pred), col='purple', lwd=2) #Draws the observed
density function
abline(v=upr_pred, col='lightblue', lwd=3)
abline(v=lwr_pred, col='lightblue', lwd=3)
abline(v=mean_pred_y, lwd=2, lty='dotted')
mtext(print(round(mean_pred_y),2), 1:0, font = 1 )
```

```
# Preamble -----

# Import relevant Packages
#install.packages("EnvStats")
library(EnvStats)

#install.packages("plotly")
library(plotly)

#Import relvant Data Sets
#So we don't have to bother importing data sets, just create the vector in the
script.
response_ozone_y <- c(41, 36, 12, ...
predict_solar_x1 <- c(190, 118, 149,...
predict_wind_x2 <- c(7.4, 8.0, 12.6,...
predict_temp_x3 <- c(67, 72, 74, 62, ...

ozone.csv <- data.frame("Observation Number"=seq(1:111),"Solar"=predict_solar_x1,
"Wind"=predict_wind_x2, "Temperature"=predict_temp_x3, "Ozone"=response_ozone_y )

#Written by Ryan G. - 17805315 - 7 August 2017 (Wk. 4)

# Create the Linear Model (With Multiple Predictors) -----

Ozone_multiple.lm <-
lm(response_ozone_y~predict_solar_x1+predict_wind_x2+predict_temp_x3)

# Summarise the Linear Model -----
summary(Ozone_multiple.lm)

# Print the F-statistic -----
summary_ozone.lm <- summary(Ozone_multiple.lm)
print(summary_ozone.lm$fstatistic[1])

# Forecast Data from the Linear Model -----
#Forecast Values
fcastval <- data.frame(predict_solar_x1=184, predict_temp_x3=78,
predict_wind_x2=12)
alphaint <- 0.95

# Confidence Interval -----
predict(Ozone_multiple.lm, fcastval, level=alphaint, interval='confidence')

# Prediction Interval -----
predict(Ozone_multiple.lm, fcastval, level=alphaint, interval='predict')
```


Time Series Analysis

Week 6 Material | 21 August 2017 | Due: 28 August 2017

Contents

| | |
|--|-------------------------------------|
| Introduction | 58 |
| Plotting Time Series Data | 58 |
| Additive and Multiplicative Models | 59 |
| Additive Model | 59 |
| Multiplicative Model | 59 |
| Parts of time series | 60 |
| Deterministic | 60 |
| Stochastic | 60 |
| Functions in Time Series Analysis | 61 |
| General Parameters | 61 |
| Mean and Variance function | 61 |
| Covariance | 61 |
| Auto-covariance function (ACVF) | 61 |
| Autocorrelation function (ACF) | 62 |
| Purely random Process (White noise) | 63 |
| Simulating in R | 63 |
| Examples | 63 |
| Random Walk Process | 64 |
| Mean Value | 65 |
| Variance | 65 |
| Autocovariance | 65 |
| Autocorrelation | 65 |
| Drift | 65 |
| Simulating in R | 65 |
| Moving Average Process | 66 |
| Variance of Z_t | 66 |
| Autocovariance | 66 |
| Autocorrelation | 66 |
| Example Problem from Lecture Notes | 66 |
| Solving the Mean value function | 67 |
| Solving the Mean Value function in R | Error! Bookmark not defined. |
| Stationary Time Series | 68 |
| Example | 68 |
| Appendix | 69 |
| Plot of Various Time Series Functions | 69 |
| Comparison of White Noise and Random Walk Data | 70 |

Introduction

Classical Statistical methods require the following assumptions:

1. Observations are independent
2. Observations are from the same population (i.e. probability distribution)
3. If the Statistical Method is *parametric*:
 - The data follows a particular probability distribution (e.g. Normal or Chi)

Where observations are non-independent due to the fact that they were collected sequentially over time classical statistical methods may not be appropriate and Time Series Analysis comes into play.

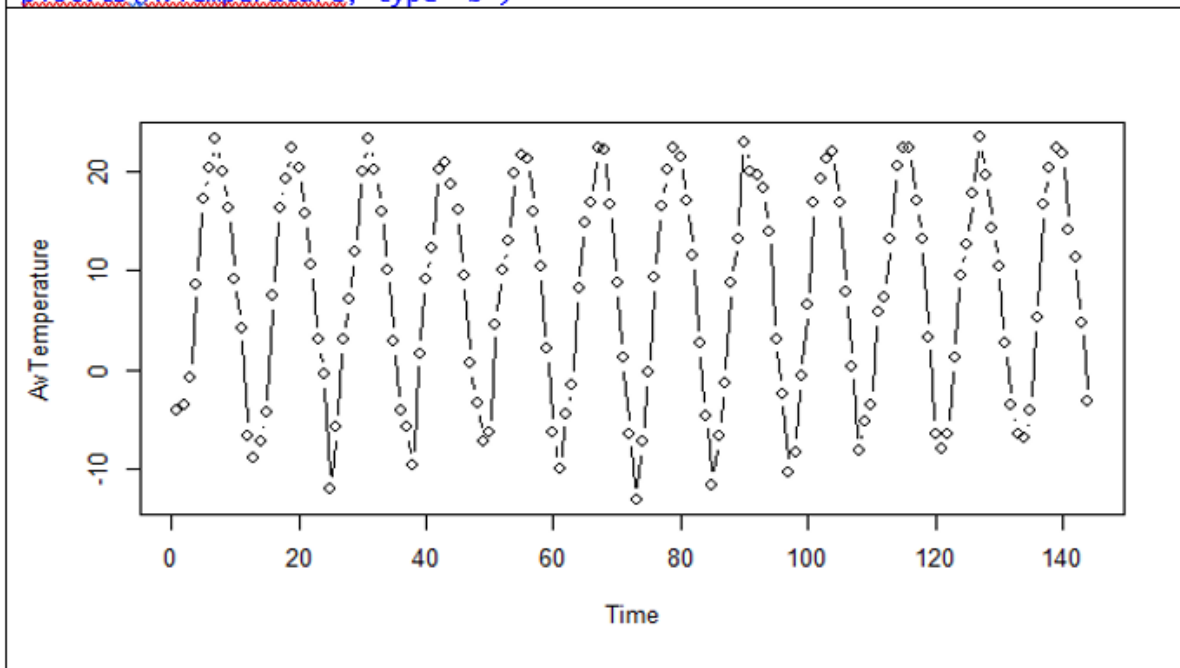
Imagine, rolling a dice, if at 10am a dice was rolled and the result was a 6, the probability of rolling a 6 at 11 am would not be affected. However if at 10am the thermometer read 35 °C it would be quite unlikely for the thermometer to read 5 °C at 11am, due merely to the fact of the past observation. This is why Time Series Analysis is useful.

Where observations are non-independent due to the fact that they were collected in close proximity (space) techniques known as spatial statistics are used.

Plotting Time Series Data

Time series data is an observation (y_t) plotted against the date (x), e.g.:

```
par(mfrow=c(1,1),cex=0.8)  
plot.ts(AvTemperature, type='b')
```

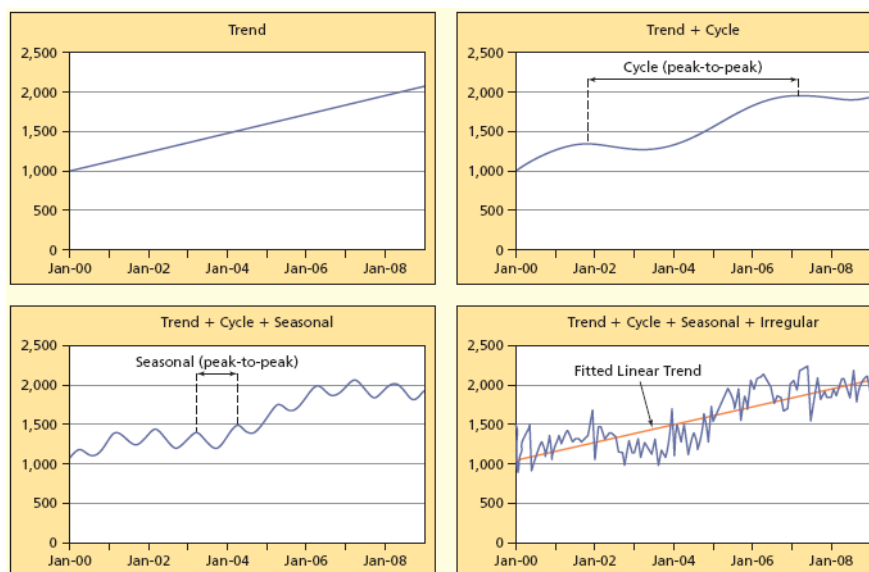


Additive and Multiplicative Models

Time series can be broken up into four types of components:

- Trend (T)
- Cycle (C)
- Seasonal (S)
- Irregular (I)

These are best illustrated by way of a diagram:



It is assumed that these components interact in either an additive or multiplicative fashion

Additive Model

$$Y = T + C + S + Z$$

This is used where data is of similar magnitude (trend-free or over a short run) with constant absolute growth or decline

Multiplicative Model

$$Y = T \times C \times S \times Z$$

This is used where data is of increasing or decreasing magnitude (long run or trended data) with constant percent growth or decline.

The multiplicative model becomes additive as logarithms are taken (of non-negative data).

Parts of time series

Time series have two major parts, a **deterministic** part and a **stochastic** part.

Deterministic

The **deterministic** part may consist of various effects, such as,

- T_t Long-term Trend-The general movement over all years;
- C_t Cyclical effect – repetitive up and down movements about a trend that covers several years; Cyclical trends will go up and down randomly.
- S_t Seasonal effect – repetitive cyclical pattern within a year (or a week or other smaller time period), Seasonal effects will have a constant pattern.

Stochastic

The **stochastic** component is the random variation Z_t .

- Z_t Essentially this is just random error and fluctuation, irregular difficult to explain movement of data. (Also denoted by I_t).

If data is *independent and identically distributed* (i.i.d.) then the additive time series model becomes a multiple linear regression (from wk. 5 material; how to use a multiple linear regression to model this is discussed in Wk. 7).

Generally the data is a sequence of successive dependent observations and special time series models are used (discussed in Wk. 8).

Functions in Time Series Analysis

General Parameters

Mean and Variance function

The **mean** and **variance** function of a time series $\{Z_t\}$ is defined by:

$$\mu_t = E(Z_t) \mid \sigma_t^2 = \text{var}(Z_t)$$

$E(X)$ refers to the expected value of random variable x^{28} , these are both functions of t .

Covariance

In the previous notes Covariance was defined as:

$$\text{Cov}(X, Y) = \frac{1}{(n-1)} \sum_{i=1}^n [(x_i - \bar{x})(y_i - \bar{y})]$$

Covariance can also be defined more broadly as the expected product of their deviations from their individual expected values.²⁹

$$\begin{aligned}\gamma_k &= \text{Cov}(X_t, X_{t-k}) \\ &= E[(X_t - E[X_t]) \times (X_{t-k} - E[X_{t-k}])] \\ &= E(X_t \cdot X_{t-k}) - E(X_t) \times E(X_{t-k})\end{aligned}$$

The reference book provides:³⁰

$$\text{Cov}(X_k) = \frac{1}{n} \sum_{i=1}^n [(x_t - \bar{x})(x_{t+k} - \bar{x})]$$

Auto-covariance function (ACVF)

The **auto covariance** function of Z_t is defined by:

$$\gamma_{t,s} = \text{Cov}(Z_t, Z_s)$$

$$\gamma_k = \text{Cov}(Z_t, Z_{t-k})$$

This is a function of t and s ; $k = t - s$

²⁸ Stat Trek.com. (2017). *Statistics Notation*. [online] Available at: <http://stattrek.com/statistics/notation.aspx> [Accessed 15 Aug. 2017].

²⁹ Oxford Dictionary of Statistics, Oxford University Press, 2002, p. 104.

³⁰ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC. Equation (2.6.1), (p. 40 of 265)

Autocorrelation function (ACF)

The autocorrelation is a measure of how a random variable (say x), varies over time relative to past observations (of say x_{t-1}). E.g. If the temperature was higher than average at 12 pm, we would expect the temperature to be above average at 1pm.

The **autocorrelation function** (ACF) is defined by:

$$\rho_{t,s} = \text{Corr}(Z_t, Z_s) = \frac{\text{Cov}(Z_t, Z_s)}{\sqrt{\text{Var}(Z_t) \cdot \text{Var}(Z_s)}}$$

The lag-1 auto correlation for some time series is:

$$\rho_1 = \text{Corr}(Z_t, Z_{t-1}) = \frac{\text{Cov}(X_t, X_{t+1})}{\text{Var}(X_t)}$$

The lag- k auto correlation is defined:

$$\rho_k = \frac{\text{Cov}(X_t, X_{t-k})}{\sqrt{\text{Var}(X_t) \cdot \text{Var}(X_{t-k})}} = \frac{\gamma_1}{\gamma_0} = \frac{y_1}{\sigma_X^2}$$

Where:

- $k = t - s$
- γ_1 denotes the lag-1 autocovariance,
- γ_0 is the lag-0 autocovariance and is equal to the variance.
- The lag-0 auto correlation is 1 (because obviously each value is correlated with itself).
- $\text{Cov}(X, Y) = \frac{\sum_{i=1}^n [(x_i - \bar{x})(y_i - \bar{y})]}{(n-1)}$

Purely random Process (White noise)

A white noise process has a mean value of zero and NO correlation between its values at different times, hence any random distribution (discrete or continuous) can be white noise (e.g. Normal, Uniform, Poisson, Binomial, Log-Normal, Chi, Gamma, Beta, Weibull, Exponential etc.).

So the definitions are:

Purely Random Process (AKA White Noise):

- Autocorrelation of 0 for all values
 - This is for all the data lag-1, lag-2, lag-3 ...
- Auto covariance of 0
 - This is for all the data lag-1, lag-2, lag-3 ...
- Constant Mean value (e.g. 0)

A white noise process may represent the error observed in a model, in this case the white noise process will have a mean value of 0.

Expressed more formally a white noise process is:

$$\begin{aligned}\mu_t &= E(x_t) = \mu \\ \sigma_t^2 &= Var(X_t) = \sigma \\ \gamma_{t,s} &= Cov(X_t, X_s) = \begin{cases} \sigma^2, & t = s \\ 0, & t \neq s \end{cases} \\ \gamma_{t,s} &= Corr(X_t, X_s) = \begin{cases} 1, & t = s \\ 0, & t \neq s \end{cases}\end{aligned}$$

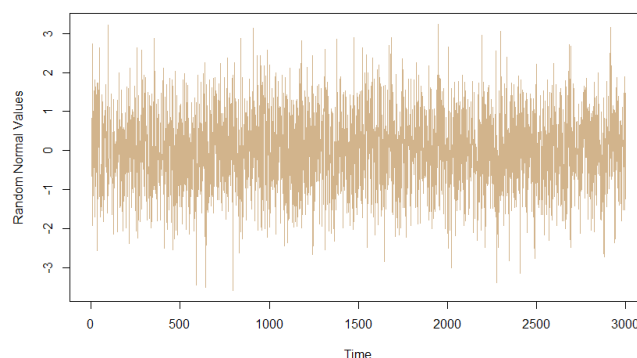
Simulating in R

```
WN_1 <- arima.sim(model=list(order=c(0,0,0)), n=50, mean=0, sd=1)
as.xts(WN_1)
```

A white noise process would look like:

```
WN_1 <- arima.sim(model=list(order=c(0,0,0)), n=300)
plot(as.xts(WN_1), lwd=0.01, col="tan", grid.ticks.on = FALSE)

plot.ts(rnorm(3000), ylab="Random Normal Values", col="tan")
```



Examples

- if \vec{x} contained 300 values $\sim \mathcal{N}(0,1)$ it would be a white noise process.
 - Because the values have $ACF = ACVF = \mu = 0$
- However a vector \vec{y} containing those 300 values and another 200 values from a normal distribution with $\mu = 0, \sigma = 29$ would NOT be a white noise process because the variation is not constant over time.

Random Walk Process

A random walk process, is such that:

$$\textit{Today} = \textit{Yesterday} + \textit{Noise}$$

And is such that:

- There is no specified mean
- Strong dependence over time
- It's changes are white noise

A Random Walk process (Z) requires a starting point, usually $Z_0 = 0$.

A Random Walk Process has only one parameter, the variance of the white noise.

The difference between each value on a random walk, is white noise, i.e. $\text{diff}(Z) = \text{WN}$.

If a_t is a white noise process with $E(a_t) = 0$ and a constant variance $\text{Var}(a_t) = \sigma_a^2$, then Z_t is a random walk process:

$$Z_t = Z_{t-1} + a_t$$

For $t = 1, 2, 3, 4 \dots$

$$z_0 = 0$$

$$Z_1 = Z_0 + a_1$$

$$= a_1$$

$$Z_2 = Z_1 + a_2$$

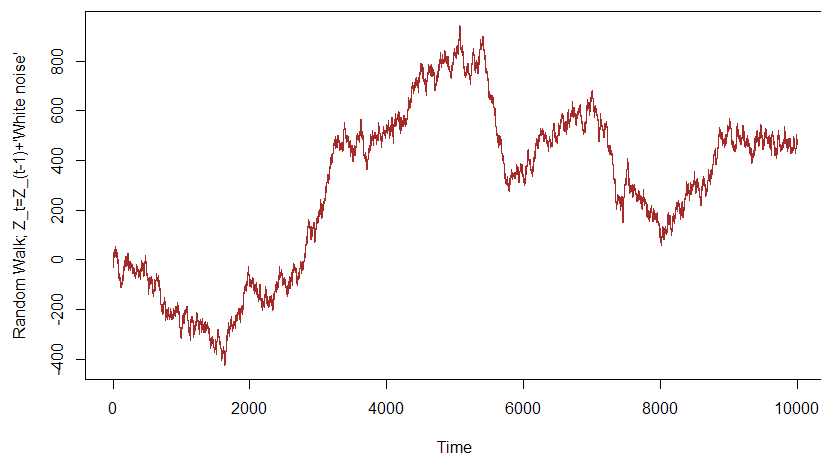
$$= a_1 + a_2$$

...

$$Z_t = \sum_{i=1}^t [a_i]$$

A random walk process could look like this:

$$\sigma_a^2 = 8 \mid n = 10000$$



Mean Value

$$\mu_t = E(Z_t) = E \left[\sum_{i=1}^t [a_i] \right] = 0$$

Variance

$$\sigma_t^2 = \text{Var}(Z_t) = \text{Var}(a_1 + a_2 + a_3 \dots a_t) = t \times \text{Var}(a_t) = t\sigma_a^2$$

Autocovariance

$$\begin{aligned} \gamma_k &= \text{Cov}(Z_t, Z_{t+k}) \\ &= t\sigma_a^2 \end{aligned}$$

I need to prove this rather than just writing it out.

Autocorrelation

$$\begin{aligned} \rho_k &= \frac{\text{Cov}(Z_t, Z_{t+k})}{\sqrt{\text{Var}(Z_t) \cdot \text{Var}(Z_{t+k})}} \\ &= \sqrt{\frac{t}{s}} \end{aligned}$$

Drift

A random Walk with drift is:

$$\text{Today} = \text{Constant} + \text{Yesterday} + \text{Noise}$$

And merely adds a constant value to the form.

A white noise process with a drift has two parameters:

- White Noise Variance
- Constant c
 - The mean of the white noise is c

Simulating in **R**

```
RW_1 <- arima.sim(model=list(order=c(0,1,0)), n=50)
as.xts(RW_1)
```

Moving Average Process

A moving average process of order 1 (**MA (1)**) is:

$$Z_t = \beta_0 a_t + \beta_1 a_{t-1}$$

Where β_0 and β_1 are constants, mean is 0 and variance is constant (σ_a^2).

Variance of Z_t

Not totally clear

Autocovariance

It can be shown that a *Moving Average Process* of order 1 has autocovariance:³¹

$$\gamma_k = \rho_k = \begin{cases} (1 + \beta_1^2), & k = 0 \\ \beta_1, & k = 1 \\ 0, & k = 1 \end{cases}$$

A *Moving Average Process* of order q has autocovariance:³²

$$\gamma_k = \begin{cases} \sum_{i=0}^q [\beta_i^2], & k = 0 \\ \sum_{i=0}^{q-k} [\beta_i \beta_{i+k}], & k = 1, 2 \\ 0, & k > q \end{cases}$$

Where: $k = t - s$

Autocorrelation

It can be shown that a *Moving Average Process* of order 1 has autocorrelation:³³

$$\rho_k = \begin{cases} 1, & k = 0 \\ \frac{\theta}{(1 + \theta^2)}, & k = 1 \\ 0, & k > 1 \end{cases}$$

A *Moving Average Process* of order q has auto correlation:³⁴

$$\rho_k = \begin{cases} 1, & k = 0 \\ \frac{\sum_{i=0}^{q-k} [\beta_i \beta_{i+k}]}{\sum_{i=0}^q [\beta_i^2]}, & k = 1, 2, 3 \dots q \\ 0, & k > q \end{cases} \quad \text{Where } k = t - s$$

Example Problem from Lecture Notes

In the lecture notes the following time series is proposed:

³¹ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC.

Equation (2.5.6), (p. 39 of 265); Compare this with the lecture notes and the formula to conclude that the numerator of the autocorrelation corresponds to the autocovariance.

³² Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC.

Equation (3.1.2), (p. 46 of 265)

³³ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC.

Equation (2.5.6), (p. 39 of 265)

³⁴ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC.

Equation (3.1.2), (p. 46 of 265)

$$X_t = a + bt + Z_t$$

Where:

- $a, b \in \mathbb{R}$; constant values
- t is a variable representing the time
 - We will assume that $t = 1, 2, 3, 4, 5, 6 \dots n$
- Z_t is a stationery series, with a mean value of 0.

Solving the Mean value function

The mean value in this case is the expected value:

$$\begin{aligned}\mu_t &= E(X_t) \\ &= E(a + bt + Z_t) \\ &= E(a) + E(bt) + E(Z_t) \\ &= E(a) + b \cdot E(t) + E(Z_t)\end{aligned}$$

| | | |
|--|--|--------------------------------------|
| $E(a) = a$ a is just a constant value, e.g. if I had a constant value, say 5, I would expect that value to be 5 because, well, it's 5. | $E(t) = \text{mean}(t)$ $= \text{mean}(1, 2, 3, 4, 5, \dots n)$ $= \frac{1}{n} \sum_{t=1}^n [t]$ $= \frac{1}{n} \left(\frac{n(n+1)}{n} \right)$ $= \frac{(n+1)}{n}$ | $E(Z_t) = \text{mean}(Z_t)$ $= 0$ |
|--|--|--------------------------------------|

By substituting this into the equation:

$$\begin{aligned}\mu_t &= E(a) + b \cdot E(t) + E(Z_t) \\ &= a + \frac{b(n+1)}{n}\end{aligned}$$

Stationary Time Series

A process is strictly stationary if all probabilistic behaviour is unchanged by shifts in time.³⁵

This is a very strong assumption, it will often suffice to assume that a process is weakly stationary.

A process is weakly stationary if its mean, variance and covariance are unchanged by shifts in time.

A time series is stationary if the properties of the underlying model do not change through time,³⁶ i.e. the properties do not depend on the time at which the series is observed.³⁷

A time series is said to be stationary if:³⁸

- The mean function is a constant value, it does not change through time and
 - $\mu_t \in R$; i.e. constant
- The auto covariance function does not change through time.
 - $\gamma_{t,s} = \gamma_{0,|t-s|}$

Observe that:

- Purely random processes are stationary
- MA(1) is Stationary
- Random Walk Processes are not stationary
 - Because $\text{Cov}(X_k, X_{k+t}) = t\sigma_a^2$ is dependent on time.

A time series that is stationary in this sense is known as a weak stationary, covariance stationary, second-order stationary or wide sense stochastic process.³⁹

A stationary process can be modelled with fewer parameters, that's why it is useful.

Example

For a stationary process (Y_t) at time (t), the following should be true:

$$\text{Cov}(Y_2, Y_5) = \text{Cov}(Y_5, Y_8)$$

35

https://s3.amazonaws.com/assets.datacamp.com/production/course_1143/slides/ch2_4_supplementary.pdf

³⁶ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC. Equation (2.4.1), (p. 34 of 265)

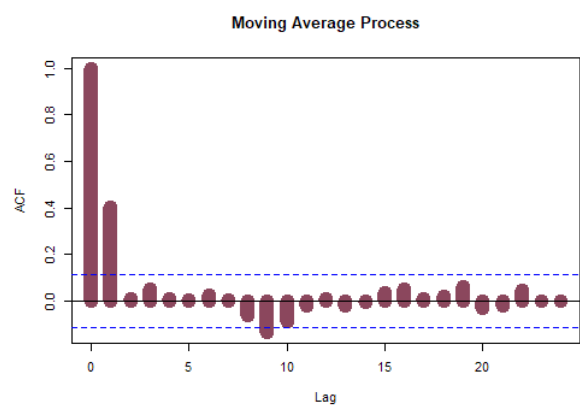
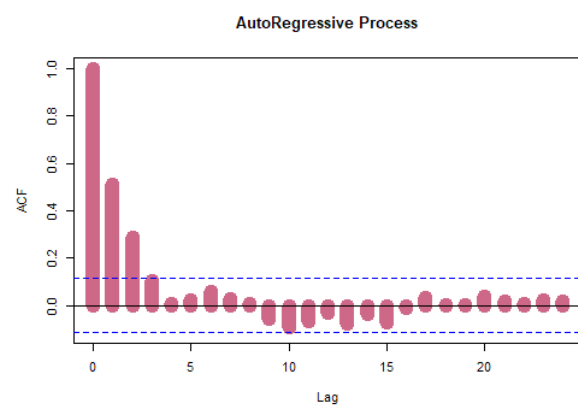
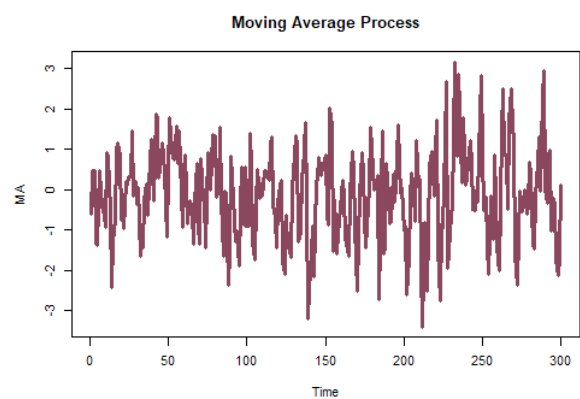
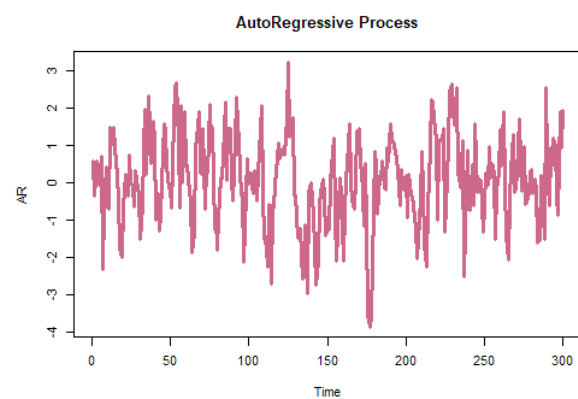
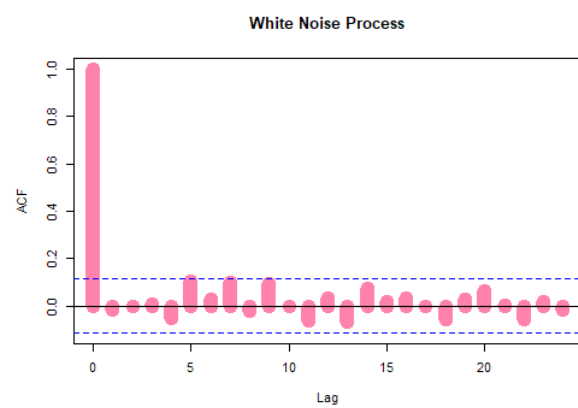
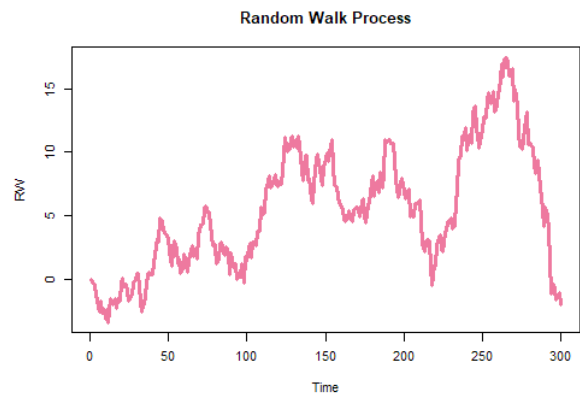
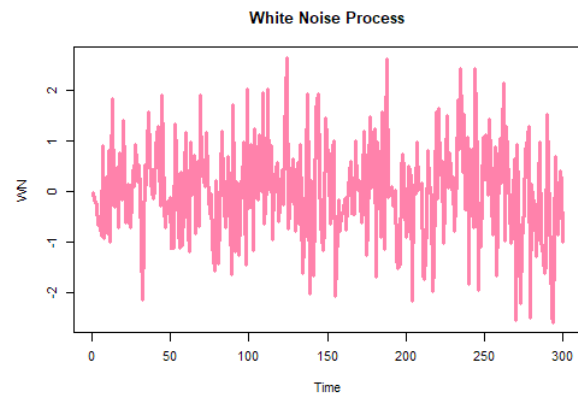
³⁷ Hyndman Rob J. and Athanasopoulos George (2013) *Forecasting: Principles and Practice*, OTexts, <<https://www.otexts.org/fpp>>

³⁸ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC. Equation (2.4.1), (p. 34 of 265)

³⁹ Imdadullah, M. (2016). *Stationary Stochastic Process*. [online] Basic Statistics and Data Analysis. Available at: <http://itfeature.com/time-series-analysis-and-forecasting/stationary-stochastic-process> [Accessed 18 Aug. 2017].

Appendix

Plot of Various Time Series Functions



Comparison of White Noise and Random Walk Data

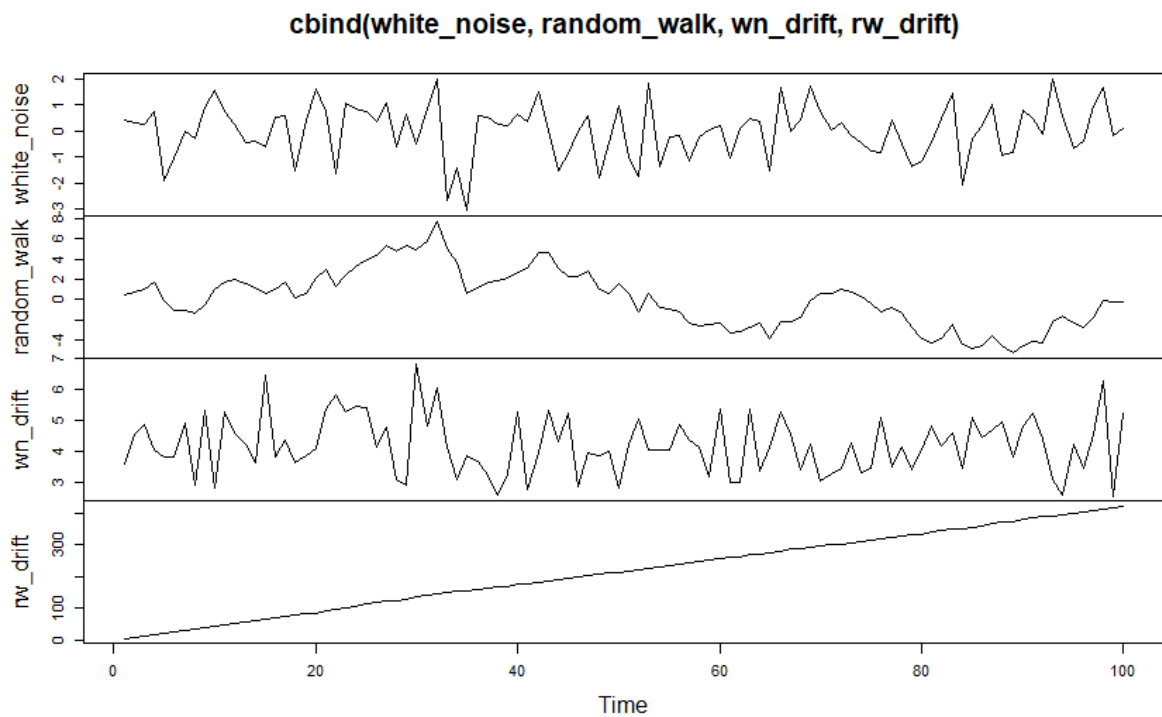
```
# Use arima.sim() to generate WN data
white_noise <- arima.sim(model=list(order=c(0,0,0)), n=100)

# Use cumsum() to convert your WN data to RW
random_walk <- cumsum(white_noise)

# Use arima.sim() to generate WN drift data
wn_drift <- arima.sim(model=list(order=c(0,0,0)), n=100,
mean=4)

# Use cumsum() to convert your WN drift data to RW
rw_drift <- cumsum(wn_drift)

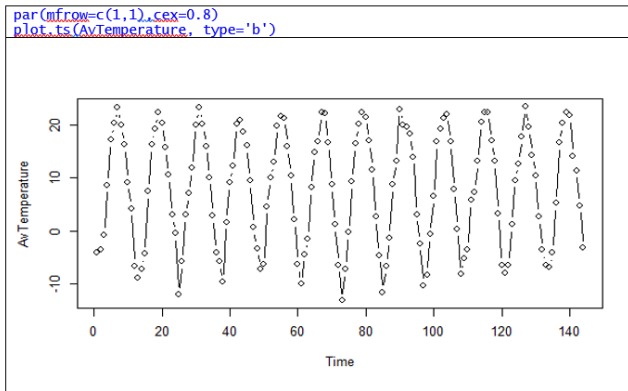
# Plot all four data objects
plot.ts(cbind(white_noise, random_walk, wn_drift, rw_drift))
```



Exercise 6.1

Part (i) (e.g. 6.2)

Suggest a model with specific trends for the following time series:



This plot is of monthly temperatures within a city.

$$Y_t = (L_t + C_t + S_t) + Z_t$$

- **Deterministic**
 - **Overall trend L_t**
 - Positive due to:
 - increased pavement area absorbing solar radiation with higher thermal heat capacity and conductivity than grass
 - Increase in Tall buildings
 - reflecting sunlight to be absorbed (Urban Canyon effect)
 - Greenhouse effect
 - **Cyclical trend C_t (unpredictable period):**
 - El Niño and La Niña weather patterns that result from variations in ocean temperatures may increase or decrease the temperature in periods ranging from nine months to a couple years.⁴⁰
 - **Seasonal trend S_t (predictable period):**
 - Seasonal trends will oscillate with a 12 month period.
- **Stochastic**
 - **Random Variation Z_t**
 - **Random Error**
 - Unforseeable events (natural disasters, sunspots etc.)
 - **Systematic Errors**
 - Inability to measure temperature at certain areas

So the model may look something like this:

$$Y_t = L_t + S_t + C_t + Z_t$$

$$L_t = mt + b: b, m \in \mathbb{R} \quad (\text{perhaps } b \text{ is the global mean temperature value})$$

$$S_t = A \cdot \sin(\omega \cdot t + \phi) + C: A, \omega, \phi, C \in \mathbb{R}$$

$$C_t = B \cdot \sin(\Omega \cdot t + \Phi) + D: B, \Omega, \Phi, D \in \mathbb{R}$$

⁴⁰ Oceanservice.noaa.gov. (2017). *What are El Niño and La Niña?*. [online] Available at: <https://oceanservice.noaa.gov/facts/ninonina.html> [Accessed 16 Aug. 2017].

Creating the model in R

```
# Load Packages -----
library(scales)

# Create Variables -----
n <- 144 #no. of observations
mean_temp <- 10 #assume mean value is 10 deg celsius

time_month <- seq(1:144)
Random_error <- set.seed(654);rnorm(n = 144, mean = 0, sd = 3) #The points jump around
at about 3sd eyeballing it
Long_trend <- (3/100/12)*time_month+mean_temp #Assume 3 degrees a century (due to
various effects)
Observed_seasonal_change <- c(12, 10, 6, 0, -7, -11, -20, -15, -8, -5, 2, 10)

# Find Seasonal Variation -----
#By running and stopping the code, closer approximations can be found for these

A <- 11
B <- 36
C <- 2
D <- 0

sin_model_change <- A*sin(B*sample_year+C)+D
diff <- sum((sin_model_change-
Observed_seasonal_change)*100/sin_model_change)/length(Observed_seasonal_change)
#Average Percent Error

inc <- 0.01
max_val <- 100 #period < 12, magnitude < 25, phase < period,

#This was done with 'while' loops, but apparently "if(){}else()" is much faster,
confirm this

# A Loop -----
while(diff>5|A<max_val){
  A <- A+0.01
  # B Loop -----
  while(B<max_val){
    B <- B+inc
    # C Loop -----
    while(C<6){
      C <- C+0.01
    }
    # D Loop -----
    while(D<12){
      D <- -0.1 # D+0.1
      #Test
      sin_model_change <- A*sin(B*sample_year+C)+D
      diff <- sum((sin_model_change-
Observed_seasonal_change)*100/sin_model_change)/
length(Observed_seasonal_change) #Average Percent Error
    }
    D <- 9
    C <- 2
    B <- 0
    print(percent(A/max_val));print(diff)
  }
}

print(A);print(B);print(C);print(D)
```

```
print(A);print(B);print(C);print(D)
```

```
#By playing around with this wildly inefficient loop code, you will come to find that  
the seasonal variation is  
#approximately something like 11*sin(36*sample_year+C)+D
```

```
#Now by tweaking the variables in response to the plot:
```

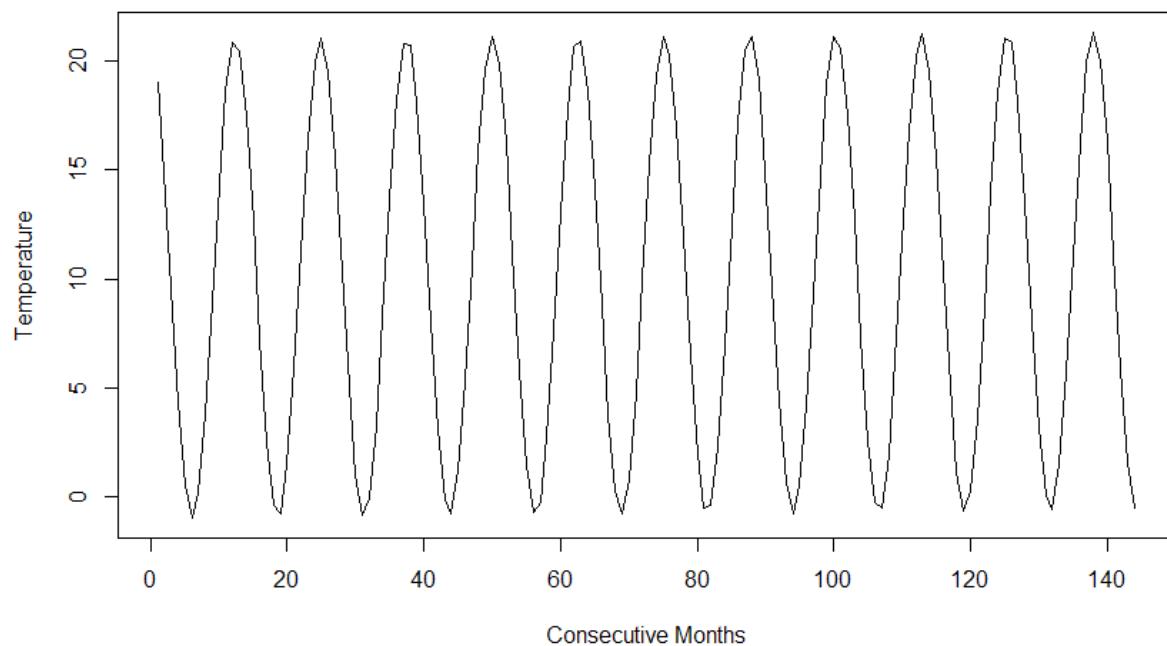
```
seasonal_trend <- 11*sin(0.5*sample_year-80)+0  
plot.ts(seasonal_trend, ylab="Temp Difference")  
plot.ts(Observed_seasonal_change, ylab="Observed Temp Difference")
```

```
#Thus the Seasonal Trend is:  
seasonal_trend <- 11*sin(0.5*time_month-80)+0
```

```
#Now calculate the Temperature (Without the Cyclical Trend, because time).
```

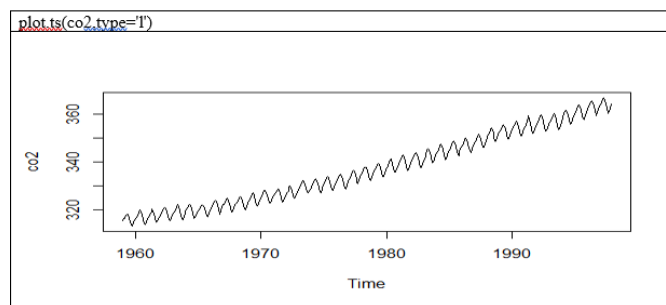
```
Temp_Yt <- Long_trend+seasonal_trend  
plot.ts(Temp_Yt, ylab="Temperature", xlab="Consecutive Months")
```

Which provides the output:



Part (ii) (e.g. 6.3)

Suggest a model with specific trends for the time series in the following time series:



This plot is of CO_2 concentrations within a city.

- **Deterministic**
 - **Overall trend L_t**
 - Positive due to:
 - Burning fossil fuels
 - Industrial Processes
 - Farming
 - Land Use
 - **Cyclical trend C_t (unpredictable period):**
 - El Niño and La Niña weather patterns that result from variations in ocean temperatures may increase or decrease the temperature in periods ranging from nine months to a couple years.⁴¹
 - This temperature may effect plant growth and effect the
 - **Seasonal trend S_t (predictable period):**
 - Seasonal trends will oscillate with a 12 month period.
 - This will affect the rate at which photosynthetic plants use CO_2 to create sugars (hence diminishing the atmospheric concentration).
 - Although, this gets complicated, as there are two hemispheres that lead to CO_2 being approximated by a sine curve, the greater land mass at the Northern hemisphere may outcompete the southern hemisphere leading to greater CO_2 concentrations in May.⁴²
- **Stochastic**
 - **Random Variation Z_t**
 - **Random Error**
 - Unforseeable events (wildfires, volcanic eruptions etc.)
 - **Systematic Errors**
 - Inability to measure CO_2 at certain areas or to certain accuracies.

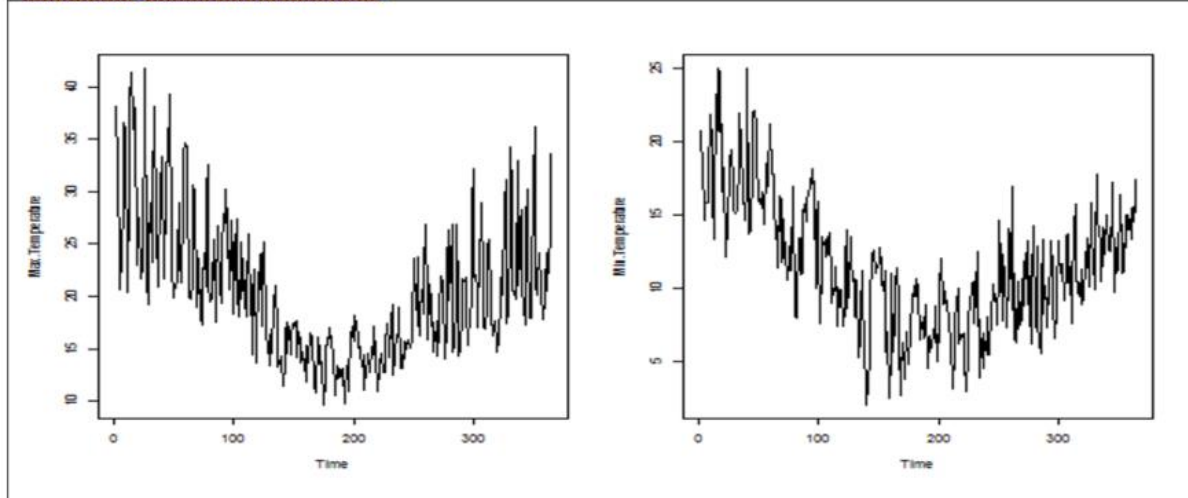
The model will be similar to part (i) (e.g. 6.2)

⁴¹ Oceanservice.noaa.gov. (2017). *What are El Niño and La Niña?*. [online] Available at: <https://oceanservice.noaa.gov/facts/ninonina.html> [Accessed 16 Aug. 2017].

⁴² Monroe, R. (2017). *Why Does Atmospheric CO_2 Peak in May?*. [online] The Keeling Curve. Available at: <https://scripps.ucsd.edu/programs/keelingcurve/2013/06/04/why-does-atmospheric-co2-peak-in-may/> [Accessed 16 Aug. 2017].

Part (iii) (e.g. 6.1)

```
par(mfrow=c(1,2),cex=0.5)  
plot.ts(Max.Temperature)  
plot.ts(Min.Temperature)
```



- **Deterministic**
 - **Overall trend L_t**
 - Very slightly Positive due to:
 - Urbanisation
 - Greenhouse effect
 - **Cyclical trend C_t (unpredictable period):**
 - Day Night Cycle will affect temperature, although the onset of day/night is predictable via a sinusoidal wave, the change that may have in temperature may as well be random as it is dependent on (humidity, wind, temperature etc.)
 - **Seasonal trend S_t (predictable period):**
 - Seasonal trends will oscillate with a 12 month period, reasonably predictable temperature change.
 - I guess theoretically things like eclipses could be included here if that crossed the measurement region.
- **Stochastic**
 - **Random Variation Z_t**
 - **Random Error**
 - Unforseeable events (weather patterns, cloud coverage,)
 - Systematic Errors
 - Inability to measure temperature to a certain accuracy.

Example of Time Series Data.

$$Y_t = (L_t + C_t + S_t) + Z_t$$

A time series model could be made to track the GDP per capita produced by a country, this value would be Y_t :

- **Deterministic**
 - **Overall trend L_t**
 - Positive increase in production due to social factors such as education and availability of means.
 - **Cyclical trend C_t :**
 - Boom/Bust cycles may take place over approximate 6 year intervals.⁴³
 - **Seasonal trend S_t :**
 - GDP may drop in western countries over the holiday period in December/January
 - GDP may rise in the dry season (i.e. winter) in tropical areas like Florida or Darwin (due to the impact of pleasant climate)
 - GDP may drop in Middle Eastern countries over Ramadan due to the reduced diet.
- **Stochastic**
 - **Random Variation Z_t**
 - **Random Error**
 - Unforseeable events (natural disasters, riots etc.)
 - Legislative decisions.S
 - **Systematic Errors**
 - Inability to measure production to false reporting or tax avoidance etc.

⁴³ Fontinelle, A. (2017). *Boom And Bust Cycle*. [online] Investopedia. Available at: <http://www.investopedia.com/terms/b/boom-and-bust-cycle.asp> [Accessed 16 Aug. 2017].

Trend Estimation and Residual Analysis

Wk. 7 Material | Due 4 Sep 2017

Contents

| | |
|--|----|
| Trends in Time Series | 78 |
| Estimating Time Series Trends by Regression..... | 79 |
| Using the Ordinary Least Squares Methods | 79 |
| Example..... | 80 |
| Residuals | 81 |
| Residual Analysis..... | 82 |
| Residual Analysis..... | 82 |
| Autocorrelation..... | 82 |
| Residual ACF in R | 82 |
| Model Choice | 83 |
| ACF and PACF test..... | 83 |
| Residual Analysis in R | 84 |

Trends in Time Series

There are two types of trends in time series:

- Deterministic
 - E.g. long term trend
 - E.g. temperature rising due to greenhouse effect.
 - Cyclical trend
 - E.g. oscillation of temperature due to El Niño and La Niña cycles
 - Recall that a cycle is a less predictable changing trend in data, like a boom or bust cycle in economics.
 - Seasonal trend
 - E.g. oscillation of temperature due to the Earth's seasons
 - Recall that a season is a predictable oscillating trend in data, like the Earth's seasons.
- Stochastic
 - Random Error
 - Unforeseeable fluctuations in data
 - Systemic Error
 - Shortcomings of your capacity to measure accurately
 - E.g. measuring a planet may effected significantly by the fact that you can only tell it'd diameter by $\pm 5 \text{ km}$

The material here is concerned with estimating deterministic error via regression.

Material from Lectures 8 and 9 provide models for estimating stochastic data.

Estimating Time Series Trends by Regression

A trend in a time series can be thought of like a long-term change in the mean response value.

Using the Ordinary Least Squares Methods

Given a time series, we can use the **ordinary least-squares** method for linear regression (as in Lectures 4 and 5, concerned with simple linear regression and multiple linear regression).

An ordinary least-squares method assumes independent observations, time series data is not independent because observations are correlated, they affect one another (e.g. measuring the temperature and observing $30^{\circ}C$ at 9AM makes it pretty unlikely to be $-3^{\circ}C$ at 11AM)

The **generalised least-squares method** assumes dependent observations and is what we are supposed to use.

The ordinary least squares method is easier/quicker than the **generalised least-squares method**, hence it's use. This 'shortcut' is justified as such:

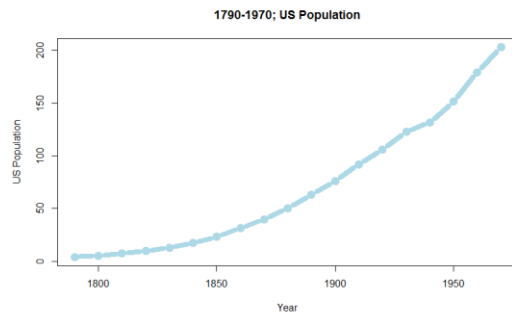
1. The **ordinary least-squares** is close to the **generalised least-squares method** for large sample sizes.
2. The unknown correlation between observations (which measures our dependence, think acf) will not be detected by the **ordinary least-squares** and can be analysed with a stochastic trend analysis later.

Example

The US population data for 1790-1970 is included in **R**:

If we look at a plot of that data:

```
ts.plot(uspop, type="b", ylab="US Population", xlab="Year", main="1790-1970; US  
Population", lwd=8, col="lightblue")
```



Observe the dip around 1930 is likely attributed to the Great Depression, this is an example of a stochastic cyclical trend that could affect the model.

The trend is likely exponential, it could potentially be modelled by a quadratic function which we will try and fit:

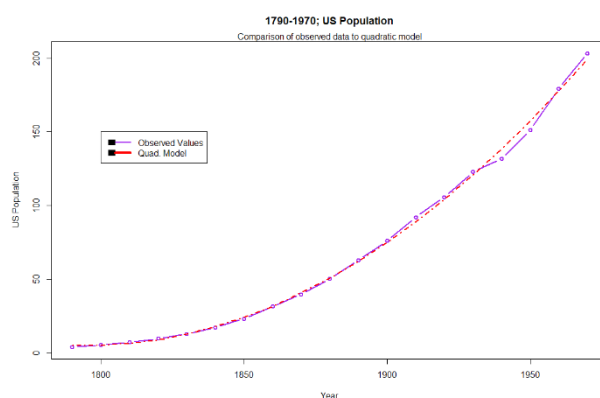
```
#Create a Quadratic model to compare

#Create the quadratic regression y=ax+bx^2

#x-value
usoptimevar1 <- as.numeric(index(uspop))
usoptimevar2 <- as.numeric(index(uspop))^2

#Find the coefficients
uspopt.qm <- lm(uspopt~usoptimevar1+usoptimevar2)
uspopt.qm_sum <- summary(uspopt.qm)

#Plot a dashed line over using the points command
uspopt.qm_fit <- uspop.qm$fitted.values
points(y=uspopt.qm_fit, x=usoptimevar1, type="l", lty=4, lwd=2, col="red")
mtext("Comparison of observed data to quadratic model")
legend(1800,150, legend = c("Quad. Model", "Observed Values"),
      fill = TRUE, col=c("purple", "Red"),lty=(1), lwd=c(2,4))
```



The equation for the quadratic model is:

$$US\ Pop = 0.006345 \cdot t^2 - .02278 \cdot t + 0.0002045$$

Where t is the relevant year.

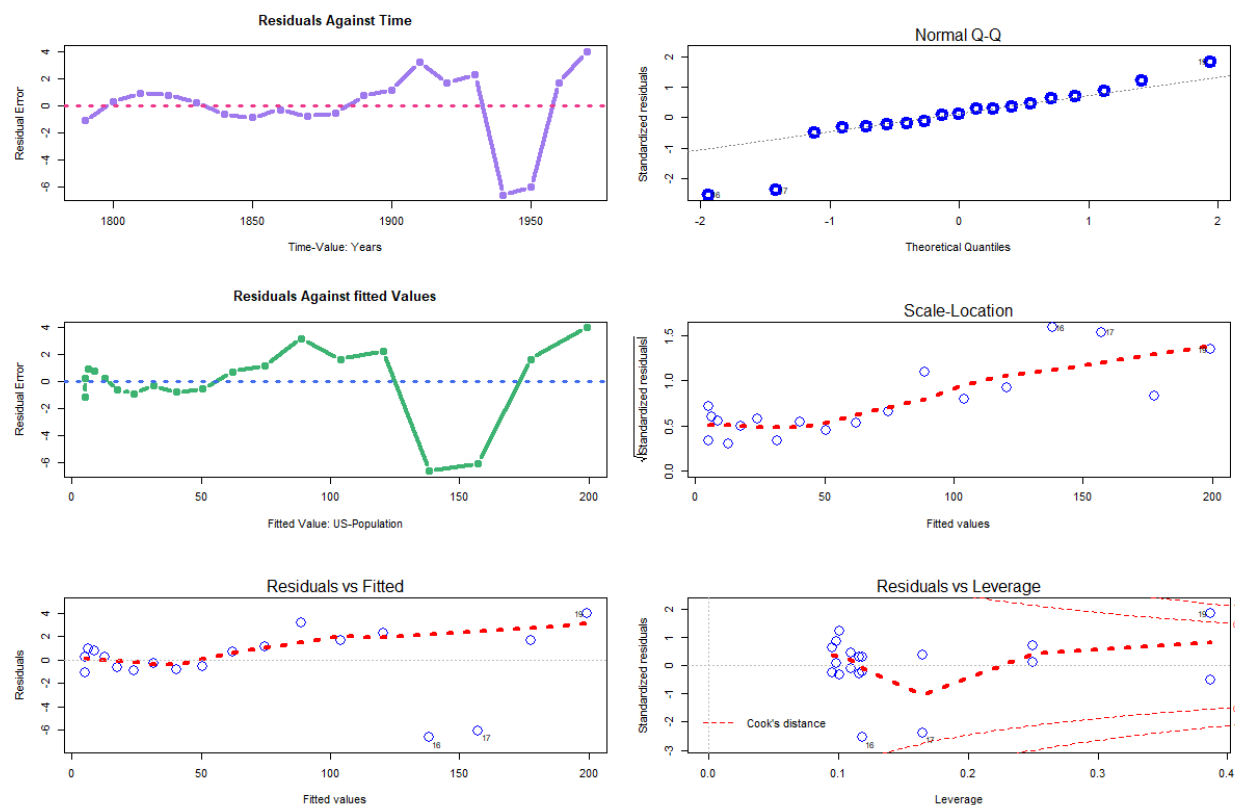
Residuals

Although the model fits well we ought to consider the residuals before settling on the model:

```
# #Create a residual plot -----
layout(matrix(nrow=3, data = 1:6))
uspopqmresid <- uspop-qm_fit
plot(y=uspopqmresid, x=uspop.timevar1, type = 'b', lwd=4, col="mediumpurple2", xlab=
"Time-Value: Years", ylab="Residual Error", main="Residuals Against Time")
par(xpd=FALSE);abline(0,0, col="violetred2", lwd=3, lty=3)

plot(y=uspopqmresid, x=uspop.qm_fit, type = 'b', lwd=4, col="mediumseagreen", xlab=
"Fitted Value: US-Population", ylab="Residual Error", main="Residuals Against fitte
d Values")
par(xpd=FALSE);abline(0,0, col="royalblue2", lwd=3, lty=3)

plot(uspop.qm, lwd=4, col="Blue", lty=3, cex=2)
```



Residuals should represent normally distributed white noise (aka *Gaussian White Noise*), if there is still correlation within the residuals a different model or additional parameters should be sought.

Observe that there are certain patterns in the residuals that potentially undermine the assumption that the error is normally distributed.

However the large residual (indicating model overestimation) around 1930 is the result of stochastic error (flowing from the Great Depression) and is likely outside the scope of this model and hence we will allow it.

Residual Analysis

Residual Analysis

The residual(e) is the difference between the model(\hat{y}_i) and the observed data (y_i) and should be the random error ε :

$$e_i = y_i - \hat{y}_i = \varepsilon$$

Where:

ε is normally distributed white noise

In modelling time series it is important that the residuals are normally distributed white noise.

White noise is the expected random error in a model, if the residuals are not distributed as such, it is likely that the model is false and is causing the residuals to be distributed in another fashion. If the model is correct the residuals will be normally distributed white noise.

Autocorrelation

For residual values corresponding to different times (e_t):

$$e_1, e_2, e_3, e_4 \dots$$

The autocorrelation for lag k , is given by:

$$\rho_k = \frac{\sum_{t=1}^{n-k} [(e_t - \bar{e}) \cdot (e_{t+k} - \bar{e})]}{\sum_{t=1}^n [(e_t - \bar{e})^2]}$$

As the residuals should be normally distributed white noise, the ACF should be 0 for all lags.

Residual ACF in **R**

The ACF of the residuals can be calculated and determined in **R** thusly:

```
residuals <- data-lm(data)$fitted; acf(residuals)
```

However the observations must be evenly spaced, so be aware that this could be a problem for an `xts` object (but should be fine with a `ts` object which must be evenly spaced by definition of the class).

Model Choice⁴⁴

In Time series analysis a few models may seem reasonable (e.g. exponential vs. quadratic or MA(1) or AR(1)).

In **R** the two most common tests to decide between models in are the AIC and BIC tests and where those tests conflict the simpler model is usually chosen.

The *parsimony principle* suggests that the simplest explanation that fits the evidence should be chosen, like *Occam's razor*.

ACF and PACF test

AR and MA models look very similar and a model can't be determined simply by looking at the data.

The **autocorrelation function (acf)** and **partial autocorrelation function (pacf)** are used to determine the model orders.

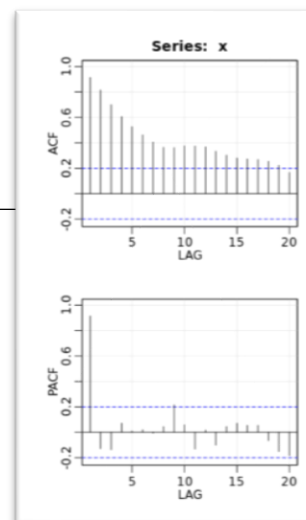
The ACF and PACF can be used thusly:⁴⁵

| | $AR(p)$ ⁴⁶ | $MA(q)$ ⁴⁷ | $ARMA(p, q)$ |
|-------------|-----------------------|-----------------------|--------------|
| acf | Tails off | Cuts off lag q | Tails off |
| pacf | Cuts off lag p | Tails off | Tails off |

```
# Generate 100 observations from the AR(1) model
x <- arima.sim(model = list(order = c(1, 0, 0), ar = .9), n = 100)
```

```
# Plot the generated data
plot(x)

# Plot the sample P/ACF pair
require(astsa)
acf2(x)
```



As the acf tails off but the last significant pacf value occurs before 1, we will model an AR(1) time series.

```
#Generate an AR(1) model for the data
Sarima(x, p=1, q=0, d=0)
```

⁴⁴ Data Camp video by David Stoffer.

⁴⁵ Stoffer, D. (2017). *Time Series Analysis*. Free Dog Publishing, p.34.

⁴⁶ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.2 p. 45 of 265.

⁴⁷ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.3 p. 46 of 265.

Residual Analysis in **R**

Built in

Built-in Tools to analyse the residuals:

```
# Built in Residual Diagnostics -----
arima_resid <- function(resid, p=0, q=0){
  my_colours <- c("#798BC6", "#61CC96", "#242038")

  #Residual diagnostics
  layout( matrix(nrow=3, ncol=2, byrow=1, data=c(1,1,2,3,4,4)))

  ts.plot(resid, xlab="Residuals",
           main="Residuals over Time", col=my_colours[3], lwd=2)
  abline(0,0, col=my_colours[2], lwd=3)

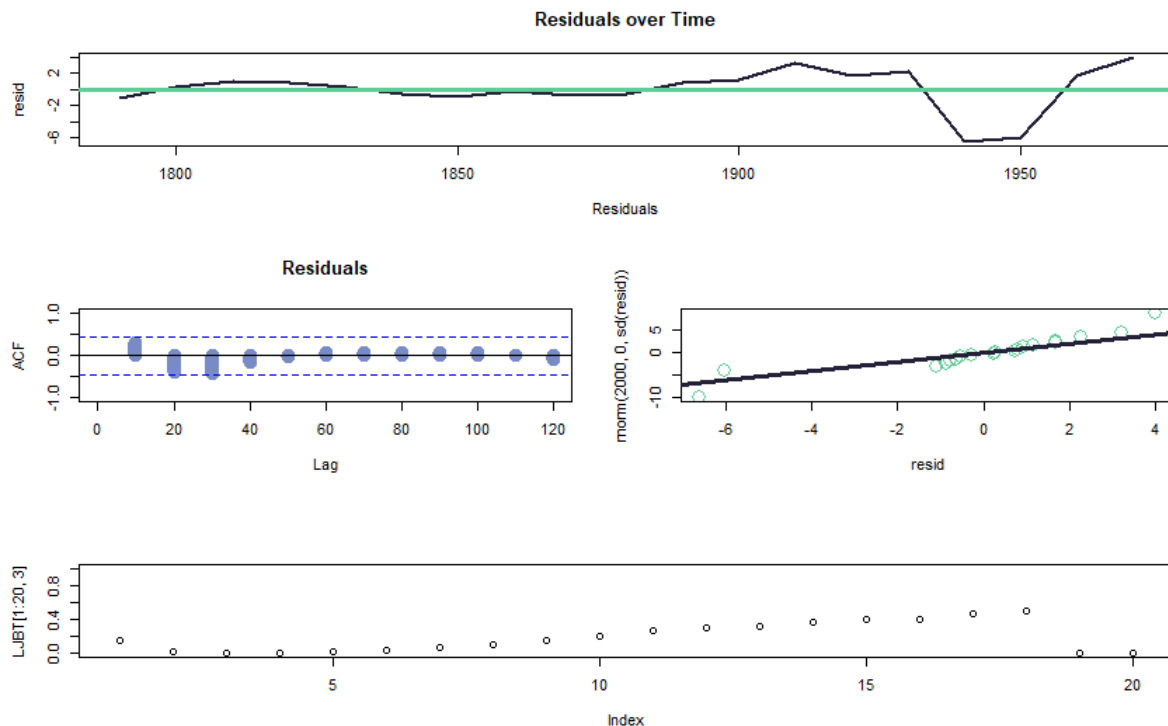
  x.acf <- acf(resid, plot = 0)
  #Remove Lag 0
  x.acf$acf[1] <- NA
  plot(x.acf,lwd=10, col=my_colours[1], ylim=c(-1,1), main="Residuals")

  qqplot(resid, rnorm(2000,0, sd(resid)), col=my_colours[2], cex=2)
  abline(0,1, lwd=3, col=my_colours[3])

  require(FitAR)
  # LJBT <- LjungBoxTest(resid, k=1)
  # plot(LJBT[1:20,3], ylim=c(0,1))
  LBQPlot(resid, k=p+q)

  layout(matrix(1))
}

arima_resid(uspopqmrresid)
```

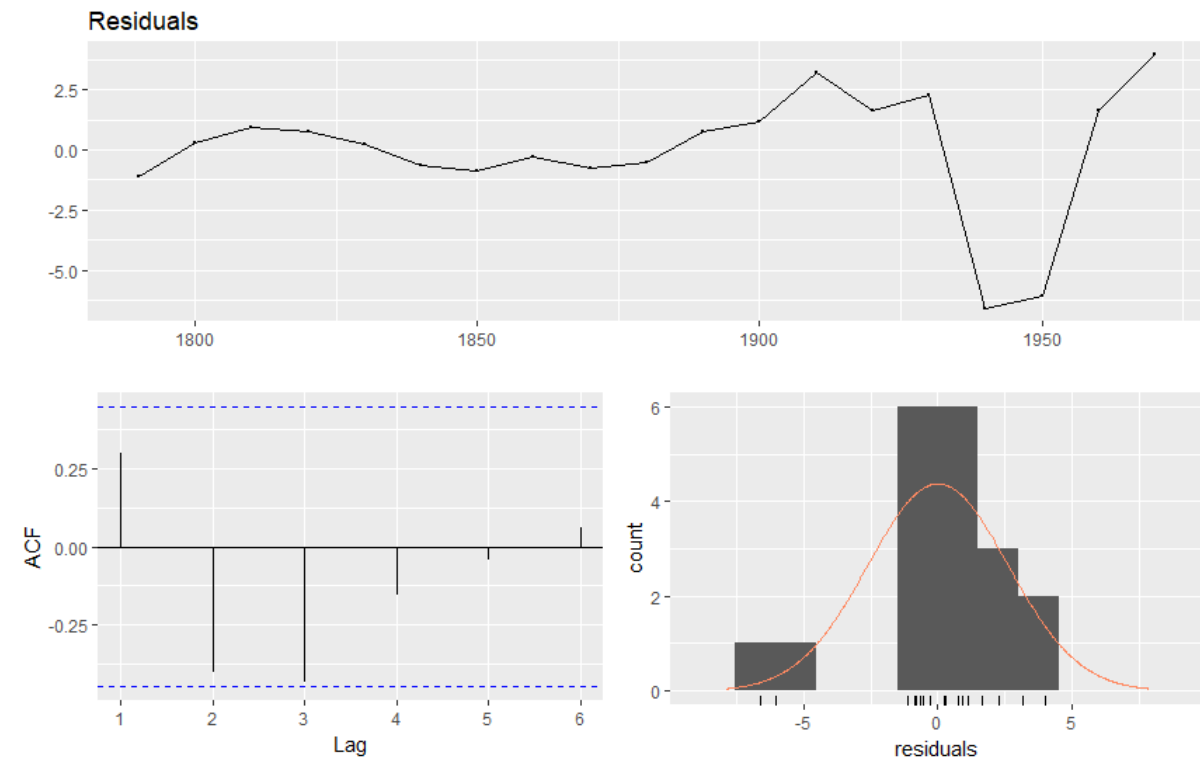


Forecast Package

The following code will produce residual diagnostics including a time series plot, ACF plot and

```
require(forecast)
my_colours <- c("#798BC6", "#61CC96", "#242038")
checkresiduals(uspopgmresid, col=my_colours[1])
```

histogram:

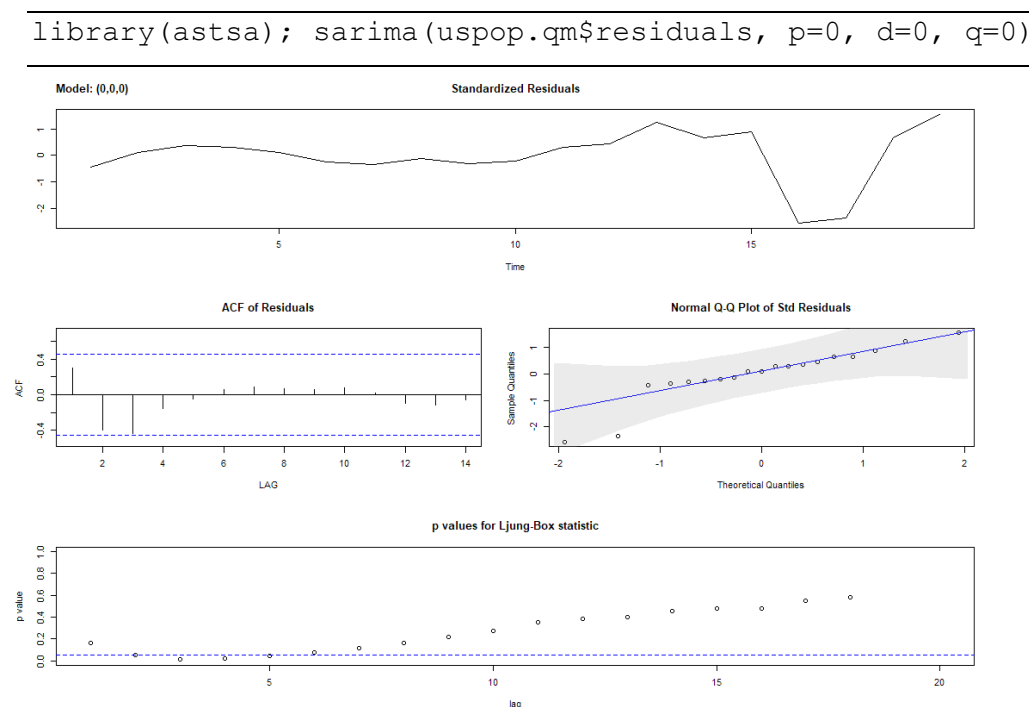


ASTSA Package

The 'sarima(x, p=, d=, q=)' function can fit data to an AR, MA, ARMA or ARIMA model and provide the following residual analysis:

1. Standardized residuals
 - a. The standardised residuals can be inspected for patterns that challenge the assumption of normality.
2. Sample ACF of residuals
 - a. White noise should not have any autocorrelation, if the residuals have autocorrelation then the residuals are not normally distributed white noise and the model may not be correct.
3. Normal Q-Q plot
 - a. If the residuals are normal the points should line up 1-1, if the points do not line up this may be evidence that the residuals are not normally distributed, and hence the model is not appropriate.
4. Q-Statistic p-values
 - a. A statistical test for 'whiteness', if most blue points are above the line it's safe to assume the noise is white, i.e. residuals are normally distributed.
5. Ljung-Box Statistic
 - a. If many of the points are below the blue line there is still some correlation left in the residuals and another model should be fitted or a parameter added.
 - i. Null hypothesis: the noise is White, rejected at 5% significance (which is the p-value)
 - ii. Alternate hypothesis: the noise is not white, receted at 5% significance.

Moreover by choosing a white noise model the 'sarima(resid, 0, 0, 0)' function will pass residuals (or any values technically) straight through to the generated plots, e.g. from the US population example:



Theory on Residuals

Assumptions made about residuals in modelling and forecasting time series:

Essential Assumptions

1. Residuals should be uncorrelated
 - a. Otherwise there is information in the residuals that should have been captured by the forecasting/modelling method
2. Residuals should have a zero mean
 - a. Otherwise the model/forecast would be biased and the model would be adjusted until the residuals had a mean of zero.

Useful Properties (for Prediction Intervals)

3. The residuals have constant variance.
4. The residuals should be normally distributed.

Conclusion

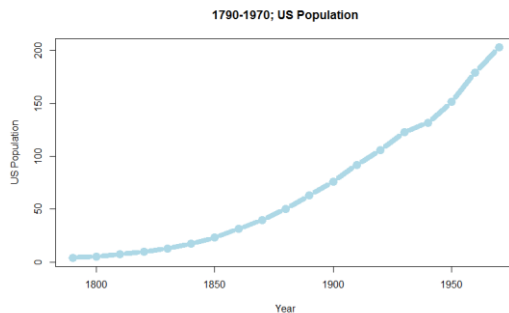
So our residuals should look like white noise (properties 1, 2 and 3), this white noise should be normally distributed (4).

Exercise 7.1 (e.g. 7.1)

The us population data for 1790-1970 is included in **R**:

If we look at a plot of that data:

```
ts.plot(uspop, type="b", ylab="US Population", xlab="Year", main="1790-1970; US  
Population", lwd=8, col="lightblue")
```



Observe the dip around 1930 is likely attributed to the Great Depression, this is an example of a stochastic cyclical trend that could affect the model.

The trend is likely exponential, it could potentially be modelled by a quadratic function which we will try and fit:

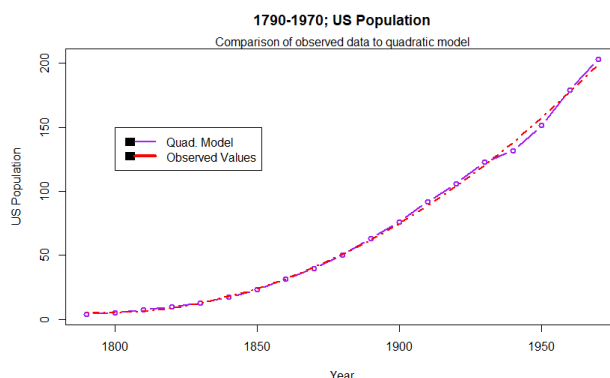
```
#Create a Quadratic model to compare

#Create the quadratic regression y=ax+bx^2

#x-value
uspoptimevar1 <- as.numeric(index(uspop))
uspoptimevar2 <- as.numeric(index(uspop))^2

#Find the coefficients
uspop.qm <- lm(uspop~uspoptimevar1+uspoptimevar2)
uspop.qm_sum <- summary(uspop.qm)

#Plot a dashed line over using the points command
uspop.qm_fit <- uspop.qm$fitted.values
points(y=uspop.qm_fit, x=uspoptimevar1, type="l", lty=4, lwd=2, col="red")
mtext("Comparison of observed data to quadratic model")
legend(1800,150, legend = c("Quad. Model", "Observed Values"),
      fill = TRUE, col=c("purple", "Red"),lty=(1), lwd=c(2,4))
```



The equation for the quadratic model is:

$$US\ Pop = 0.006345 \cdot t^2 - .02278 \cdot t + 0.0002045$$

Where t is the relevant year.

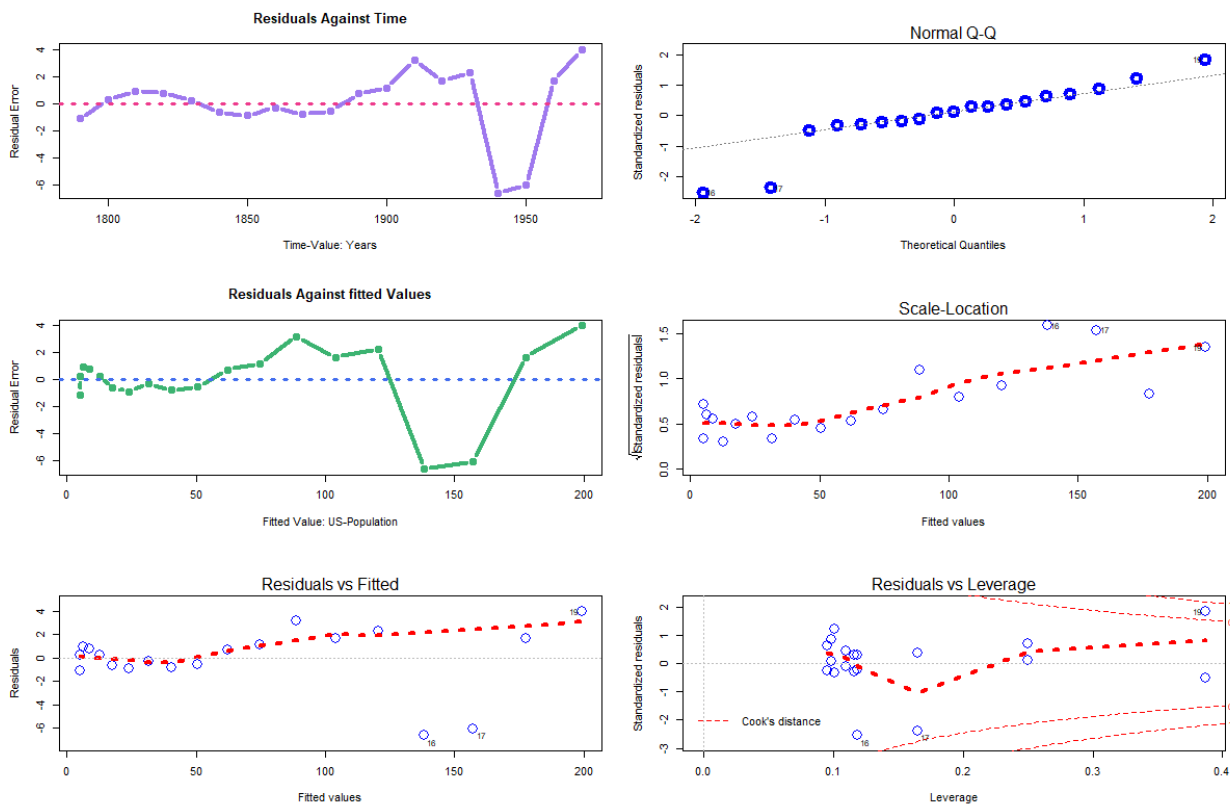
Residuals

Although the model fits well we ought to consider the residuals before settling on the model:

```
# #Create a residual plot -----
layout(matrix(nrow=3, data = 1:6))
uspopqmresid <- uspop-qm_fit
plot(y=uspopqmresid, x=uspop.timevar1, type = 'b', lwd=4, col="mediumpurple2", xlab=
"Time-Value: Years", ylab="Residual Error", main="Residuals Against Time")
par(xpd=FALSE);abline(0,0, col="violetred2", lwd=3, lty=3)

plot(y=uspopqmresid, x=uspop.qm_fit, type = 'b', lwd=4, col="mediumseagreen", xlab=
"Fitted Value: US-Population", ylab="Residual Error", main="Residuals Against fitte
d Values")
par(xpd=FALSE);abline(0,0, col="royalblue2", lwd=3, lty=3)

plot(uspop.qm, lwd=4, col="Blue", lty=3, cex=2)
```



Observe that there are certain patterns in the residuals that potentially undermine the fit of this model.

However the large residual around 1930 is the result of stochastic error (flowing from the Great Depression) and is likely outside the scope of a model.

AR, MA, ARMA, ARIMA Models

Wk. 8 Material | 4 September 2017

Introduction

These models are for stationary time series, the ARIMA Model however is for either stationary or non-stationary.

In the discussion below, the following variables are used:

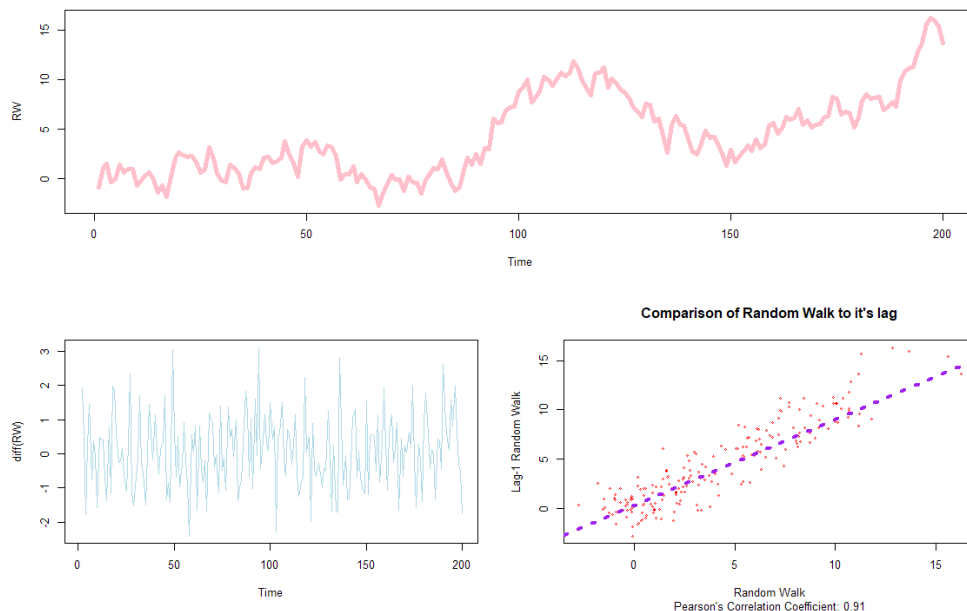
| Variable | Description |
|------------------------|---|
| t | The relevant point in the Time Series |
| Z_t | The Time Series Value (Response/Observation) |
| ε_t | The White noise for the corresponding time value. |
| σ_ε^2 | The variance of the White Noise |
| ϕ | The slope of the last observation used in an AR process |
| θ | The slope of the last error used in an MA process |
| μ | Is the mean value |

Autocorrelation⁴⁸

Autocorrelation helps us see how each time series observation is related to its recent past, (e.g. temperature would have autocorrelation, but rolling a die would not).

Processes with greater autocorrelation are more predictable than those with none.

This can be automated with the `acf()` function, (the `acf()` function uses $1/n$ which is preferable in time series, whereas `cor()` uses $\frac{1}{n-1}$ which adjusts for sampling bias.)



White Noise

White noise can be created in R with the `ARIMA` package:

⁴⁸ **David S. Matteson**

Assistant Professor at Cornell University
Introduction to Time Series, Data Camp

```
WN_1 <- arima.sim(model=list(order=c(0,0,0)), n=50)
as.xts(WN_1)
#To specify mean value or sd:
WN_1 <- arima.sim(model=list(order=c(0,0,0)), n=50, mean=0,
sd=1)
```

A white noise model can be fitted to some time series (Z) data by:

```
arima(y, order=c(0,0,0))
```

Random Walk

A random walk is:

$$\text{Today} = \text{Yesterday} + \text{Noise}$$

More formally:

$$Y_t = Y_{t-1} + \varepsilon_t$$

It is an AR process with $\phi = 1$

Simulating in R

White noise can be created in R with the `cumsum()` command, or,

White noise can be created in R with the `ARIMA` package:

```
RW_1 <- arima.sim(model=list(order=c(0,1,0)), n=50)
as.xts(RW_1)
#To specify mean value or sd:
RW_1 <- arima.sim(model=list(order=c(0,1,0)), n=50, mean=0,
sd=1)
```

A Random Walk model can be fitted to some time series (Z) data by:

```
arima(y, order=c(0,1,0))
```

Autoregressive Process (AR)

A time series is said to be an autoregressive process of order p if it is a weighted linear sum of the past p values plus a random shock so:⁴⁹ 3.11

A first order AR Process **AR(1)** is:

$$Today = Slope \times Yesterday + Constant + Noise$$

More formally:

$$Y_t = \phi \cdot Y_{t-1} + \varepsilon_t$$

A second order AR process (**AR(2)**) is:

$$Y_t = \phi_{t-1} \cdot Y_{t-1} + \phi_2 \cdot Y_{t-2} + \varepsilon_t$$

A p^{th} order AR Process (**AR(p)**) is:

$$Y_t = \phi_{t-1} \cdot Y_{t-1} + \phi_{t-2} \cdot Y_{t-2} + \phi_{t-3} \cdot Y_{t-3} \dots \phi_{t-p} \cdot Y_{t-p}$$

The focus here is on lower order Processes.

Mean Centred Version

The Mean-centred (**AR(1)**) version, which is used in **R**:

$$(Today - Mean) = Slope \times (Yesterday - Mean) + Noise$$

So formally:

$$(Y_t - \mu) = \phi(Y_{t-1} - \mu) + \varepsilon_t$$

Parameters

The mean centred autoregressive model has three parameters:

- μ the mean
- ϕ the slope
- σ_ε^2 the variance of the white noise

⁴⁹ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.1

Stationarity and Auto-correlation for AR(1) and AR(2)

AR(1)

An AR(1) process:

$$Y_t = \phi \cdot Z_{t-1} + \varepsilon_t$$

Auto Correlation

Has an Auto-correlation function:

$$\rho_k = \phi^k, \text{ for } k = 0, 1, 2, 3 \dots$$

Stationarity

Is stationary if and only if:⁵⁰
 $|\phi| < 1$

AR(2)

An AR(2) process:

$$Y_t = \phi_{t-1}Y_{t-1} + \phi_2Y_{t-2} + \varepsilon_t$$

Auto Correlation

Has an Auto-correlation function:

$$\begin{aligned}\rho_1 &= \frac{\phi_1}{1 - \phi_2} \\ \rho_2 &= \phi_1 \cdot \rho_1 + \phi_2 \rho_0 \\ &= \frac{\phi_2(1 - \phi_2) + \phi_1^2}{(1 - \phi_2)} \\ &\dots \\ \rho_k &= \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2}\end{aligned}$$

Stationarity

Is stationary if and only if all three conditions are true:

$$\begin{aligned}(\phi_1 + \phi_2) &< 1 \\ (\phi_2 - \phi_1) &< 1 \\ |\phi_2| &< 1\end{aligned}$$

⁵⁰ It is technically more accurate to say that there is a unique stationary solution of $X_t = \phi \cdot X_{t-1} + \varepsilon$ which is causal provided that $|\phi| < 1$, refer to 3.1.4 of Chatfield (2000).

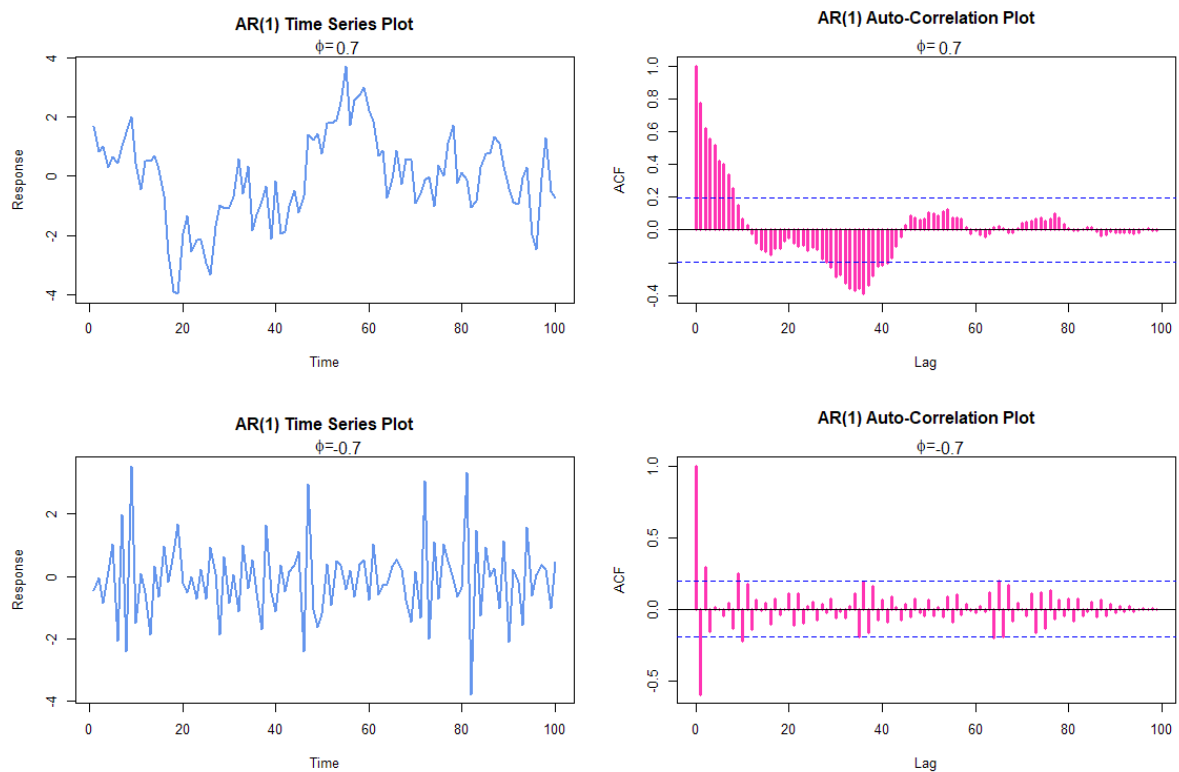
Plots

AR(1)

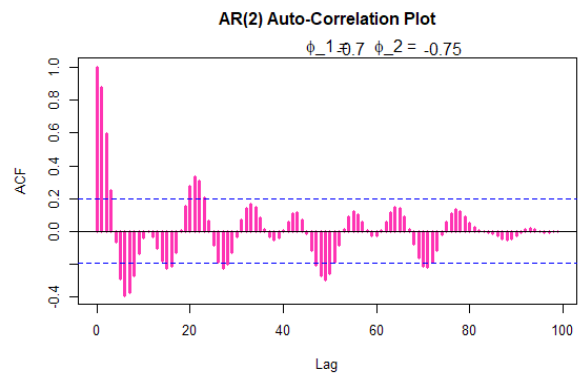
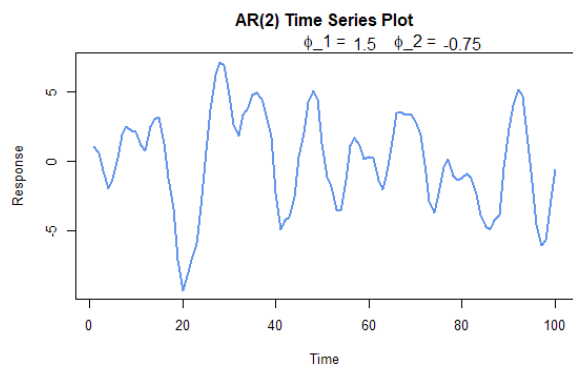
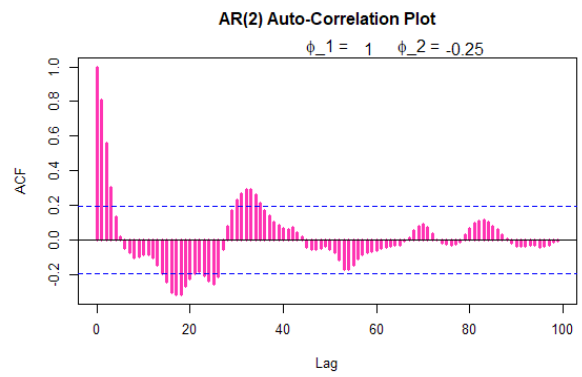
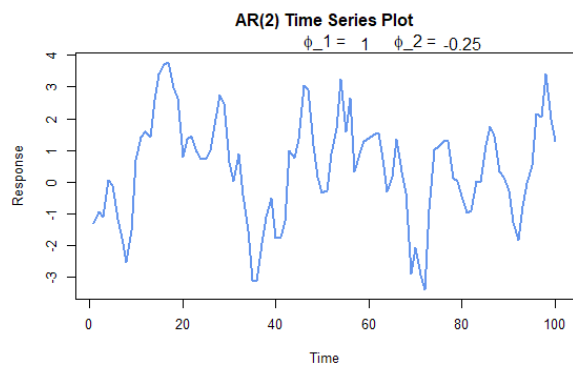
Slope

- Zero slope $\phi = 0$
 - A slope of zero ($\phi = 0$) is a white noise function.
 - A non-zero slope ($\phi \neq 0$) means Y_t is dependent on ε_t and Y_{t-1}
- A slope of 1 ($\phi = 1$) is a random walk function.
- Larger slope values (ϕ) imply larger auto correlation
- Negative ϕ values results in oscillatory behaviour
 - Because a preceeding positive value will result in a negative value following and vice versa

Observe that a larger ϕ leads to greater persistence and a negative ϕ leads to oscillatory behaviour:



AR(2)



Simulating Models in **R**

AR(1)

The `arima.sim()` function can simulate a first order autoregressive function like so:

```
arima.sim(model = list(ar=phi), n)
```

```
# Simulate an AR model with 0.5 slope
x <- arima.sim(model = list(ar=0.5), n = 100)

# Simulate an AR model with 0.9 slope
y <- arima.sim(model=list(ar=0.9), n=100)

# Simulate an AR model with -0.75 slope
z <- arima.sim(model=list(ar=-0.75), n=100)

# Plot your simulated data
plot.ts(cbind(x,y,z), main="Autoregressive Models")
```

AR(2)

The `arima.sim()` function can simulate a first order autoregressive function like so:

```
arima.sim(model = list( ar=c(phi1, phi2) ), n)
```

```
# Simulate an AR model with 0.5, 0.25 slope
x <- arima.sim(model = list(ar=c(0.5, 0.25), n = 100)

# Simulate an AR model with 0.9, 0.8 slope
y <- arima.sim(model=list(ar=c(0.9, 0.8)), n=100)

# Simulate an AR model with -0.75, -0.25 slope
z <- arima.sim(model=list(ar=(-0.75, -0.25)), n=100)

# Plot your simulated data
plot.ts(cbind(x,y,z), main="Autoregressive Models")
```


Moving Average Process

A time series is said to be a moving average process of order q if it is a weighted linear sum of the last q random shocks so that:⁵¹

A first order MA Process **MA(1)** is:

$$Today = Mean + Noise + Slope \times (Yesterday's\ Noise)$$

More formally:

$$Y_t = \mu + \varepsilon_t + \theta \cdot \varepsilon_{t-1}$$

A first order MA Process **MA(1)** in the lecture notes is provided as:

$$Today = Noise + Slope \times (Yesterday's\ Noise)$$

More formally:

$$Y_t = \theta \cdot \varepsilon_{t-1} + \varepsilon_t$$

A second order MA process (**MA(2)**) is:

$$Y_t = \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \varepsilon_t$$

A p^{th} order MA Process (**MA(q)**) is:⁵²

$$Y_t = \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \theta_3 \cdot \varepsilon_{t-3} + \dots \theta_q \cdot \varepsilon_{t-q} + \varepsilon_t$$

Some authors use the equation of the following equivalent form:⁵³

$$Y_t = \varepsilon_t - \theta_1 \cdot \varepsilon_{t-1} - \theta_2 \cdot \varepsilon_{t-2} - \theta_3 \cdot \varepsilon_{t-3} \dots - \theta_q \cdot \varepsilon_{t-q}$$

The focus here is on lower order Processes.

Parameters

There are three parameters to a moving average process:

- μ is the mean value
- θ is the slope value
- σ_ε^2 is the White noise variance.

⁵¹ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.2

⁵² Applied Time Series Analysis. (2017). *Moving Average Models (MA models)*. [online] Available at: <https://onlinecourses.science.psu.edu/stat510/node/48> [Accessed 27 Aug. 2017], see also *Data Camp*.

⁵³ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.2 p. 45 of 265.

Stationarity and Auto-correlation for MA(1) and MA(2)

A $MA(q)$ process has auto correlation of 0 for lags greater than q

MA(1)

An MA(1) process:

$$Y_t = \theta \cdot \varepsilon_{t-1} + \varepsilon_t$$

Auto Correlation

Has an Auto-correlation function:

$$\begin{aligned}\rho_0 &= 1 \\ \rho_1 &= \frac{\theta}{1 + \theta^2} \\ \rho_k &= 0, \quad k > 1\end{aligned}$$

Stationarity

$MA(q)$ process always stationary.

MA(2)

An MA(2) process:

$$Y_t = \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \varepsilon_t$$

Auto Correlation

Has an Auto-correlation function:

$$\begin{aligned}\rho_0 &= 1 \\ \rho_1 &= \frac{\theta_1 + \theta_1 \cdot \theta_2}{1 + \theta_1^2 + \theta_2^2} \\ \rho_2 &= \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2} \\ \rho_k &= 0, \quad k \geq 3\end{aligned}$$

Stationarity

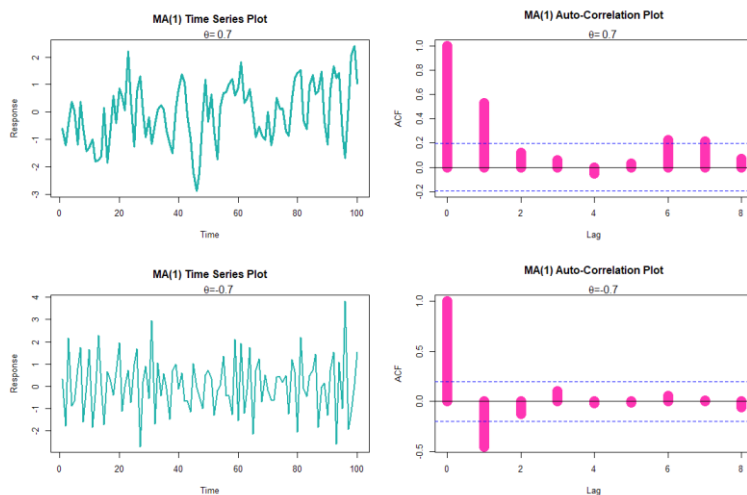
$MA(q)$ process always stationary.

Plots

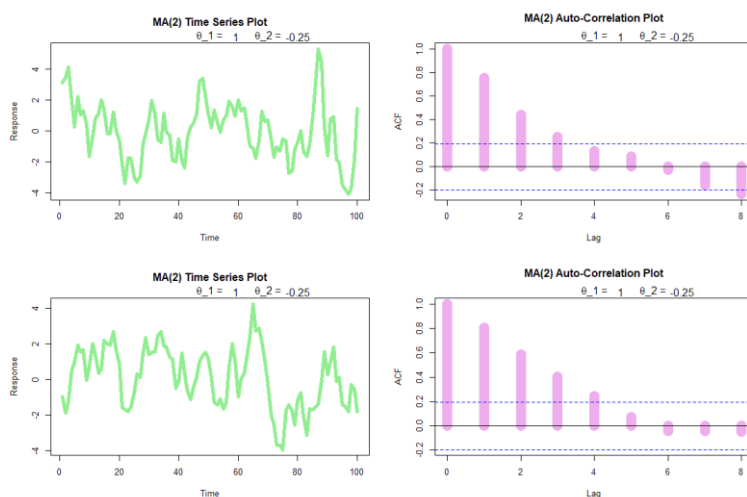
MA(1)

Slope

- $\theta = 0$
 - If the slope is zero, Y_t is just a white noise process
 - $\theta = 0 \Rightarrow Y_t$ is WN
 - If the slope is non-zero, Y_t depends on the current and previous noise
 - $\theta \neq 0 \Rightarrow Y_t = f(\varepsilon_t, \varepsilon_{t-1})$
 - Larger values of θ lead to greater auto correlation
 - Negative values of θ result in oscillatory time series.



MA(2)



Simulating in R

MA(1) A moving average process can be simulated with the `arima.sim()` command:

```
arima.sim(model = list(ma = theta), n)
```

```
# MA Process -----  
  
# Generate MA model with slope 0.5  
x <- arima.sim(model = list(ma = 0.5), n = 100)  
  
# Generate MA model with slope 0.9  
y <- arima.sim(model=list(ma=0.9), n=100)  
  
# Generate MA model with slope -0.5  
z <- arima.sim(model=list(ma=-0.5), n=100)  
  
# Plot all three models together  
plot.ts(cbind(x, y, z))
```

MA(2)

```
arima.sim(model = list(ma = c(theta1, theta2) ), n)
```

```
# MA Process -----  
  
# Generate MA model with slopes 0.5 and 0.25  
x <- arima.sim(model = list(ma = c(0.5, 0.25) ), n = 100)  
  
# Generate MA model with slopes 0.9 and 0.8  
y <- arima.sim(model = list(ma = c(0.9, 0.8) ), n = 100)  
  
# Generate MA model with slopes -0.5 and -0.2  
z <- arima.sim(model=list(ma=c(-0.5, -0.2), n=100)  
  
# Plot all three models together  
plot.ts(cbind(x, y, z))
```

Mixed Autoregressive-Moving Average Process (ARMA)

A mixed autoregressive moving average process **ARMA** is the combination of an AR and MA process:

An **ARMA(1,1)** process is:

$$\text{Today} = \text{Slope} \times \text{Yesterday} + \text{Slope} \times (\text{Yesterday's Noise}) + \text{Noise}$$

More formally:

$$Y_t = \phi \cdot Y_{t-1} + \theta \cdot \varepsilon_{t-1} + \varepsilon_t$$

An **ARMA(2,2)** process is:

$$Y_t = \phi_1 \cdot Y_{t-1} + \phi_2 \cdot Y_{t-2} + \theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \varepsilon_t$$

An **ARMA(p,q)** process can be expressed as a linear combination⁵⁴ or a summation:⁵⁵

$$\begin{aligned} Y_t &= (\phi_{t-1} \cdot Y_{t-1} + \phi_{t-2} \cdot Y_{t-2} + \dots \phi_{t-p} \cdot Y_{t-p}) + (\theta_1 \cdot \varepsilon_{t-1} + \theta_2 \cdot \varepsilon_{t-2} + \dots \theta_q \cdot \varepsilon_{t-q}) + \varepsilon_t \\ Y_t &= \sum_{j=1}^p [\phi_j \cdot Y_{t-j}] + \sum_{j=1}^q [\theta_j \cdot \varepsilon_{t-j}] + \varepsilon_t \\ \text{ARMA}(p, q) &= \text{AR}(P) + \text{MA}(q) \end{aligned}$$

Some authors use the equation of the following equivalent form:⁵⁶

$$Y_t = (\phi_{t-1} \cdot Y_{t-1} + \phi_{t-2} \cdot Y_{t-2} \dots \phi_{t-p} \cdot Y_{t-p}) + \varepsilon_t - \theta_1 \cdot \varepsilon_{t-1} - \theta_2 \cdot \varepsilon_{t-2} \dots - \theta_q \cdot \varepsilon_{t-q}$$

The focus here is on lower order Processes.

Parameters

There are three parameters to a moving average process:

- μ is the mean value
- θ the slope value for error
- ϕ the slope value for previous observations
- σ_ε^2 is the White noise variance.

⁵⁴ Stoffer, D. (2017). *Time Series Analysis*. Free Dog Publishing, p.60.

⁵⁵ Huerta, G. (2017). *AR, MA and ARMA models*. [online] UNM; Department of Mathematics and Statistics. Available at: http://www.math.unm.edu/~ghuerta/tseries/week4_1.pdf [Accessed 30 Aug. 2017].

⁵⁶ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.2 p. 45 of 265.

Stationarity and Auto-correlation for ARMA(1,1) processes

An ARMA(1,1) process:

$$Y_t = \phi \cdot Y_{t-1} + \theta \cdot \varepsilon_{t-1} + \varepsilon_t$$

Auto Correlation

Has an Auto-correlation function:

$$\rho_k = \frac{(1 - \theta\phi) \cdot (\phi - \theta)}{\theta^2 - 2\theta\phi + 1} \times \phi^{k-1}$$

Stationarity

If the AR and MA components are stationary.

Auto-Regressive Integrated Moving Average Process (ARIMA)

A time series exhibits ARIMA behaviour if the differenced data has ARMA behaviour.

Many processes are non-stationary so AR, MA or ARMA processes cannot not be used directly. One way of handling non-stationary series is to apply **differencing** to make the series stationary, the first differences may further be differenced and so-on until the data is stationary and then an ARMA process can be applied.⁵⁷

Differencing is analogous to differentiating, imagine that the second order differential of a quadratic function is a stationary horizontal line, if we can model the stationary data we may later integrate in order to make interpretations relevant to the initial data.

Difference Operator

Define the function ∇ :

$$\nabla(Y_t) = Y_t - Y_{t-1}$$

This will be a new time series corresponding to the differences between each observation

The second order difference would be defined as:

$$\begin{aligned}\nabla^2 &= \nabla(\nabla(Y_t)) \\ &= (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2})\end{aligned}$$

And so on.

The ARIMA(p,q) process

An 'autoregressive integrated moving average process' (**ARIMA(p, d, q)**) is a time series Y_t if the function:

$$\nabla^d(Y_t)$$

Is a stationary **ARMA(p,q)** process.

Exercise 8.1

Sketch the acfs for each of the following ARMA models:

⁵⁷ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.5

- I. AR(2) with
 - a. $\phi = 1.2$
 - b. $\phi_2 = -0.7$
- II. MA(2) with
 - a. $\theta_1 = 1.2$
 - b. $\theta_2 = -0.7$
- III. ARMA(1,1) with
 - a. $\phi = 0.7$
 - b. $\theta = 0.4$

Estimation/Sketch

Second Order Auto Regressive:

Recall that:

$$\rho_1 = \frac{\phi_1}{1 - \phi_2}$$

$$\begin{aligned} \rho_2 &= \phi_1 \cdot \rho_1 + \phi_2 \rho_0 \\ &= \frac{\phi_2(1 - \phi_2) + \phi_2^2}{(1 - \phi_2)} \end{aligned}$$

...

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2}$$

$$\rho_1 = \frac{\phi}{1 - \phi_2} = \frac{1.2}{1 + 0.7} = 0.7$$

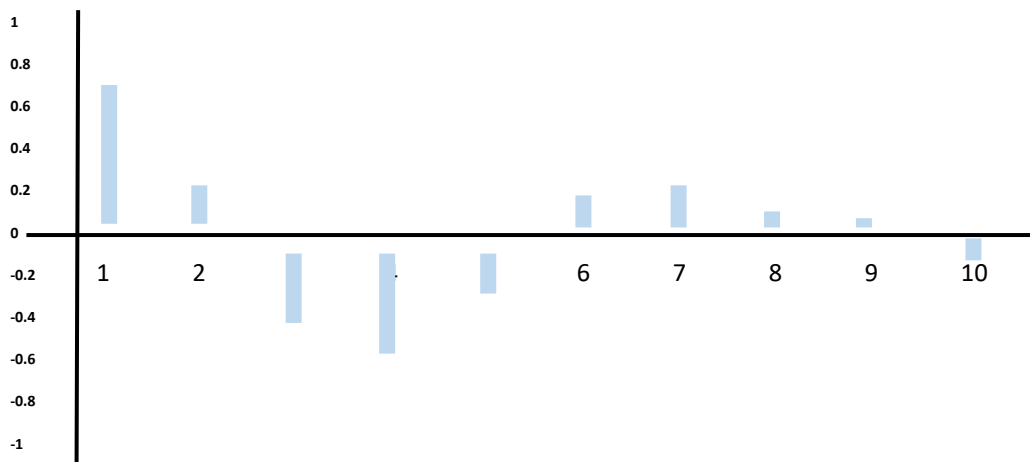
$$\begin{aligned} \rho_2 &= \phi_1 \cdot \rho_1 + \phi_2 \rho_0 = 1.2 \times 0.7 - 0.7 \\ &= 0.14 \end{aligned}$$

$$\begin{aligned} \rho_3 &= \phi_1 \times \rho_2 + \phi_2 \times \rho_1 \\ &= 1.2 \times 0.14 - 0.7 \times 0.7 \\ &= -0.3 \end{aligned}$$

$$\begin{aligned} \rho_4 &= \phi_1 \times \rho_3 + \phi_2 \times \rho_2 \\ &= 1.2 \times -0.3 - 0.7 \times 0.14 \\ &= -0.458 \end{aligned}$$

$$\begin{aligned} \rho_5 &= \phi \times \rho_4 + \phi_2 \times \rho_3 \\ &= 1.2 \times -0.458 + 0.7 \times -0.3 \\ &= -0.934 \end{aligned}$$

Thus we would expect the acf plot to look something like this:



Second Order Moving Average

Recall that

$$\rho_0 = 1$$

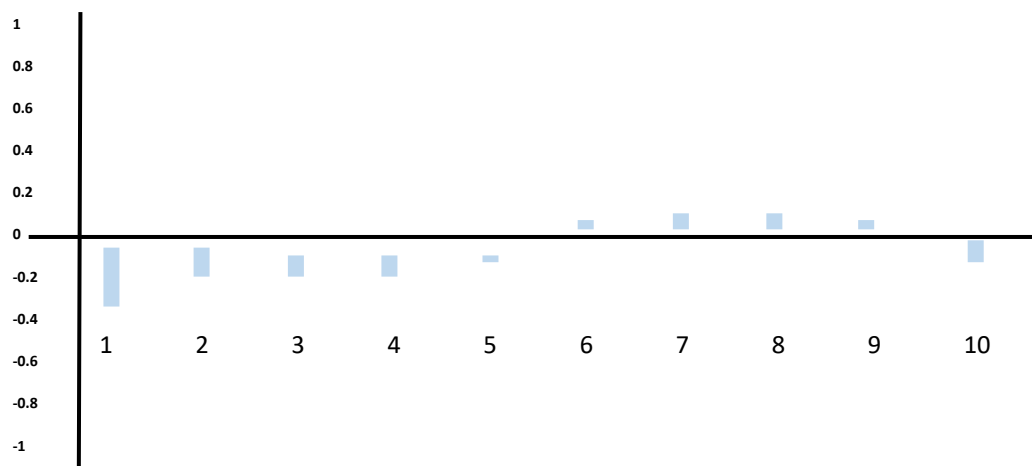
$$\begin{aligned} \rho_1 &= \frac{\theta_1 + \theta_1 \cdot \theta_2}{1 + \theta_1^2 + \theta_2^2} \\ &= \frac{1.2 - 1.2 \times 0.7}{1 + 1.2^2 - 0.7} \\ &= -0.40 \end{aligned}$$

$$\rho_1 = \frac{\theta_1 + \theta_1 \cdot \theta_2}{1 + \theta_1^2 + \theta_2^2}$$

$$\rho_2 = \frac{\theta_2}{1 + \theta_1^2 + \theta_2^2}$$

$$\rho_k = 0, \quad k \geq 3$$

Thus we would expect the acf plot to look something like this:



ARMA(1,1) Process

Recall that:

$$\rho_k = \frac{(1 - \theta\phi) \cdot (\phi - \theta)}{\theta^2 - 2\theta\phi + 1} \times \phi^{k-1}$$

Thus by substitution

$$\begin{aligned}\rho_k &= \frac{(1 - \theta\phi) \cdot (\phi - \theta)}{\theta^2 - 2\theta\phi + 1} \times \phi^{k-1} \\ &= \frac{(1 - 0.7 \times 0.4) \cdot (0.7 - 0.4)}{0.7^2 - 2 \times 0.7 \times 0.4 + 1} \times 0.7^{k-1} \\ &= 0.36 \times 0.7^{k-1}\end{aligned}$$

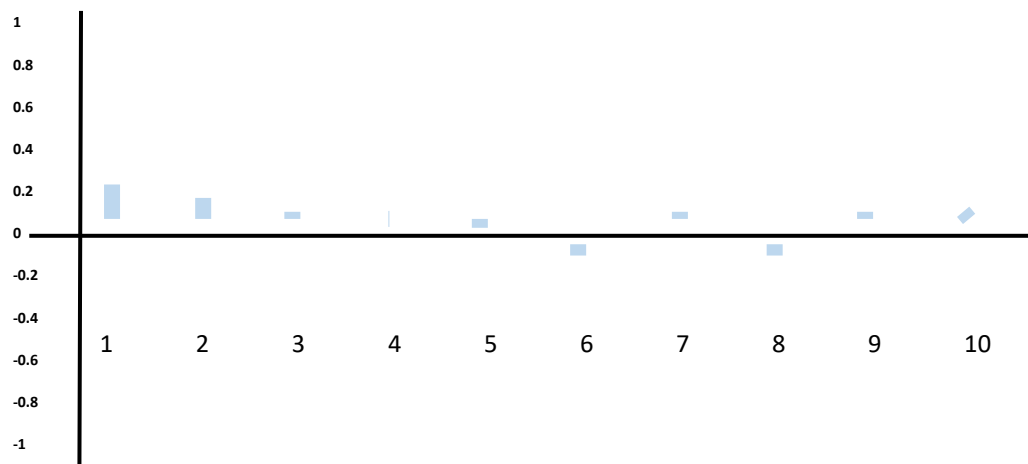
$$\rho_1 = 0.232 \times 0.7^0 = 0.36$$

$$\rho_2 = 0.232 \times 0.7^1 = 0.25$$

$$\rho_3 = 0.232 \times 0.7^2 = 0.18$$

$$\rho_4 = 0.232 \times 0.7^3 = 0.12$$

Thus the acf plot would look something like this:



Although the later plot does not agree with this formula and I have triple checked...

Generate the Plots in R

```
# Create Plots of AR(2) and MA(2) -----

# AR(2) Plots -----

plotar2 <- function(phi1=1.2, phi2=-0.7, n=100){
x <- arima.sim(model = list(ar=c(phi1, phi2)), n = n)
ts.plot(x, ylab="AR(2) Response", col="palegreen2", lwd=5, lty=1, main="AR(2)
Time Series Plot")
mtext(expression(paste(phi, "_1 =")), adj=0.5)
mtext(paste(phi1), adj = 0.585)
mtext(expression(paste(phi, "_2 =")), adj=0.7)
mtext(paste(phi2), adj = 0.8)

acf(x, col="plum2", lwd=5, main="AR(2) Auto-Correlation Plot")
mtext(expression(paste(phi, "_1 =")), adj=0.5)
mtext(paste(phi1), adj = 0.585)
mtext(expression(paste(phi, "_2 =")), adj=0.7)
mtext(paste(phi2), adj = 0.8)
}

# MA(2) Plots -----

plotma2 <- function(theta1=1.2, theta2=-0.7, n=100){
x <- arima.sim(model = list(ma=c(theta1, theta2)), n = n)
ts.plot(x, ylab="MA(2) Response", col="palegreen2", lwd=5, lty=1, main="MA(2)
Time Series Plot")
mtext(expression(paste(theta, "_1 =")), adj=0.5)
mtext(paste(theta1), adj = 0.585)
mtext(expression(paste(theta, "_2 =")), adj=0.7)
mtext(paste(theta2), adj = 0.8)

acf(x, lag.max = 8, col="plum2", lwd=15, main="MA(2) Auto-Correlation Plot")
mtext(expression(paste(theta, "_1 =")), adj=0.5)
mtext(paste(theta1), adj = 0.585)
mtext(expression(paste(theta, "_2 =")), adj=0.7)
mtext(paste(theta2), adj = 0.8)
}

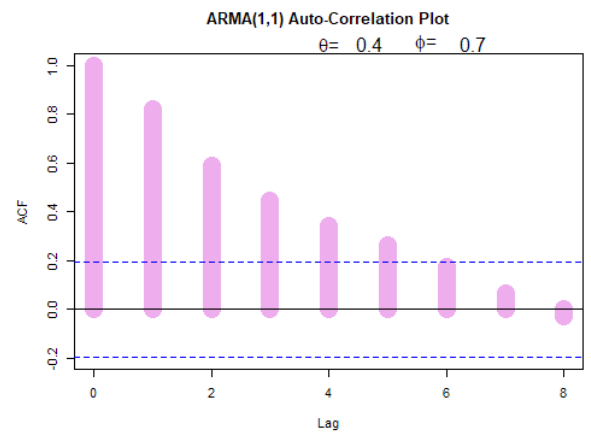
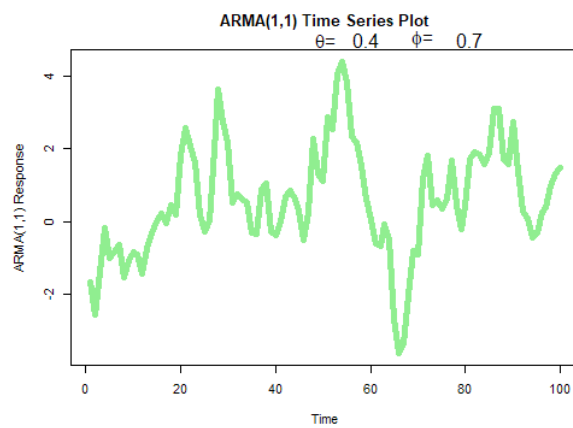
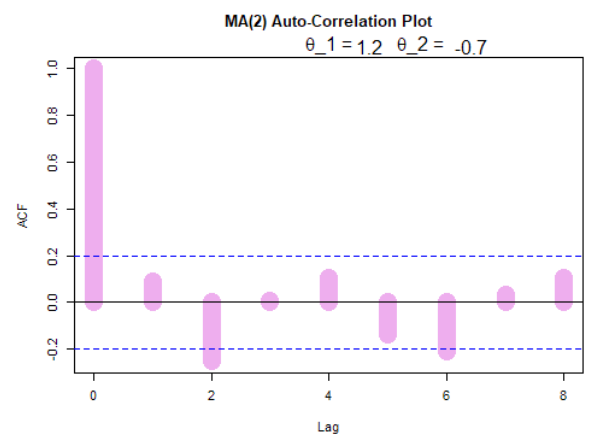
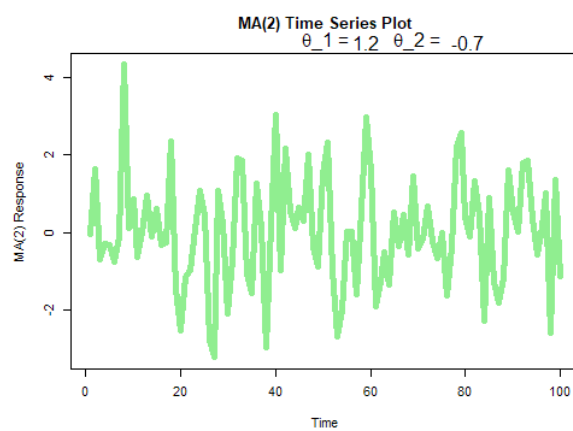
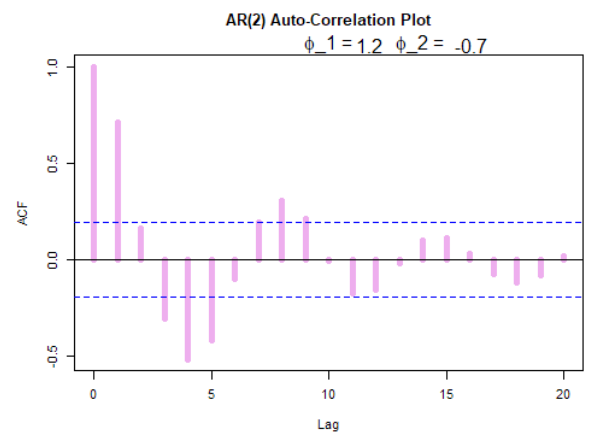
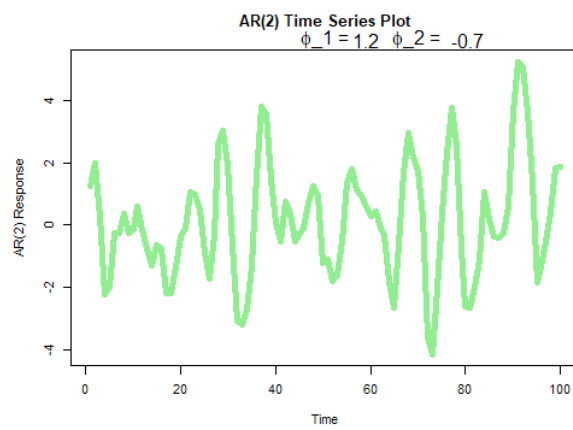
# ARMA (1,1) Plot -----

plotarmall <- function(phi=0.7, theta=0.4, n=100){
x <- arima.sim( model = list(order = c(1,0,1),ar=phi, ma=theta ),n=n
)
ts.plot(x, ylab="ARMA(1,1) Response", col="palegreen2", lwd=5, lty=1,
main="ARMA(1,1) Time Series Plot")
mtext(expression(paste(theta, "=")), adj=0.5)
mtext(paste(theta), adj = 0.585)
mtext(expression(paste(phi, "=")), adj=0.7)
mtext(paste(phi), adj = 0.8)

acf(x, lag.max = 8, col="plum2", lwd=15, main="ARMA(1,1) Auto-Correlation
Plot")
mtext(expression(paste(theta, "=")), adj=0.5)
mtext(paste(theta), adj = 0.585)
mtext(expression(paste(phi, "=")), adj=0.7)
mtext(paste(phi), adj = 0.8)
}

# Create the plots -----
layout(matrix(1:6, byrow=TRUE, ncol=2))
plotar2()
plotma2()
plotarmall()
layout(matrix(1))
```

Display the Plots



Fitting ARIMA Models and Estimation

Week 10 Material | Lecture 9 Material | Material of 19th Sep | Exercises Due: 26 Sept.

The **autocorrelation (acf)** and **partial autocorrelation (pacf)** can be used to discern the difference between an AR, MA and ARMA models.

Model Choice

In Time series analysis a few models may seem reasonable (e.g. exponential vs. quadratic or MA(1) or AR(1)).

In **R** the two most common tests to decide between models in are the **AIC** and **BIC** tests and where those tests conflict the simpler model is usually chosen.

The parsimony principle suggests that the simplest explanation that fits the evidence should be chosen, like Occam's razor.

Using the `sarima()` function (included with the `astsa`) the AIC and BIC values are automatically generated.

ACF and PACF test

AR, MA and ARMA models look very similar and a model can't be determined simply by looking at the data.

The **autocorrelation function (acf)** and **partial autocorrelation function (pacf)** are used to determine the model orders.

The acf and pacf can be calculated/plotted in **R** using the `acf()` and `pacf()` function or by using the `acf2()` function (included with the `astsa` package).

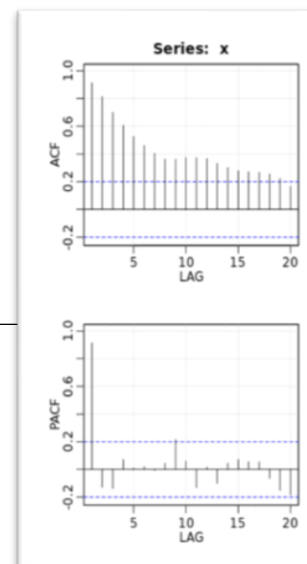
The acf and pacf can be used thusly:⁵⁸

| | $AR(p)$ ⁵⁹ | $MA(q)$ ⁶⁰ | $ARMA(p, q)$ |
|-------------|-----------------------|-----------------------|--------------|
| acf | Tails off | Cuts off lag q | Tails off |
| pacf | Cuts off lag p | Tails off | Tails off |

```
# Generate 100 observations from the AR(1) model
x <- arima.sim(model = list(order = c(1, 0, 0), ar = .9), n = 100)
```

```
# Plot the generated data
plot(x)
```

```
# Plot the sample P/ACF pair
#Built-in
Layout(matrix(ncol=2, data=1:2))
acf(x)
pacf(x)
# astsa package
require(astsa)
acf2(x)
```



As the acf tails off but the last significant pacf value occurs before 1, we will model an AR(1) time series (ARMA(1,1)=ARIMA(1,0,0)).

```
#Generate an AR(1) model for the data
sarima(x, p=1, q=0, d=0)
```

⁵⁸ Stoffer, D. (2017). *Time Series Analysis*. Free Dog Publishing, p.80.

⁵⁹ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.2 p. 45 of 265.

⁶⁰ Chatfield, C. (2000). *Time-series forecasting*. Boca Raton: Chapman & Hall/CRC, 3.1.3 p. 46 of 265.

How to fit a Model to data that may follow an ARIMA Process

Plotting the Data

Say we have some data that could be modelled with an ARIMA process:

(For example the dataset `oil` in the `astsa` package or the residuals between a multiple linear regression and `co2` levels as in the lecture notes.)

```
require(astsa)

## Loading required package: astsa

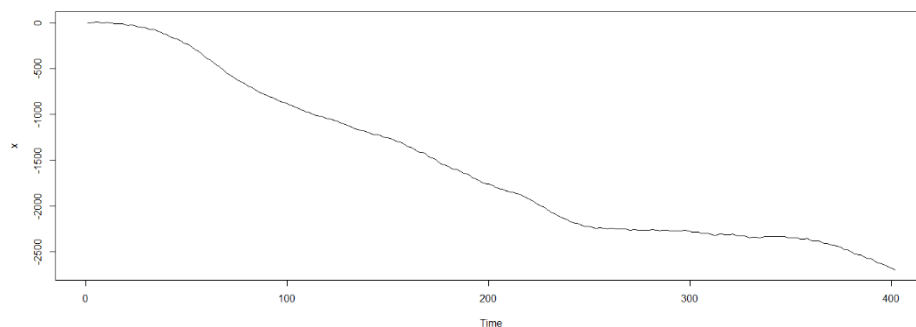
n <- 400

# 1. Create a time Series -----
x <- arima.sim(model=list(order=c(2,2,0), ar=c(0.8, -0.9)), n=n)
as.matrix(ncol=1, head(x))

##           [,1]
## [1,]  0.000000
## [2,]  0.000000
## [3,] -1.114615
## [4,] -3.500192
## [5,] -5.911657
## [6,] -8.029168
```

Our first step would be to plot the time series:

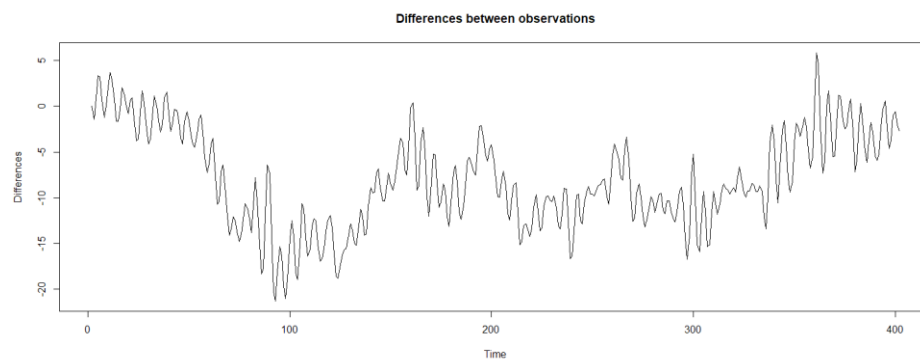
```
# Plot the Time Series -----
ts.plot(x)
```



Differencing the Data

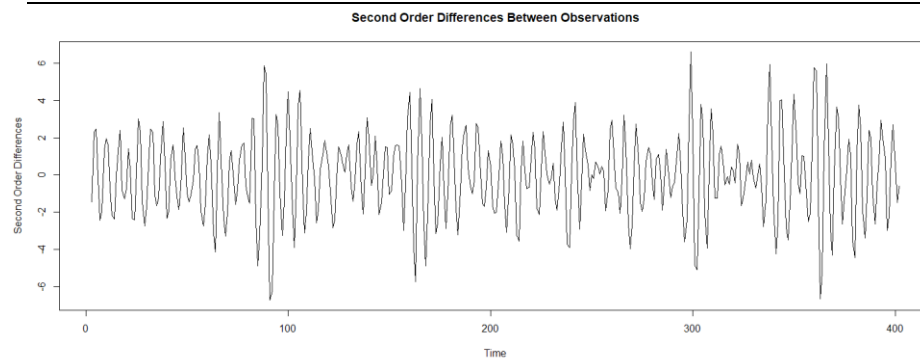
Observe that this time series is clearly not stationary because it is moving down over time, we will take the differences between observations and see if that time series is stationary:

```
# Plot the difference (1) -----  
-----  
x_diff.1 <- diff(x)  
plot(x_diff.1)
```



Now this time-series is also not stationary, but it is a little more stationary than the one before it, we will take the difference again to see if the next time series is stationary.

```
# Plot the difference (2) -----  
-----  
x_diff.2 <- diff(x_diff.1)  
plot(x_diff.2)
```



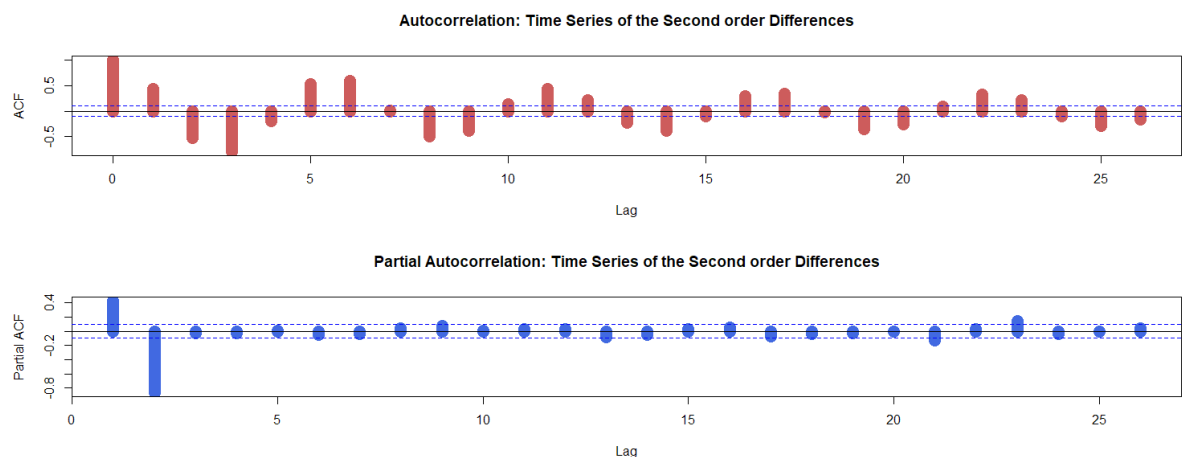
Now this time series appears stationary, so now we can fit an ARMA model to it.

It is unwise to difference further than necessary as it creates a more complicated model. A more complicated model is undesirable because it is contrary to the parsimony principle (like *Occam's Razor*) it's analogous to how any time series data can be modelled with a polynomial of degree 900, just because it fits does not mean it's an appropriate model, we saw this in Wk. 7, there were models that fit the data really well, but the residual analysis showed that they were not appropriate models.

Deciding on an ARMA Model

First generate ACF/PACF plots and analyse them to decide what model (AR(p), MA(q), ARMA(p,q)) best fits the plot of the differences:

```
# Decide on a Model -----  
#With the stationary data, analyse using acf/pacf plots  
  
#Method 1  
require(astsa)  
acf2(x_diff.2)  
  
#Method 2  
layout(matrix(nrow=2, 1:2))  
acf(x_diff.2)  
pacf(x_diff.2)
```



Observe that the ACF tapers towards zero while the pacf is cut off at the second lag, refer to the table:

Deciding on an ARIMA Model

This suggests that the ARMA model is a second order AR process (AR(2)), this is equivalent to an ARMA(2,0).

As we have taken 2 differences from the original data, we can model our data after a second difference ARIMA.

So our model is an ARIMA(2,2,0):

$$\begin{array}{l} AR(2) \wedge ARMA(2,0) \Rightarrow p = 2 \\ 2 \text{ Differences} \Rightarrow d = 2 \\ MA(0) \wedge ARMA(2,0) \Rightarrow q = 0 \end{array} \Rightarrow ARMA(p, d, q) = ARMA(2,2,0)$$

Fitting an ARIMA Model and Residual Analysis

The `sarima()` function automatically:

- Fits the model
- Calculates BIC and AIC values
- Generates Residuals

That makes it quite ideal, however, it is also important to understand how to use the **R** built-in functions.

With the `astsa` package and the `sarima()` function

Because we know that our model could be modelled by an ARIMA(2,2,0) we can create and compare a model like so:

```
sarima(x, p=2, d=2, q=0)
...
## Coefficients:
##          ar1          ar2
##          0.7907   -0.8527
## s.e.    0.0258    0.0256
##
## sigma^2 estimated as 0.9335:  log likelihood = -555.2,  aic = 1116.41
##
## $degrees_of_freedom
## [1] 400
##
## $ttable
##      Estimate      SE  t.value p.value
## ar1   0.7907 0.0258  30.6175      0
## ar2  -0.8527 0.0256 -33.3641      0
##
## $AIC
## [1] 0.9410896
##
## $AICc
## [1] 0.9462147
##
## $BIC
## [1] -0.
```

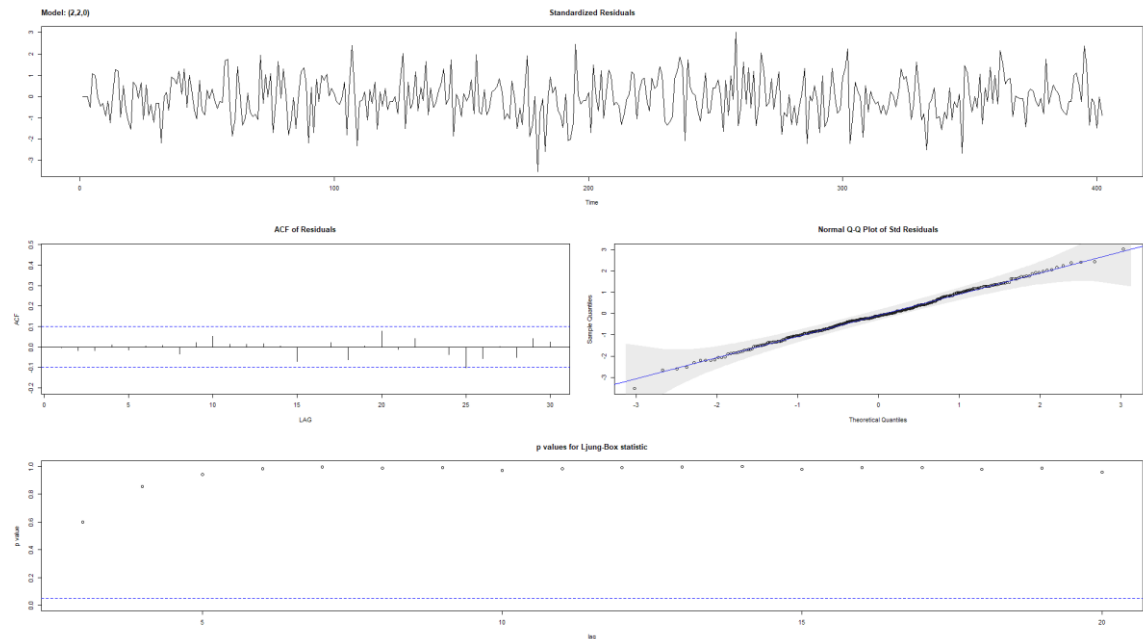
Thus a model for our 2nd order differences is:

$$\begin{aligned}\nabla(\nabla X)_t &= \nabla^2 X_t \\ &= D(D(X_t)) \\ &= \phi_1 \cdot X_{t-1} + \phi_2 \cdot X_{t-2} + \varepsilon_t \\ &= 0.7907 \cdot X_{t-1} - 0.8527 \cdot X_{t-2} + \varepsilon_t\end{aligned}$$

Model/Residual Analysis

The following plot is automatically generated via the above command:

```
sarima(x, p=2, d=2, q=
```



The AIC and BIC are low numbers and the residuals look to be normally distributed, thus the model is an appropriate fit for the data.

If all the available information in the data was captured by the model, the residuals would look like white noise.

Observe the Ljung-Box has very high p-values, this indicates that there is a high probability of incorrectly rejecting the null hypothesis that the residuals are normally distributed

(Rightly so given that the residuals are actually simulated normally distributed white noise via the `arima.sim()` and `sarima()` functions)

With the built-in functions

Because we know that our model could be modelled by an ARIMA(2,2,0) we can create and compare a model like so:

```
x_model <- arima(x, order=c(2, 2, 0))
x_model

##
## Call:
## arima(x = x, order = c(2, 2, 0))
##
## Coefficients:
##          ar1          ar2
##       0.7907   -0.8527
## s.e.  0.0258    0.0256
##
## sigma^2 estimated as 0.9335:  log likelihood = -555.2,  aic = 1116.41
```

Thus a model for our 2nd order differences is:

$$\begin{aligned}\nabla(\nabla X)_t &= \nabla^2 X_t \\ &= D(D(X_t)) \\ &= \phi_1 \cdot X_{t-1} + \phi_2 \cdot X_{t-2} + \varepsilon_t \\ &= 0.7907 \cdot X_{t-1} - 0.8527 \cdot X_{t-2} + \varepsilon_t\end{aligned}$$

Residual Diagnostics

The residual diagnostics must be plotted manually (sarima() function), it can be useful to rap them in a function like so:

```
# Fit the Model with Built in Functions -----
x_model <- arima(x, order=c(2, 2, 0))

arima_resid <- function(resid, p=0, q=0){

  #Residual diagnostics
  layout( matrix(nrow=3, ncol=2, byrow=1, data=c(1,1,2,3,4,4)))

  ts.plot(resid, xlab="Residuals", main="Residuals over Time", col="darkslategrey", lwd=2)
  abline(0,0, col="cadetblue", lwd=3)

  x.acf <- acf(resid, plot = 0)
  #Remove Lag 0
  x.acf$acf[1] <- NA
  plot(x.acf, lwd=13, col="cadetblue", ylim=c(-1,1))

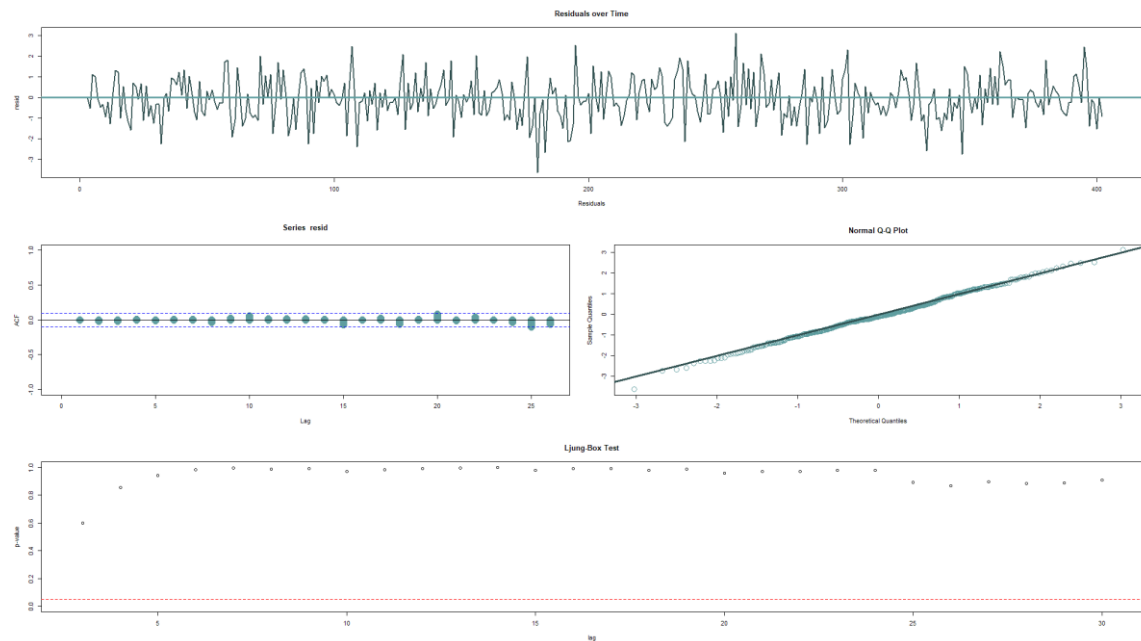
  qqnorm(y = resid, col="cadetblue", cex=2)
  abline(0,1, lwd=3, col="darkslategrey")

  require(FitAR)
  # LJBT <- LjungBoxTest(resid, k=1)
  # plot(LJBT[1:20,3], ylim=c(0,1))
  LBQPlot(resid, k=p+q)

  layout(matrix(1))
}

arima_resid(x_model$residuals, p=2, q=0)
```

Thus the plotted residuals are:



The residuals look to be normally distributed, thus the model is an appropriate fit for the data.

Observe the Ljung-Box has very high p-values, this indicates that there is a high probability of incorrectly rejecting the null hypothesis that the residuals are normally distributed

(Rightly so given that the residuals are actually simulated normally distributed white noise via the `arma.sim()` and `sarima()` functions)

Competing Models

If we had competing models, because say the acf/pacf plots were unclear, we would:

- Compare the AIC and BIC values
 - Favouring the model that has the lower values and is simpler (i.e. order 1 favoured over order 2, in line with the parsimony principle)
- Compare the residual analysis plots
 - Favouring the model with
 - more normally distributed residuals
 - Less parameters

Fitting ARIMA Models and Estimation

Week 10 Material | Lecture 9 Material | Material of 19th Sep | Exercises Due: 26 Sept.

Forecasting Values

Once a model is chosen forecasting is possible, because the model describes how the dynamics of the time series behaves over time.

Just continue the model diagnostics into the future.

For some time series:

$$Z_1, Z_2, Z_3, Z_4, Z_5 \dots$$

The **minimum mean square error forecast** is defined by:

$$\hat{Z}_t(L) = E(Z_{t+L} | Z_t, Z_{t-1}, Z_{t-2}, Z_{t-3} \dots Z_1)$$

Where t is called the **origin** of the forecast and L is the **lead time** for the forecast.

Example of Deriving forecasts

There is some theory contained in the lecture notes but I don't understand any of it at all whatsoever.

With the `forecast` package and **R** built-in

Consider the `oil` data set built into `astsa` package, start by fitting a model to that data set:

```
require(forecast)
require(astsa) #for the oil dataset

#print the data, decide on the model
matrix(head(oil), ncol=1) #Use ARIMA(1,1,1)

##      [,1]
## [1,] 26.20
## [2,] 26.07
## [3,] 26.34
## [4,] 24.95

#split up the data so the actual data can be plotted over the prediction
oil_small <- oil[208:312]
oil_future <- oil[208:364]

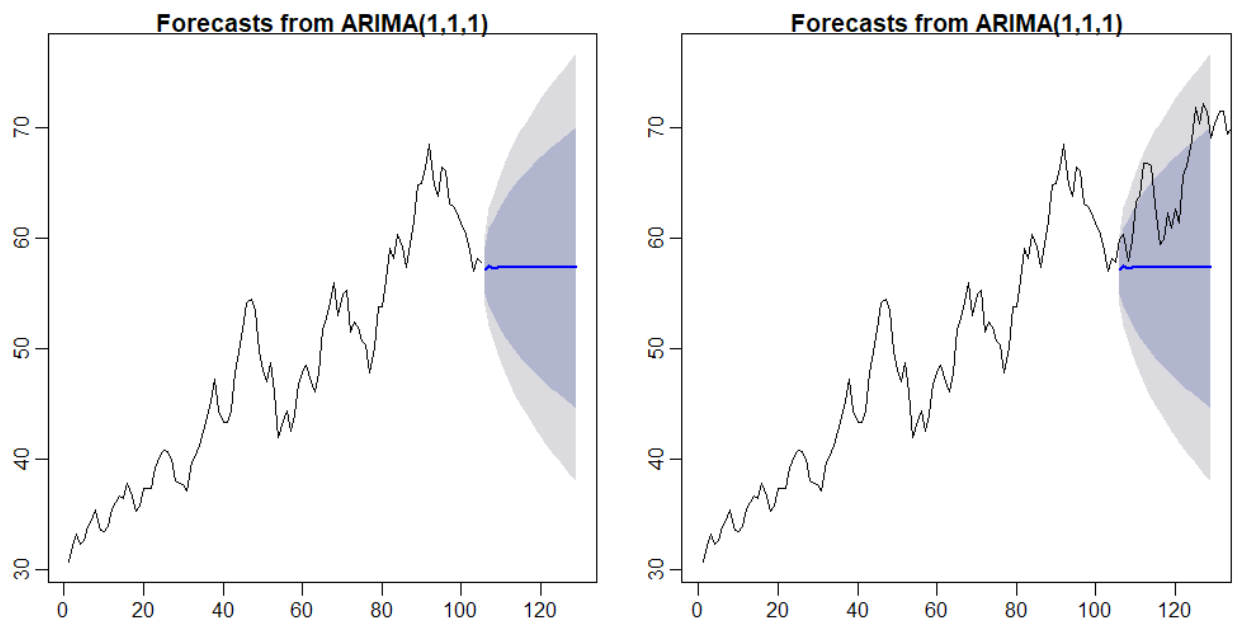
#Model the Data
oil_fit <- arima(oil_small, order=c(1,1,1))

#Forecast the data
oil_fc24 <- forecast(object = oil_fit, h = 24)

#Plot the Forecast
layout(matrix(1:2,ncol=2))
plot(oil_fc24)

#Fit the observed values
plot(oil_fc24)
lines(oil_future)
```

The following plots are hence generated:



With `astsa` package and `sarima` function

In the `astsa` package the `sarima.for()` function can be used to forecast data, a fitted model object is not first required for this method, this is best shown by way of example.

Using the `oil` data set which is built into the `astsa` package, observe that an ARIMA(1,1,1) model can describe the data. The following code will use that to predict the next 52 weeks/observations:

```
require(astsa)

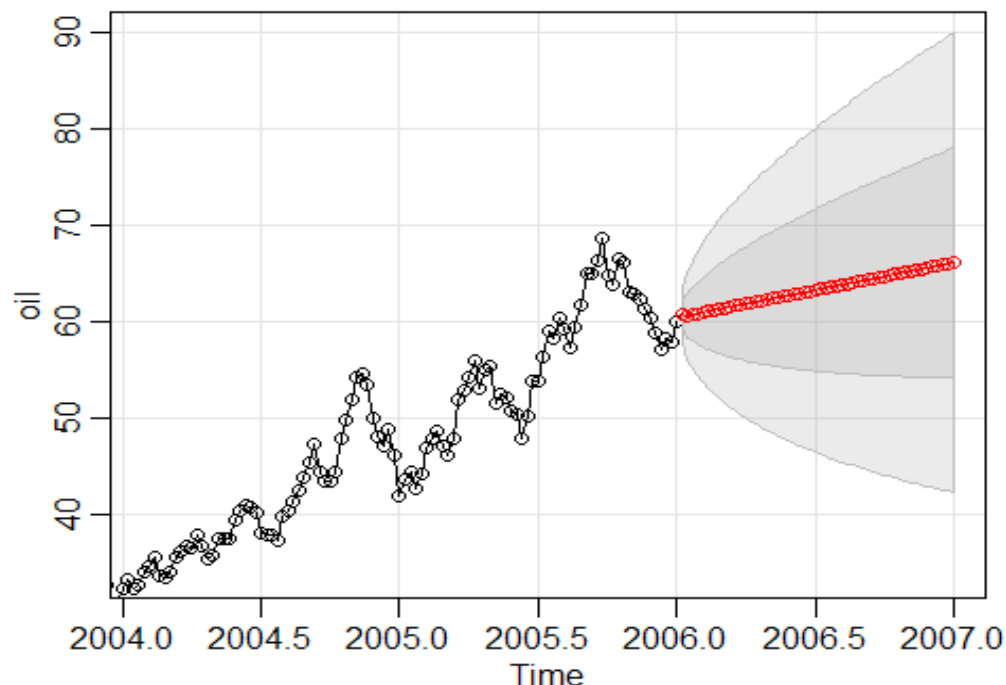
matrix(head(oil), ncol=1)

##           [,1]
## [1,] 26.20
## [2,] 26.07
## [3,] 26.34
## [4,] 24.95
## [5,] 26.27
## [6,] 29.37

#split up the data so the actual data can be plotted over the prediction
oil <- window(astsa::oil, end=2006)
oilf <- window(astsa::oil, end=2007)

#forecast the next 52 observations:
sarima.for(xdata = oil, n.ahead = 52, p = 1, d = 1, q = 1)
```

And automatically generates this output:



The forecast values are predicted in red, the dark grey area denotes ' ± 1 root mean square prediction error' and the light grey area denotes ' ± 2 root mean square prediction error', the light grey corresponds to a 95% prediction interval (I can only assume this correlates to standard error from the mean, so 2 is like 95%).

The following code will compare that forecast cone to the data that was actually observed over that period:

```
require(astsa)

## Loading required package: astsa

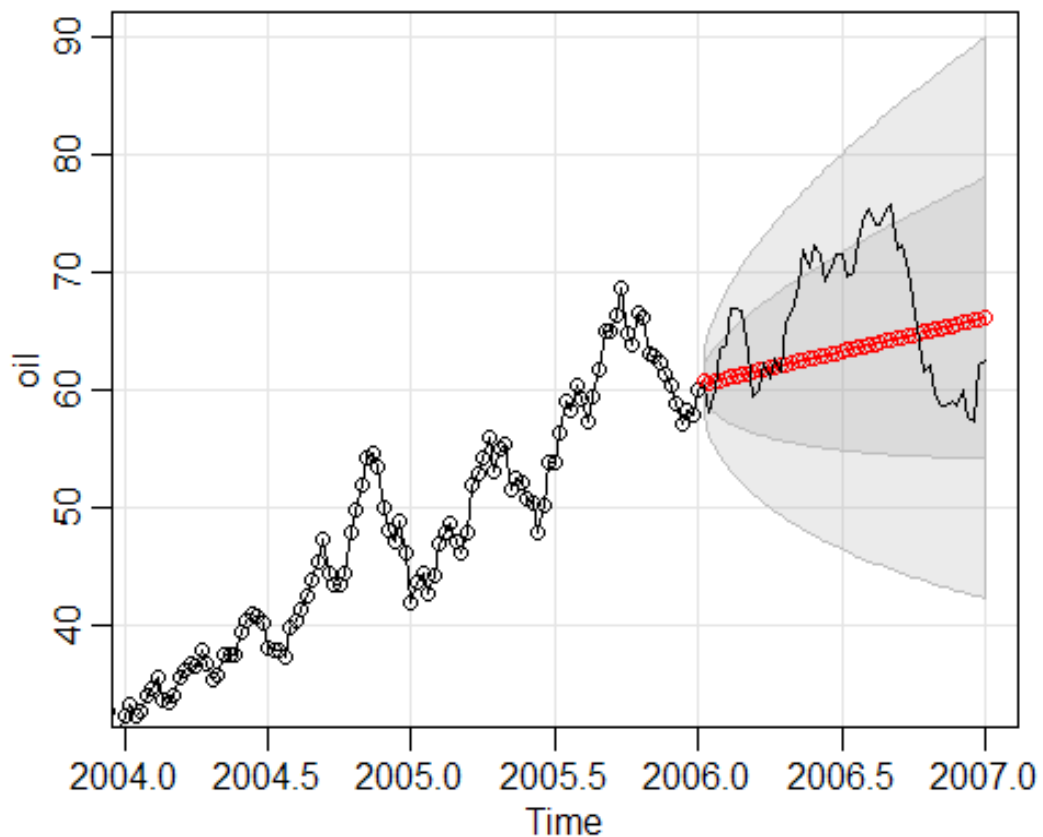
matrix(head(oil), ncol=1)

##      [,1]
## [1,] 26.20
## [2,] 26.07
## [3,] 26.34
## [4,] 24.95
## [5,] 26.27
## [6,] 29.37

#split up the data so the actual data can be plotted over the prediction
oil <- window(astsa::oil, end=2006)
oilf <- window(astsa::oil, end=2007)

#forecast the next 52 observations:
sarima.for(xdata = oil, n.ahead = 52, p = 1, d = 1, q = 1)

#Now plot in the missing data
lines(oilf)
```



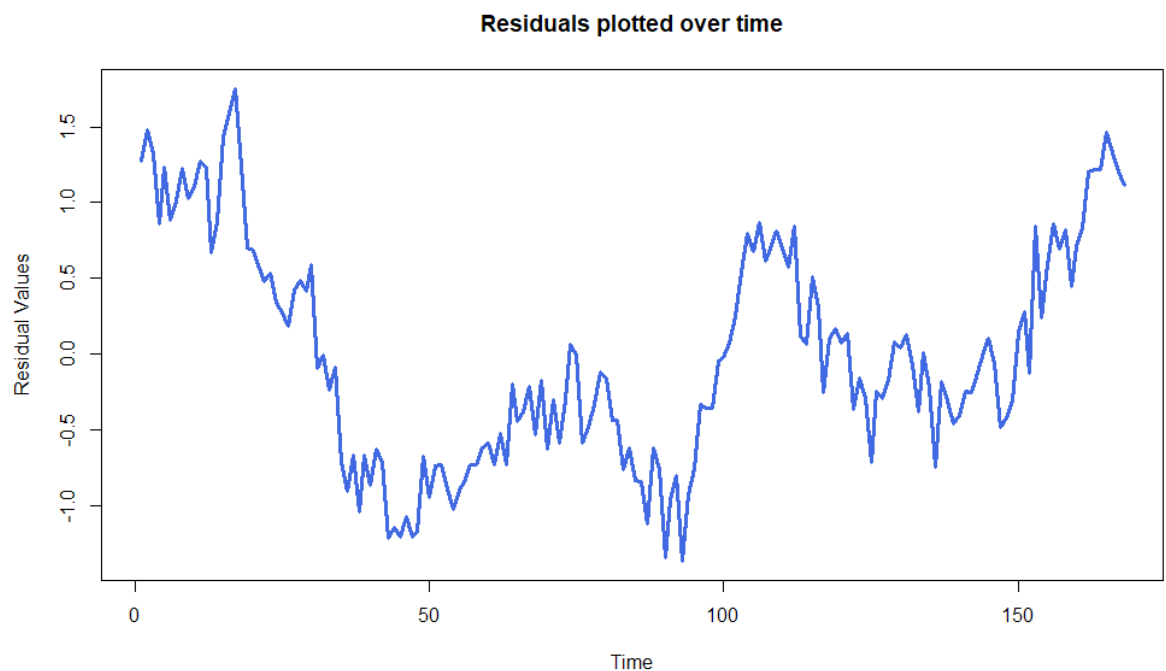
Exercise 9.1

Example 9.1

Fit an ARIMA model to the 'Mauna Loa' CO2 linear model residuals.

First create the linear model and plot the residuals:

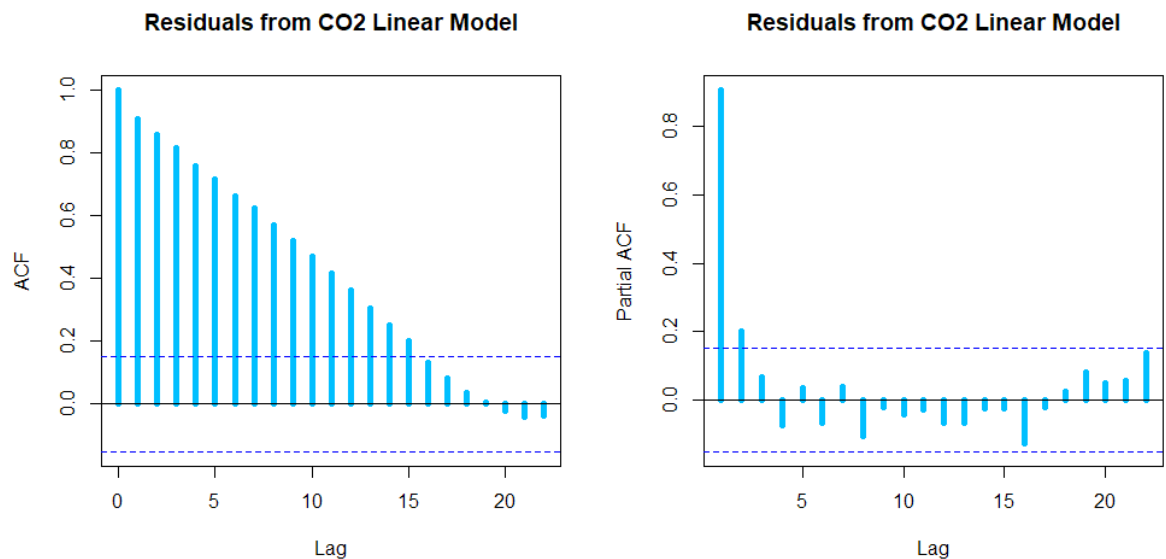
```
# Preamble -----  
  
# Import Data Set  
# this.dir <- dirname(parent.frame(2)$ofile)  
# setwd(this.dir)  
co2_df <- read.csv("MaunaLoaCO2.csv")  
  
# Create the Assignments  
co2.14 <- co2_df$co2.14  
month.14 <- co2_df$month.14  
time <- co2_df$time  
  
# Fit a Linear Model -----  
  
co2.lm <- lm(co2.14~time+factor(month.14))  
  
# Fit an ARIMA model -----  
  
# Plot the data  
plot.ts(co2.lm$residuals, col="royalblue", lwd=3)
```



This plot seems reasonably stationary, so the ACF/PACF plot will be inspected and an ARIMA model hence fit to the data.

Plot the ACF/PACF data:

```
#Plot the aCF/PACF
layout(matrix(1:2, ncol=2))
acf(co2.lm$residuals, col="deepskyblue", lwd=5,
    main="Residuals from CO2 Linear Model")
pacf(co2.lm$residuals, col="deepskyblue", lwd=5,
     main="Residuals from CO2 Linear Model")
```



As the ACF tails off to zero and the PACF cuts off at lag 2, an ARMA(2,0) model will be chosen, fitting this model in **R**:

```
#Choose ARIMA(2,0,0)

#Fit that model
z.fit <- arima(co2.lm$residuals,
               order=c(2,0,0),
               include.mean = FALSE)
require(lmtest)
coeftest(z.fit)

##      Estimate Std. Error z value Pr(>|z|)
## ar1 0.670368    0.073528  9.1172 < 2.2e-16 ***
## ar2 0.285077    0.074802  3.8111 0.0001384 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Alternative Method to fit model
```

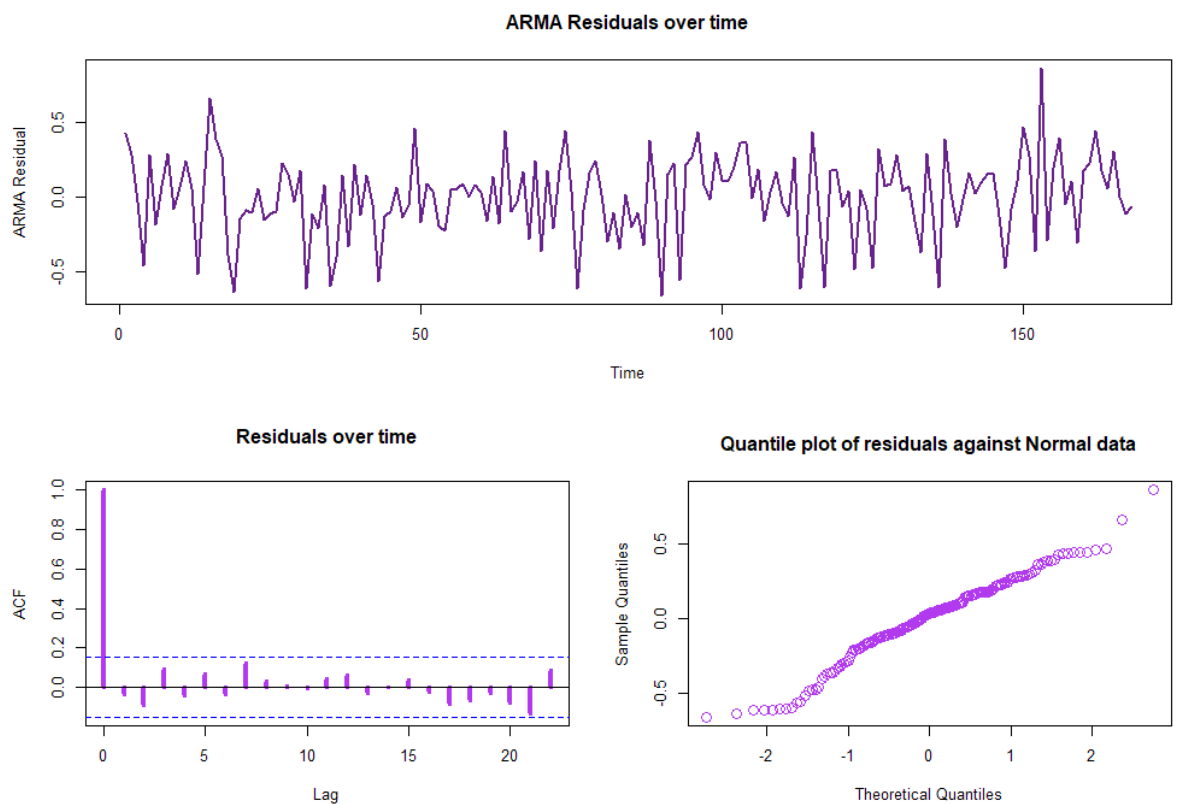
Thus the model for the residuals of the data is:

$$Y_t = 0.64 \cdot Y_{t-1} + 0.29 \cdot Y_{t-2}$$

Example 9.2

Given Example 9.1, perform residual diagnostics on the residuals of the ARMA(2,0) model:

```
# Plot the ARMA Residuals -----
layout(matrix(c(3,1,3,2), 2))
acf(z.fit$residuals, main="Residuals over time", lwd=4, col="darkorchid2")
qqnorm(z.fit$residuals,
       main="Quantile plot of residuals against Normal data",
       col="darkorchid2",
       cex=1.5)
plot.ts(z.fit$residuals, ylab="ARMA Residual",
        main="ARMA Residuals over time",
        lwd=2, col="darkorchid4")
```



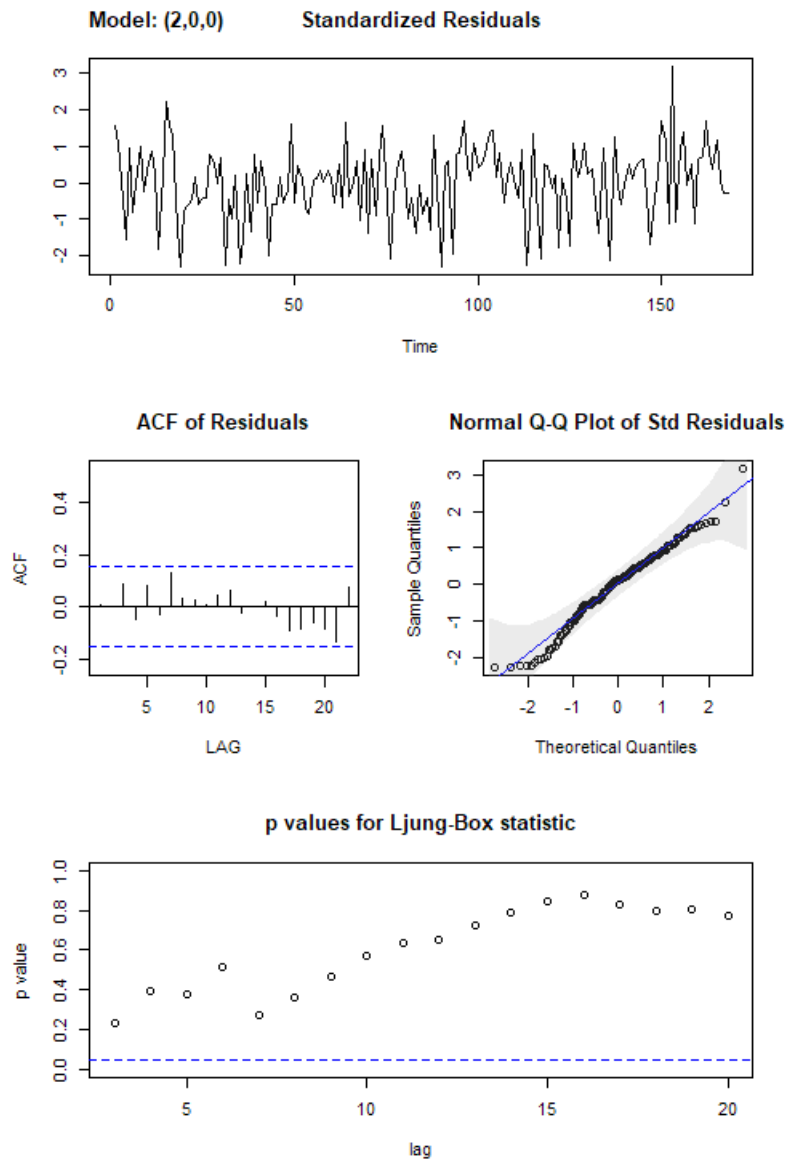
Thus the residuals provide that the model has rather normally distributed residual values, hence it is a good fit.

Alternative *sarima* method with *astsa* package

The following code will generate a model and residual diagnostics:

```
#Do it with astsa package
require(astsa)

sarima(z.fit$residuals, 2,0,0)
```



These residuals suggest the same thing, but the Ljung-Box statistic provides further suggests the normaliy of the residuals because the test is unable to prove, at a 5% significance level, that the data is non-normal.

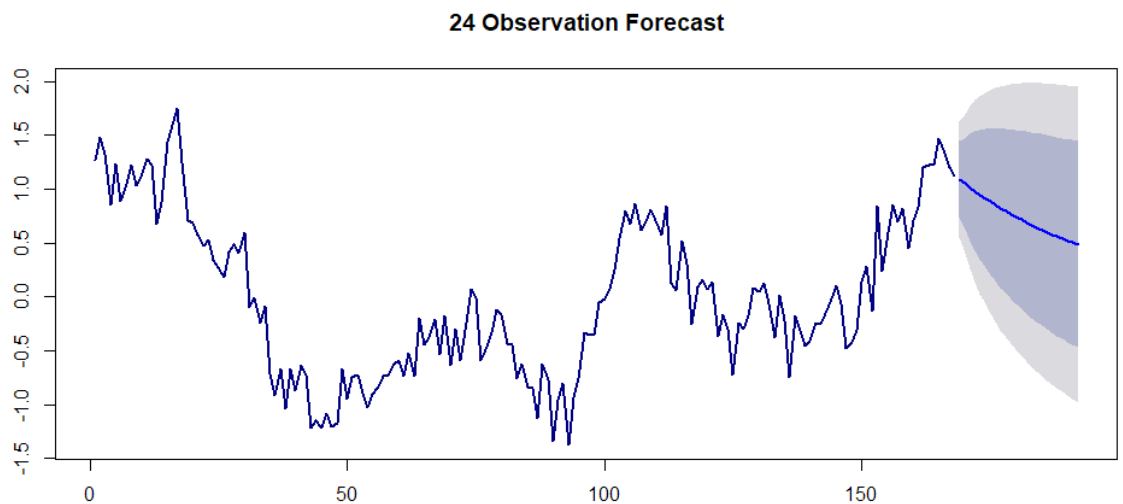
Example 9.3

Given the appropriate mode, perform a forecast 24 observations ahead.

Method using forecast package

The next 24 observations can be forecasted and plotted thusly:

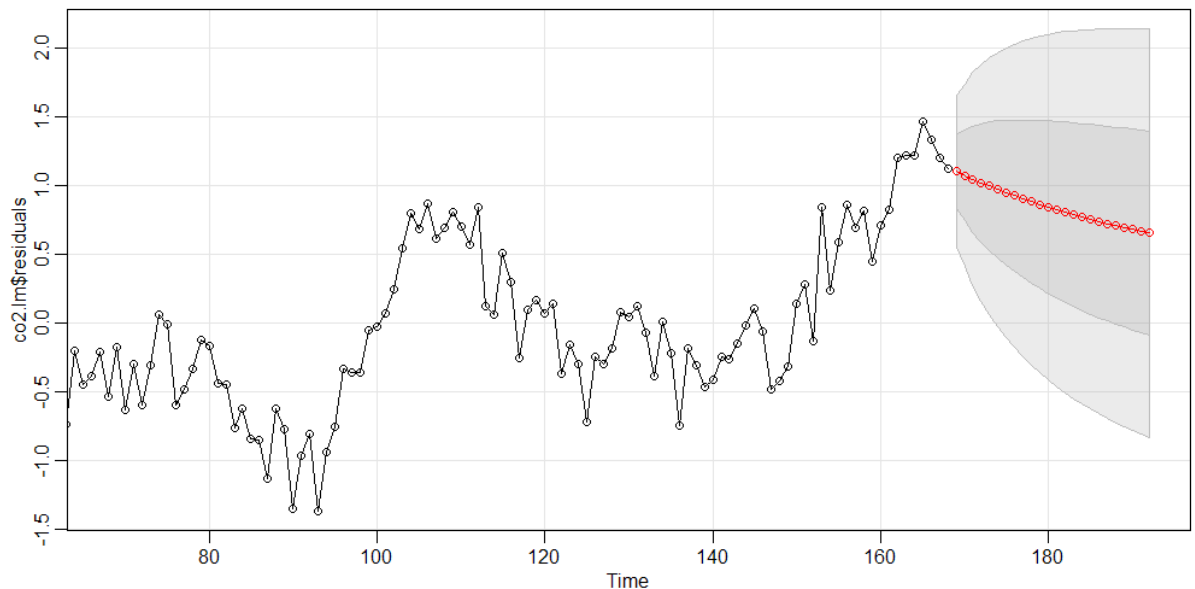
```
# Forecast 24 observations ahead -----  
layout(matrix(1))  
require(forecast)  
forecast(z.fit, 4)  
  
##      Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95  
## 169      1.091412  0.7397413  1.443083  0.5535781  1.629246  
## 170      1.050307  0.6269277  1.473686  0.4028045  1.697809  
## 171      1.015229  0.5192811  1.511177  0.2567421  1.773716  
## 172      0.979996  0.4288735  1.531118  0.1371268  1.822865  
  
plot(forecast(z.fit, 24), col="navy", lwd=2,
```



Method using *astsa* package

The next 24 observations can be predicted and plotted thusly:

```
#astsa package
sarima.for(co2.lm$residuals, 24, 2,0,0)
```



The forecast values are predicted in red, the dark grey area denotes ' ± 1 root **mean square prediction error**' and the light grey are denotes ' ± 2 root **mean square prediction error**', the light grey corresponds to a 95% prediction interval (I can only assume this correlates to standard error from the mean, so $z = 1.96$ is like 95%).

Plot the CO_2 Time Series

The CO_2 values follow a time series of the form:

$$CO_2 = 0.138 \cdot t + \hat{S}_t + \hat{Z}_t$$

- Z is the random error (which we now know follows an ARMA(2,0) process (does that mean it is actually random???)
- S is the seasonal fluctuation.

Create a plot of the expected CO_2 values using this model:

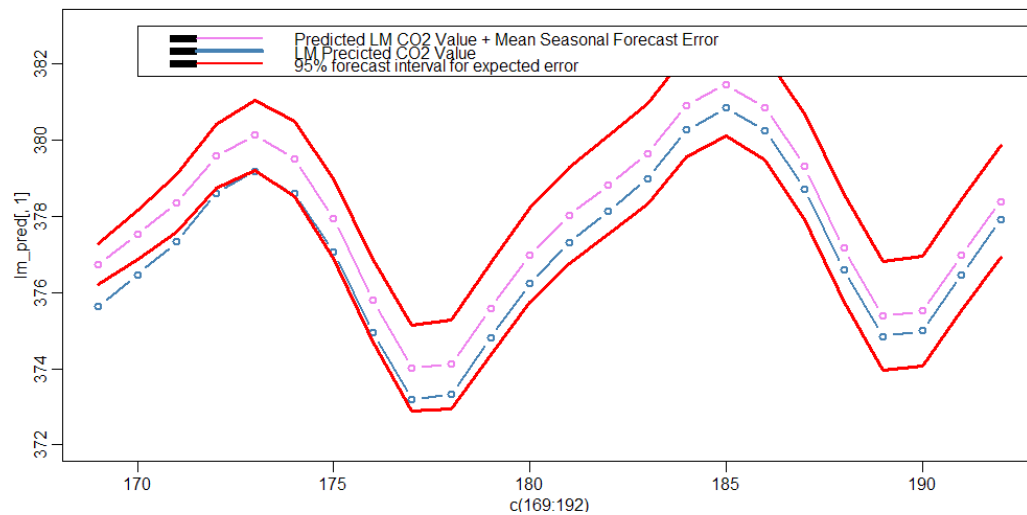
```
# Plot the expected values of CO2 -----

pred_df <- data.frame(time=c(169:192), month.14=factor(c(1:12, 1:12)))
lm_pred <- predict(co2.lm, pred_df, interval='confidence' )
error_fc <- forecast(z.fit,24)

m <- lm_pred[,1]+error_fc$mean
l <- lm_pred[,1]+error_fc$lower[,2]
u <- lm_pred[,1]+error_fc$upper[,2]

plot(c(169:192), lm_pred[,1], type = "b", ylim=c(372, 383),
     col='steelblue', lwd=2, xlab="time", ylab="CO2 Value")
lines(c(169:192), m, type="b", col="violet", lwd=2)
lines(c(169:192), l, col="red", lwd=3)
lines(c(169:192), u, col="red", lwd=3)

legend(170,383,
      legend = c("Predicted LM CO2 Value + Mean Seasonal Forecast Error",
                  "LM Predicted CO2 Value",
                  "95% forecast interval for expected error"),
      fill = TRUE,
      col=c("violet", "steelblue", "red"),
      lty=(1),
      lwd=c(2,4)
    )
)
```



Example 9.4 / Further Discussion

Observe that the prediction intervals for the model:

$$CO_2 = 0.138 \cdot t + \hat{S}_t$$

Are not particularly accurate, because it is missing the prediction for the random error:

```
# Compare the Forecasts of a Model with and Without the ARMA(2,0) -----

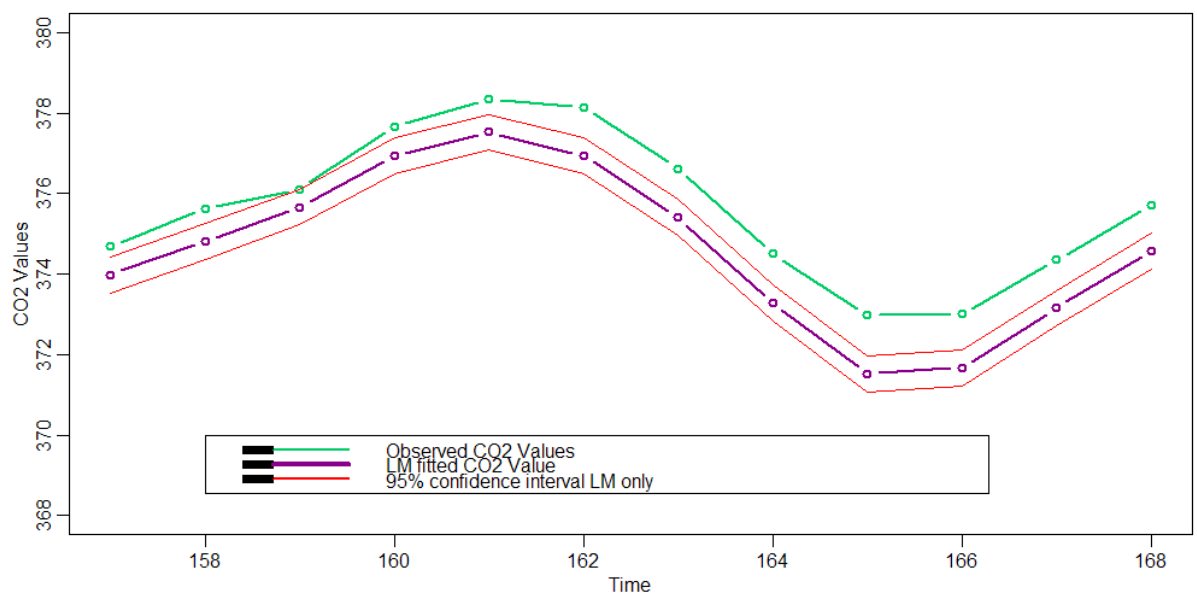
y <- co2_df[c(1:156),]
y.lm <- lm(co2.14~time+factor(month.14))
pred_df <- data.frame(time=c(157:168), month.14=factor(c(1:12)))
y_pred <- predict(y.lm, pred_df, interval='confidence')

plot_dst <- c(157:168)

plot(plot_dst, y_pred[,1], type = "b",
      ylim= c(368, 380), xlab="Time",
      ylab="CO2 Values", col="darkmagenta", lwd=2)
lines(plot_dst, tail(co2_df$co2.14,12), type = "b", col="springgreen3", lwd=2)

lines(plot_dst, y_pred[,3], col="red")
lines(plot_dst, y_pred[,2], col="red")

legend(158,370,
       legend = c("Observed CO2 Values",
                  "LM fitted CO2 Value",
                  "95% confidence interval LM only"),
       fill = TRUE,
       col=c("springgreen3", "darkmagenta", "red"),
       lty=(1),
       lwd=c(2,4))
)
```



While the prediction intervals for the model:

$$CO_2 = 0.138 \cdot t + \hat{S}_t + Z_t$$

Are within reasonable Specification:

```
#Inclusive of the ARIMA prediction
z <- y.lm$residuals
z.fit <- arima(z, order=c(2,0,0), include.mean = FALSE)
z.for <- forecast(z.fit, 12)

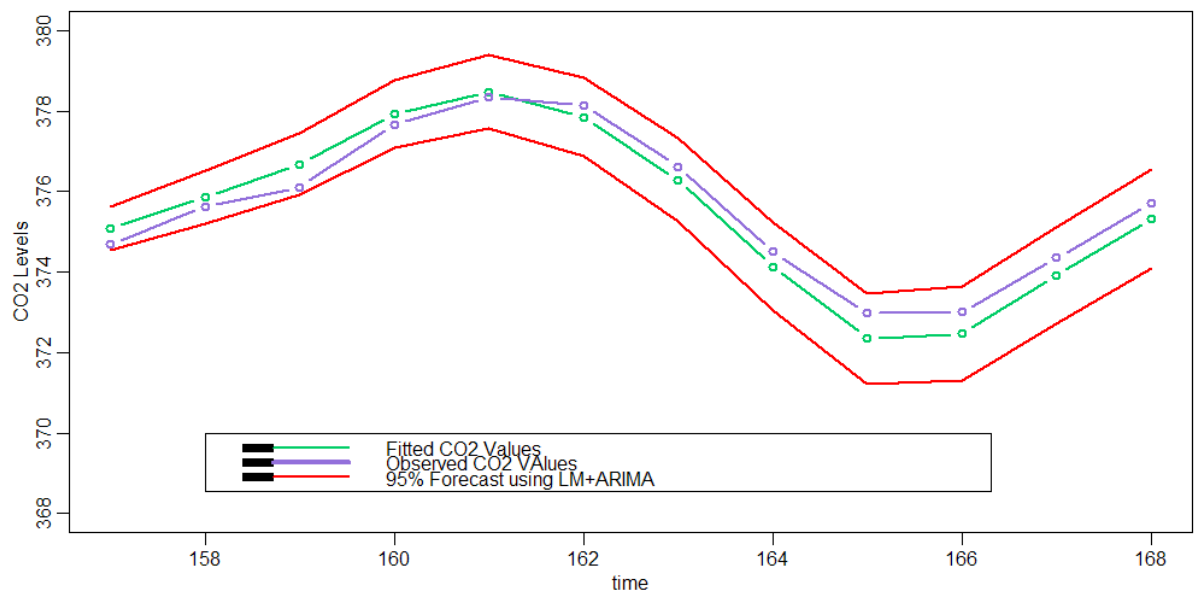
m <- y_pred[,1]+z.for$mean
l <- y_pred[,1]+z.for$lower[,2]
u <- y_pred[,1]+z.for$upper[,2]

plot_dist <- c(157:168)
obs <- tail(co2_df$co2.14,12)

#Plot the fitted/expected CO2 Values
plot(plot_dist, m, type = "b", ylim = c(368, 380),
      ylab="CO2 Levels", xlab="time", col="springgreen3", lwd=2)
#Add the Observed CO2 Levels over that interval
lines(plot_dist, obs, type = "b", col="mediumpurple", lwd=2)

#add the Confidence Intervals
lines(plot_dist, l, col="red", lwd=2)
lines(plot_dist, u, col="red", lwd=2)

#add the Legend
legend(158,370,
      legend = c("Fitted CO2 Values",
                  "Observed CO2 VAlues",
                  "95% Forecast using LM+ARIMA"),
      fill = TRUE,
      col=c("springgreen3", "mediumpurple", "red"),
      lty=(1),
      lwd=c(2,4)
    )
)
```



Spatial Data

Week 11 Material (25 Sep), Due Wk. 12 (2 Oct 2017), Lecture 10

Types of Spatial Data

Spatial Data has the following elements:

- Co-ordinates
 - Spatial Data
- Co-ordinate Reference System
 - How to interpret those co-ordinates
- Attribute
 - Associated data
 - e.g. , time of day, length of fish etc.

There are different types of spatial data:

- Points data
 - A location of points associated with other data (e.g. house location and price).
- Line data
 - Associated with a line
- Polygon data
 - Associated with the enclosed area of a shape
 - E.g. polygons describing a field and a crop
- Raster data
 - Grid data, data is correlated with each cell in a grid (think satellites)
- Lattice Data
 - Observations associated with an area or region (this could be the same as raster data).

Points Data

Geostatistical data

Is data that has a location associated with it and one or more variables measured at each location, usually the variable measured is of interest.

e.g. pollution at different places

Spatial Point Pattern Data

Is data composed of point locations associated with them and the actual location is the variable of interest.

e.g. location of trees in a forest.

Definitions

Specific definitions are important:

- **Point**
 - Is any point on the x, y plane. (e.g. GPS reading)
- **Line**
 - A set of ordered points, connected by straight line segments
- **Polygon**
 - An area marked by one or more enclosing lines (it could contain holes)
- **Grid**
 - A collection of rectangular cells, organised in a regular lattice.
- **Event**
 - An event is a recorded/observed data point.
- **Window**
 - The study area containing the observations
 - Events happening outside the window are unobserved
- **Spatial Point Pattern**
 - Is the set of observed events and the window
- **A Spatial Point Process**
 - Is a stochastic process
 - A random number generator for points within a window essentially
- **Marks**
 - Other variables pertaining to an event/observation (e.g. tree height or species)

Models for Geostatistical Data

In geostatistical data s is used to denote a location, where $s = (x, y)$ represents Cartesian coordinates of a location. An observation at location s is denoted:

$$z(s) = z(x, y)$$

A general form for a model of geostatistical data is some function + noise:

$$Z(s) = f(s) + \varepsilon(s)$$

The noise is assumed to have a mean of 0 and a constant standard deviation.

Parametric

Linear Model

The simplest parametric model is a linear one:

$$z = \beta_0 + \beta_1 \cdot x + \beta_2 \cdot y$$

Polynomial

The polynomial function is also often used, e.g. a quadratic model:

$$z = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot y + \beta_3 \cdot x^2 + \beta_4 \cdot y^2 + \beta_5 \cdot xy$$

Generally any kind of polynomial surface can be used to describe changes in the response variable Z as a function of location and other predictor variables can be added as well.

Non-parametric

Loess

The LOESS or LOWESS (locally weighted scatterplot smoothing) are non-parametric regression methods that involve fitting regression models to weighted points between observations.

Fitting models for Geostatistical Data

Given geostatistical data, in order to make 3d plots and predictions a model will be required to make a smooth surface through the points, two methods covered here are:

- Loess smooth
 - Short for locally weighted scatterplot smoothing; It inessentially creates local averaged points
 - A loess smooth model can be made in **R** using the following function:
 - `loess(Y ~ x1 * x2, normalize=FALSE, span)`
- Polynomial fitting
 - A multiple polynomial function can be used to model through the points
 - A polynomial model can be made in **R** using the following function
 - `lm(y ~ poly(x1, x2, degree = n))`
 - When trying to create a matrix for the z values, it is important to use the gam package and use the `predict.gam()` function

Visualising Spatial Data

Latitude and Longitude

- Latitude is a line that runs east to west that measures how far north of the equator
 - Latitude is assigned to the y-axis
- Longitude is a line that runs north to south that measures east/west
 - Longitude is assigned to the x-axis

Methods for Visualisation

Refer to Exercise 10.1 (example 10.1) for exemplars and methodology.

1. Scatter plot
 - a. Ideally a scatter plot over a map using the `ggmap()` package and `bbox()`, `get_map()` and `ggmap()` functions
2. Scatterplot matrix
 - a. This can be made using `pairs` as the base package
 - b. Or `ggpairs` as the preferred method.
3. Contour plot
 - a. the `contour()` function will create a base package plot
 - i. First however, it is necessary to create a model and create predictions for that model in order to plot the surfaces/contours
 - b. `ggplot2` can create a contour map with the `geom_contour()` layer
 - i. First however it will be necessary to go from the matrix created by `predict()` into a tidy data frame.
 1. This will require the `melt` function, and
 2. The `str_sub()` + `str_locate()` functions
4. 3d Surface plot
 - a. This requires x and y vectors that are equally spaced sequences of values as well as a matrix of z values, it is a little confusing, refer to the exemplar.
 - b. Fortunately enough, creating a surface plot in `plotly` is no more difficult than the base package `persp()`.

Exercise 10.1

Discussion of Benthic Models

It appears that the benthic index is the healthiest in the bay at the river.

It is rather high on the banks, and is somewhat higher near roadways.

Methodology:

Use spatial data visualising techniques to interpret the following data:

```
> library(EnvStats); head(Benthic.df)
  Site.ID Stratum Latitude Longitude Index Salinity Silt
1 89/90-1     101  38.4430   -76.4457  2.33      2.0  0.6
2 89/90-1     101  38.4232   -76.4237  3.67      2.0  0.8
3 89/90-1     101  38.4773   -76.4827  2.00      1.5  0.6
4 89/90-2     101  38.4740   -76.4792  2.67      1.6  0.8
5 89/90-2     101  38.4598   -76.4542  2.33      2.0  0.7
6 89/90-2     101  38.4718   -76.4768  2.00      1.2  0.8
...
...
```

(1) Scatter plot

R Script

```
# Preamble -----
# Packages
library(EnvStats)

##
## Attaching package: 'EnvStats'
## The following objects are masked from 'package:stats':
##
##   predict, predict.lm
## The following object is masked from 'package:base':
##
##   print.default

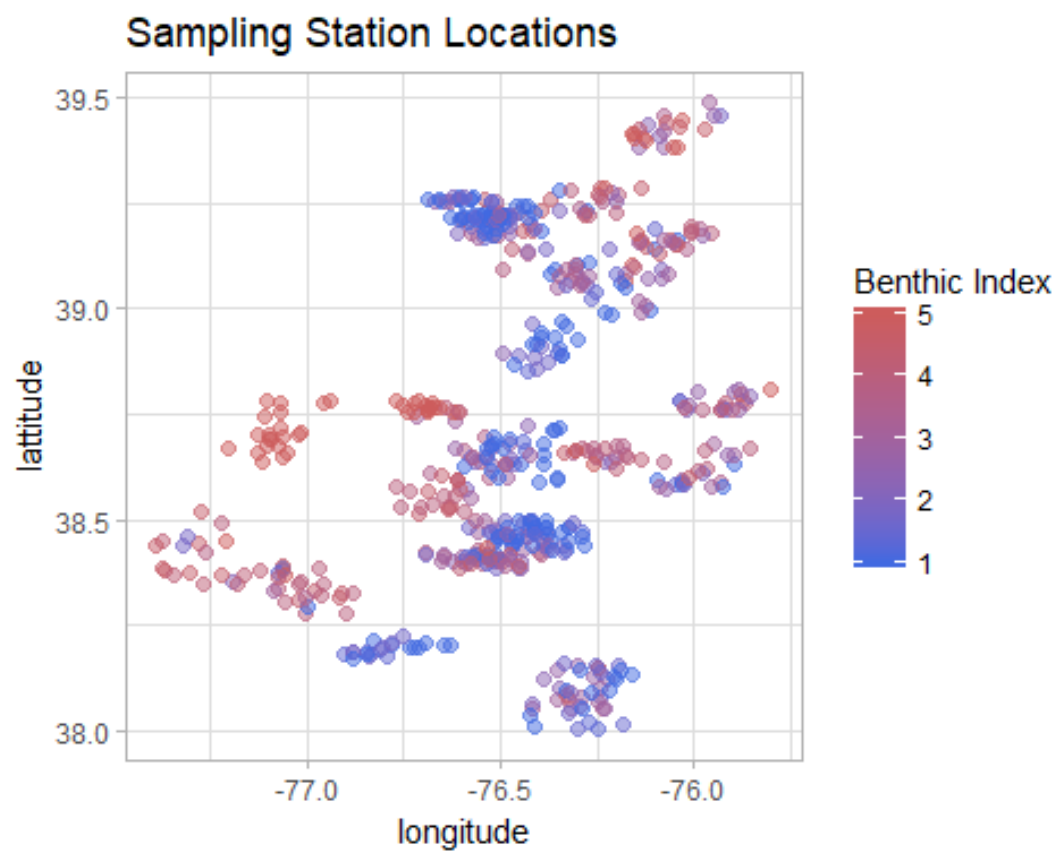
library(ggplot2)
# Assignments
benthic <- Benthic.df
lat <- benthic$Latitude
lon <- benthic$Longitude
index <- benthic$Index
sal <- benthic$Salinity
silt <- benthic$Silt

# Scatter Plot of Benthic -----

# Base Plot
plot(x = lon, y = lat, xlab = "Longitude", ylab = "Latitude",
     main = "Sampling Station Locations")

# Published plot
ggplot(data = benthic, aes(x = lon, y = lat, col = index)) +
  geom_jitter(alpha = 0.5, width = 0.1, size = 2) +
  labs(x = "longitude", y = "latitude", title = "Sampling Station Locations",
       col = "Benthic Index") +
```

Plot

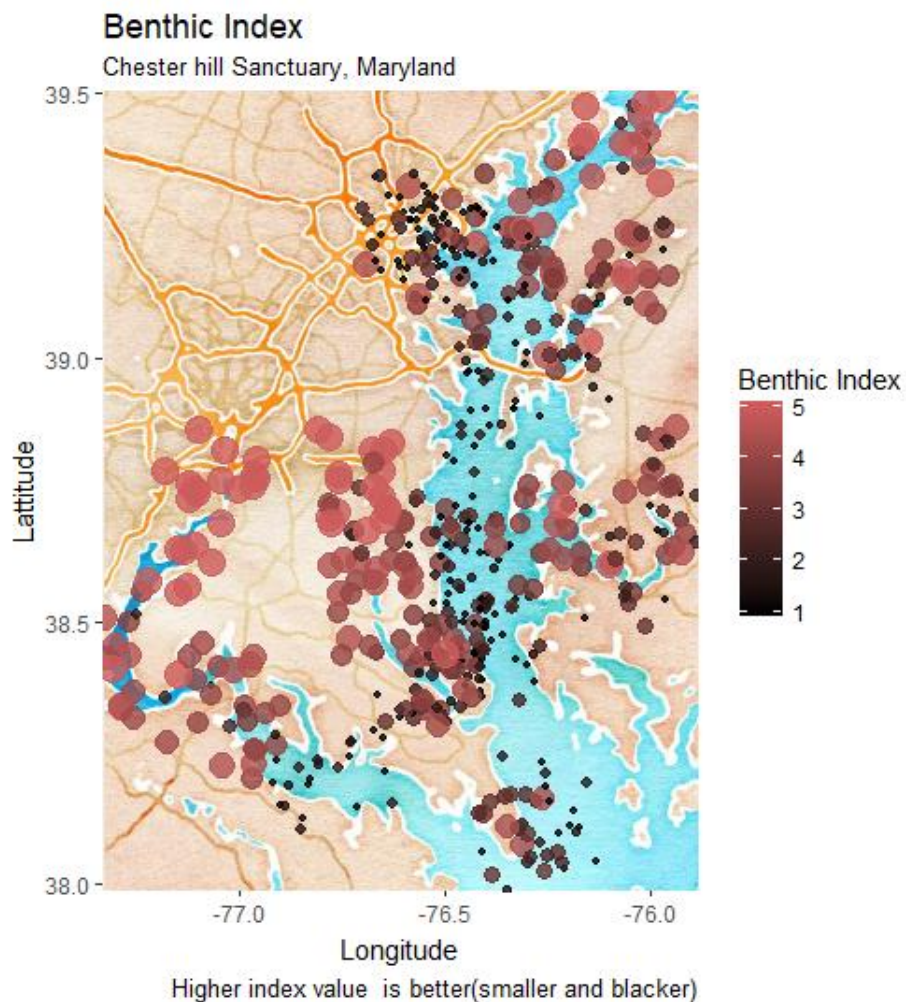


Include a Map background

Obviously, the values are more meaningful with a map background to inform the observations:

```
#Include the map
bbox <- make_bbox(lon, lat, f = 0.01)
map <- get_map(location = bbox, maptype = 'watercolor')

ggmap(base_layer = ggplot(data = benthic,
                          aes(x = lon, y = lat, col = index)),
      ggmap = map) +
  geom_jitter(width = 0.1, height = 0.1, size = index, alpha = 0.8) +
  scale_color_continuous(low = 'black', high = 'indianred') +
  labs(x = 'Longitude', y = 'Latitude',
       title = "Benthic Index",
       subtitle = "Chester hill Sanctuary, Maryland",
       caption = 'Higher index value is better(smaller and blacker)',
       col = 'Benthic Index', size = 'Benthic Index')
```



(2) Plot Matrix

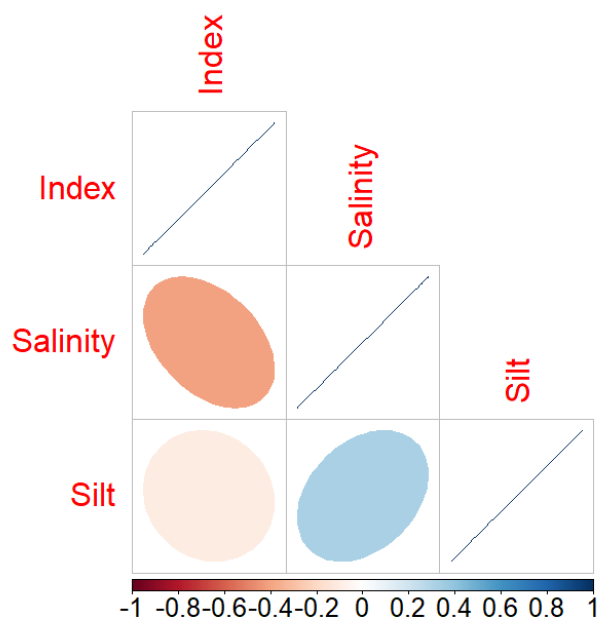
Correlation Matrix

A possible method to view the data could be a correlation matrix:

R Script

```
# Scatterplot Matrix -----  
#Correlation Matrix  
cormat <- cor(benthic[,-(1:4)], use = 'complete.obs', method = 'pearson')  
corrplot(method = 'ellipse', type = 'lower', corr = cormat )
```

Plot Matrix



Scatterplot Matrix

A scatterplot matrix can be created:

*R*Script

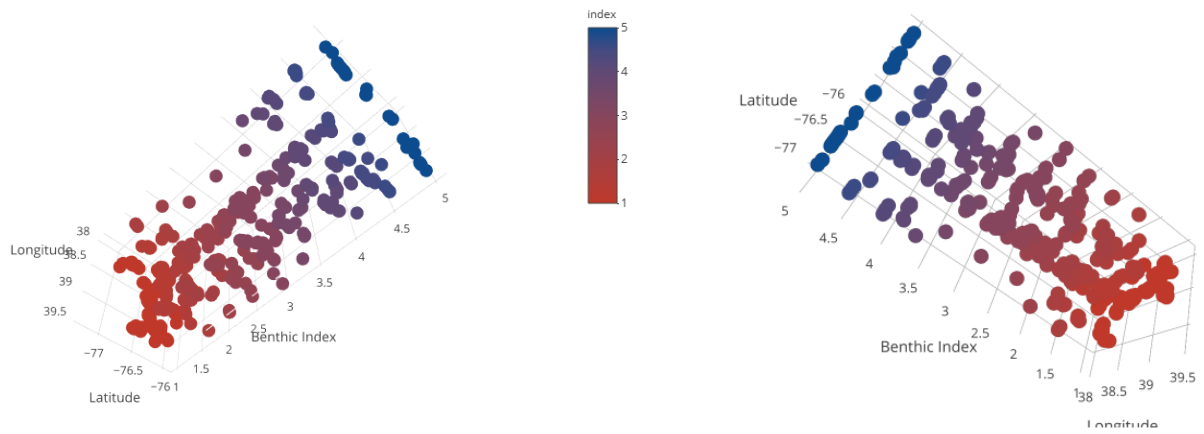


Plot

3d Scatter Plot

A 3d plot may enable a better understanding of the benthic index, this can be achieved in `plotly` relatively painlessly:

```
p <- plot_ly(z = ~index.fit) %>%  
  add_surface() %>%  
  layout(scene = list(xaxis = list(title = 'Longitude'),  
                        yaxis = list(title = 'Latitude'),  
                        zaxis = list(title = 'Benthic Index')))
```



Contour Plot

In order for a contour or surface plot, it will be necessary to use a continuous smoothed model.

```
# Contour Plot -----
#Before any 3d surface or contour plots can be made
#A smooth modelled surface needs to be predicted

#Create model for smoothed surface
alpha      <- 0.1
index.model <- loess(formula = index ~ lon * lat, span = alpha)

# Create a sequence of incrementally increasing for the X and Y axis
xgrid      <- seq(min(lon), max(lon), length.out = 100)
ygrid      <- seq(min(lat), max(lat), length.out = 100)

# Generate a dataframe with every possible combination of wt and hp
xy.surface <- expand.grid(lon = xgrid, lat = ygrid)

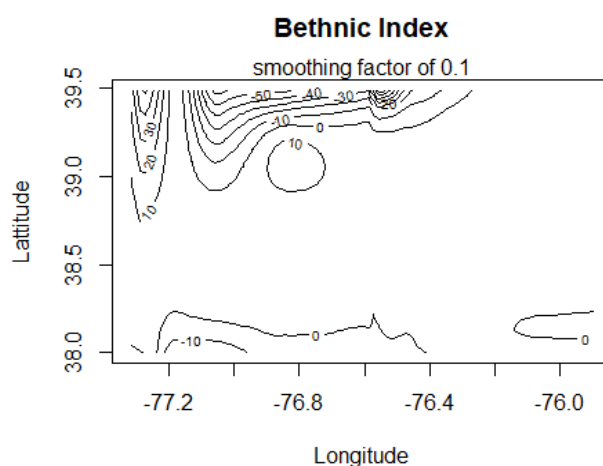
# Feed the dataframe into the loess model and receive a matrix output
#with estimates of the Z-value (Bethnic Index)

index.fit  <- predict(index.model, newdata = xy.surface)

# Abbreviated display of final matrix
index.fit[1:4, 1:4]
```

```
##           lat
## lon      lat=38.00180 lat=38.01682 lat=38.03183 lat=38.04685
## lon=-77.3157  11.072381  10.645442  10.233671  9.836950
## lon=-77.3014  10.630489  10.230144  9.844861  9.474586
## lon=-77.2871  10.158344  9.787083  9.430596  9.088908
## lon=-77.2728  9.774043  9.420654  9.080818  8.754604
```

```
#Create a Base graphic of the contours
contour(x = xgrid, y = ygrid, z = index.fit,
        xlab = "Longitude",
        ylab = "Latitude",
        main = "Bethnic Index")
mtext(paste("smoothing factor of", alpha))
```



Map plot

This plot would be more informative with a reference map:

```
#Create a ggplot2 Contour Plot
#Create a 'tidy' data frame (1 variable per column)
index.fit.melt <- melt(index.fit, varnames = c('Longitude', 'Latitude'),
                      value.name = "Benthic_Index")

head(index.fit.melt)

##      Longitude      Latitude Benthic_Index
## 1 lon=-77.3157 lat=38.00180      11.072381
## 2 lon=-77.3014 lat=38.00180      10.630489
## 3 lon=-77.2871 lat=38.00180      10.158344

#Use the regex to remove character 'lon=' / 'lat=' rubbish
index.fit.melt$Longitude <- as.numeric(
  str_sub(
    index.fit.melt$Longitude,
    str_locate(index.fit.melt$Longitude,
               '=')[1,1] + 1))

index.fit.melt$Latitude <- as.numeric(
  str_sub(
    index.fit.melt$Latitude,
    str_locate(index.fit.melt$Latitude,
               '=')[1,1] + 1))

#The window used was found to provide the most sensible contour lines
bbox <- c("left"=-77, "bottom" = 38.2, "right" = max(lon), "top" = 38.9)
map <- get_map(location = bbox, maptype = 'watercolor')

ggmap(ggmap = map, base_layer = ggplot(
  data = index.fit.melt,
  aes(x = Longitude, y = Latitude))) +

geom_contour(binwidth = 0.25,
  data = index.fit.melt,
  aes(x = Longitude,
    y = Latitude,
    z = Benthic_Index,
    colour = ..level..),
  size = .5) +

scale_color_continuous(low = 'black', high = 'indianred') +
labs(x = 'Longitude', y = 'Latitude',
  title = "Benthic Index",
```

Surface Plot

A 3d surface plot of the index values may help understand any patterns in the data .

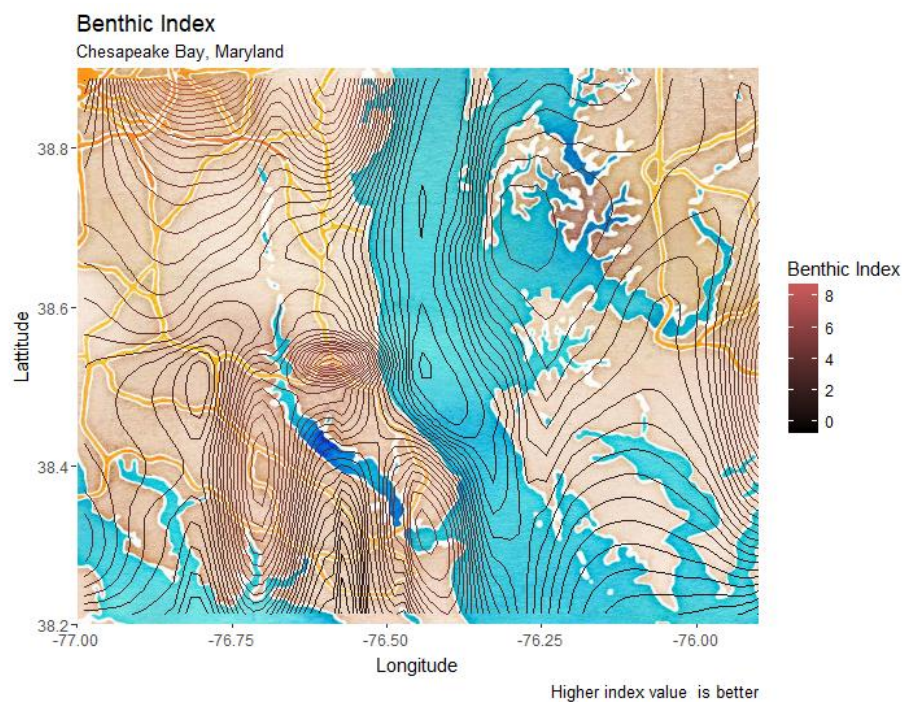
Creating Surface Plots

Creating surface plots in r is a little finnickly, three elements need to be passed to a function in order to create a surface plot:

- X vector containing equally spaced values that represent the x-values of the plot
- Y vector containing equally spaced values that represent the y-values of the plot
- Z Matrix, a matrix of values that corresponds to a table with the x and y vectors.

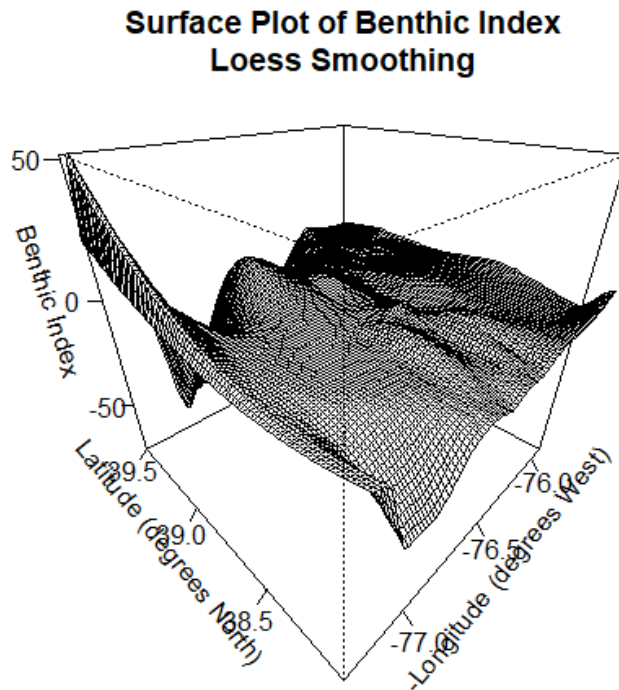
So The layout of data would look something like this:



| | | | | | | | | | | |
|--|----------------------------------|----|----|----|----|----|---|---|----|---|
| Y A x i s c (1 , 5 , 6 , 4 , 2) | X-Vector: c(1,3,5,3,2,2,7,5,9,5) | | | | | | | | | |
| | Z-values | 1 | 2 | 5 | 54 | 6 | 5 | 4 | 2 | 2 |
| | 1 | 54 | 4 | 5 | 1 | 2 | 1 | 5 | 74 | 5 |
| | 4 | 8 | 4 | 12 | 7 | 45 | 2 | 1 | 5 | 4 |
| | 2 | 1 | 22 | 15 | 4 | 5 | 4 | 5 | | 4 |
| | 21 | 4 | 54 | 4 | 45 | 5 | 5 | 4 | 7 | 8 |



Base **R** Surface Plot

```
#Base plot (from predict)
persp(xgrid, ygrid, index.fit,
      xlim = c(-77.3, -75.9),
      ylim = c(38.1, 39.5),
      # zlim = c(0, 6),
      theta = -45, phi = 30, d = 0.5,
      xlab="-Longitude (degrees West)",
      ylab="Latitude (degrees North)",
      zlab="Benthic Index", ticktype = "detailed")
```



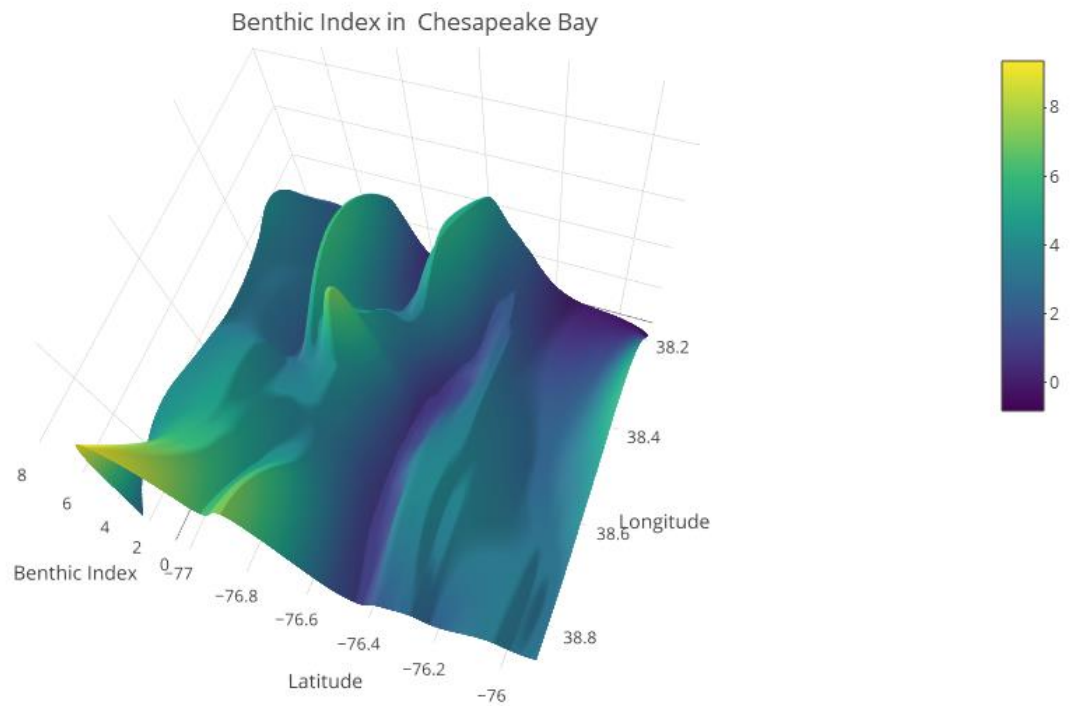
 a particularly attractive plot, and  **plotly** would offer a dynamic HTML plot

plotly surface plot

```
#Plotly plot

p <- plot_ly(x = xgrid, y = ygrid, z = index.fit) %>%
  add_surface() %>%
  layout(scene = list(xaxis = list(range = c(-77.3, -76), title =
'Longitude'),
                        yaxis = list(range = c(38.2, 38.9))),
            zaxis = list(range = c(0, 6))
          )

p
```



Discussion of Benthic Models

It appears that the benthic index is the healthiest in the bay at the river.

It is rather high on the banks, and is somewhat higher near roadways.

Spatial Correlation

Week 12 Material (2 Oct), Due Wk. 13 (9 Oct)

In spatial data, observations close to each other will often be similar to one another, i.e. they will be correlated.

This also implies that models of trend surface may have correlated errors.

In **geostatistics** spatial correlation is modelled by the **variogram** (for the scope of this work, this will be considered equivalent to a **semivariogram**).

Spatial Correlation

Stationarity

To estimate spatial correlation, it is necessary to assume that the data is stationary first.

Weak Stationarity (i.e. intrinsic stationarity)

Spatial Observations are weakly stationary if:

- The surface has a constant mean
- The covariance between two observations depends only on the distance.
 - And perhaps the direction
- Spatial observations have **isotropic** covariance if:
- The covariance only depends on the distance between locations
 - Not the direction between locations

Covariance and Covariogram

The function C is the **covariogram** function (which is like the auto-covariance function for time series data) and is defined:

$$\text{Cov}[Z(s_1), Z(s_2)] = C(h)$$

Where:

- s_1 and s_2 are locations
- h is the distance between those locations
- C is the **covariogram** function (which is like the auto-covariance function for time series data)

Correlogram

The **correlogram** function is analogous to the auto-correlation function (acf) in time series and is defined:

$$P(h) = \frac{C(h)}{C(0)}$$

Variogram

When dealing with spatial data, the **correlogram** and **covariogram** are not used, usually the **Variogram** is used instead.

The **variogram** is defined as:

$$\begin{aligned}\gamma(h) &= \frac{1}{2} \text{Var}[Z(s_1) - Z(s_2)] \\ &= \text{Var}[Z(s_1)] - \text{Cov}[Z(s_1), Z(s_2)] \\ &= C(0) - C(h)\end{aligned}$$

However the **variogram** is estimated by:

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \cdot \sum_{N(h)} [z(s_i) - z(s_j)]^2$$

Where:

- $N(h)$ is the set of all pairs that have a distance of h between them
 - $|N(h)|$ is the number of elements in that set, i.e. the number of pairs that are h units apart.
- Although the formula for $\hat{\gamma}(h)$ is a somewhat ambiguous function, I couldn't find anything clearer online.

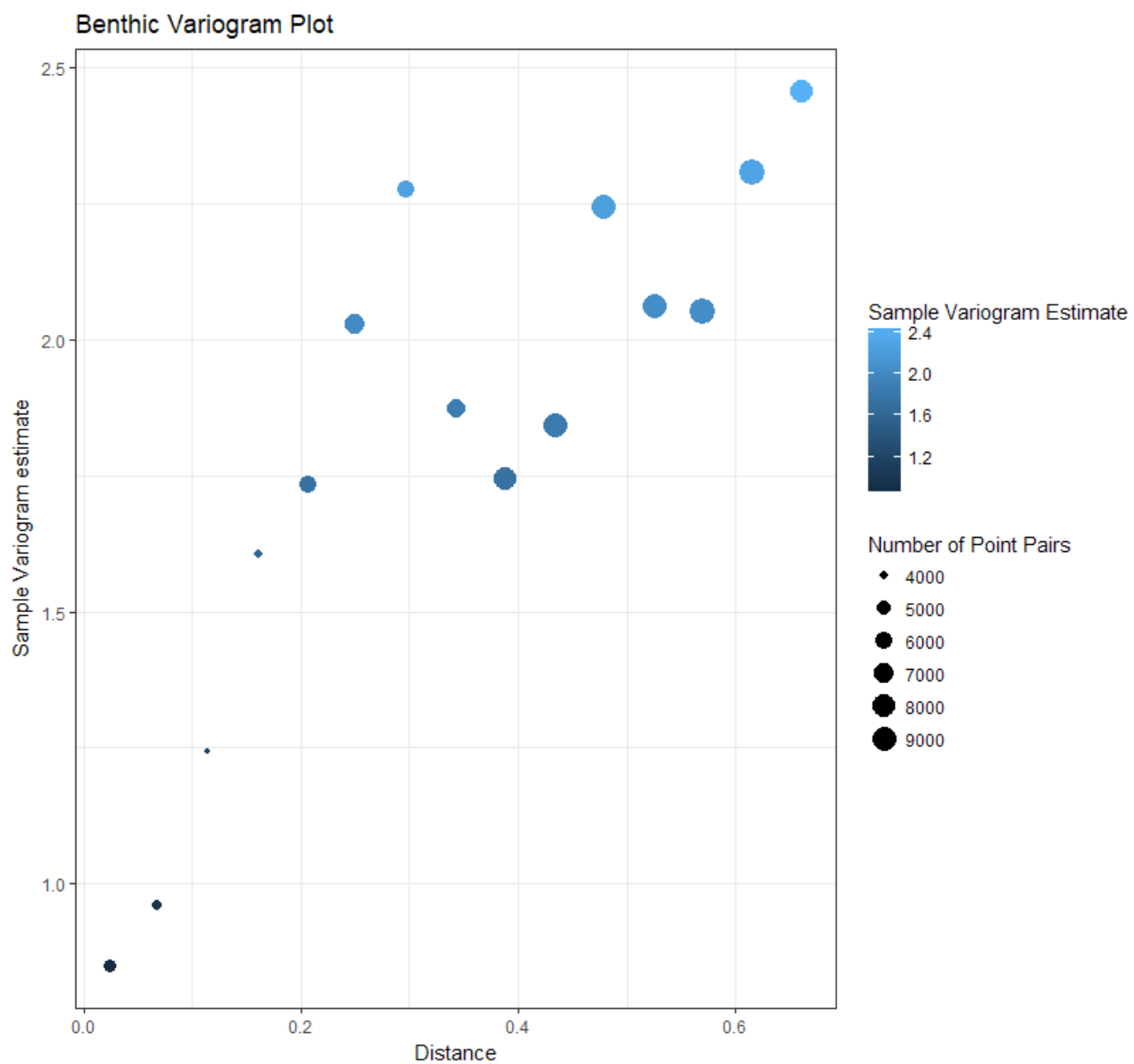
There are three features in variograms:⁶¹

- The **Sill**
 - This is the covariance ($C(h)$) at $h = 0$, i.e. $C(0)$
 - The sill is equal to the variance of the process
- The **Range**
 - The range is the distance at which observations are no longer correlated
- The **Nugget**
 - At a distance of 0, $h = 0$, the variogram should have a value of 0, however at really small variogram values there may be a larger value than 0, attributable to random error, this phenomena is known as the nugget effect.
 - If the y-intercept of a variogram is for example 3.3, the nugget is hence 3.3

⁶¹ Pro.arcgis.com. (2017). *Understanding a semivariogram: The range, sill, and nugget—ArcGIS Pro | ArcGIS Desktop*. [online] Available at: <http://pro.arcgis.com/en/pro-app/help/analysis/geostatistical-analyst/understanding-a-semivariogram-the-range-sill-and-nugget.htm#GUID-25865547-BD6A-4432-B8C8-B35F9407328B> [Accessed 10 Oct. 2017].

Variogram plot

```
ggplot(data = benthic.vg, aes(x = dist, y = gamma, col = gamma, size = np))  
+  
  geom_point() +  
  labs(size = "Number of Point Pairs", col = "Sample Variogram Estimate",  
        y = "Sample Variogram estimate", x = "Distance",  
        title = 'Benthic Variogram Plot') +  
  theme_bw()
```



Variogram Models

There are three common models for variogram data:

➤ Exponential

$$\circ C(h) = \sigma^2 \cdot e^{-\frac{h}{r}}$$

➤ Gaussian

$$\circ C(h) = \sigma^2 \cdot e^{\left(-\frac{h}{r}\right)^2}$$

➤ Spherical

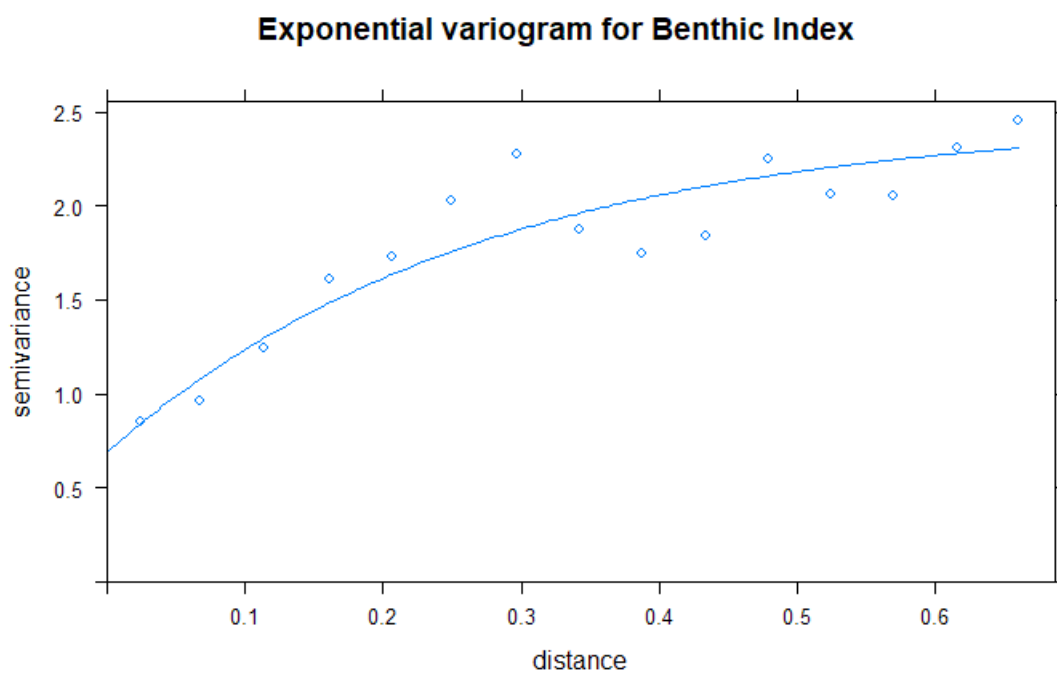
$$\circ C(h) = \sigma^2 \cdot \left(1 - \frac{3h}{2r} + \frac{h^2}{2r^3}\right), h < r$$

Where:

- σ^2 is the variance of the process
- r is the variance of the process
- The range for the exponential and Gaussian model is infinite, the range for the spherical model is r .

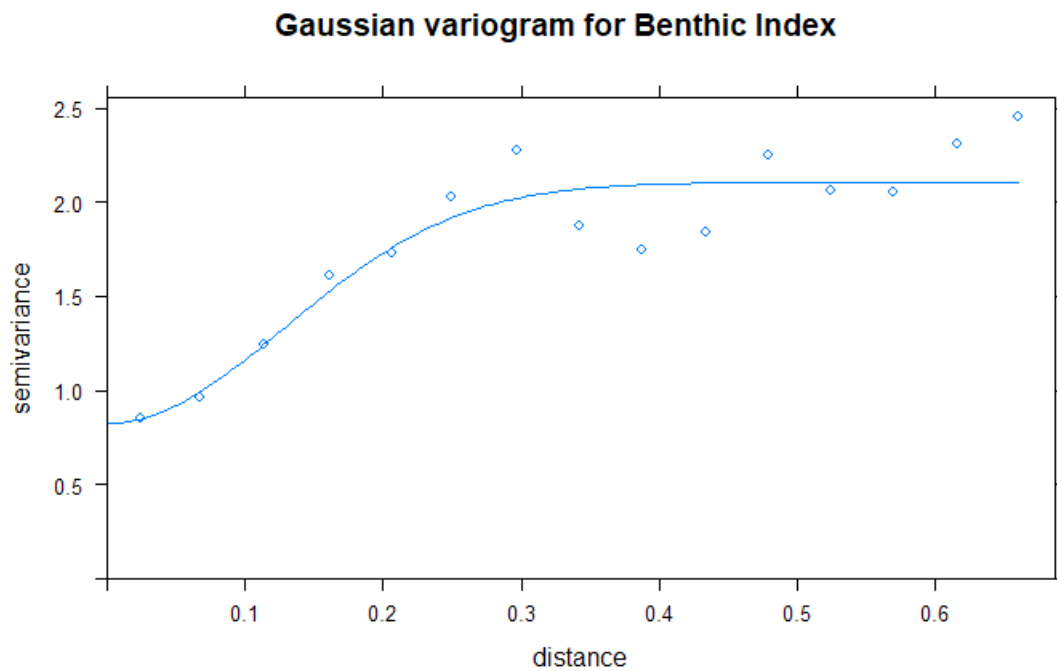
Exponential

```
benthic.vg.fit.exp <- fit.variogram(benthic.vg, model=vgm(1,"Exp", 0.5,1))  
plot(benthic.vg, benthic.vg.fit.exp, main="Exponential variogram for Benthic Index")
```



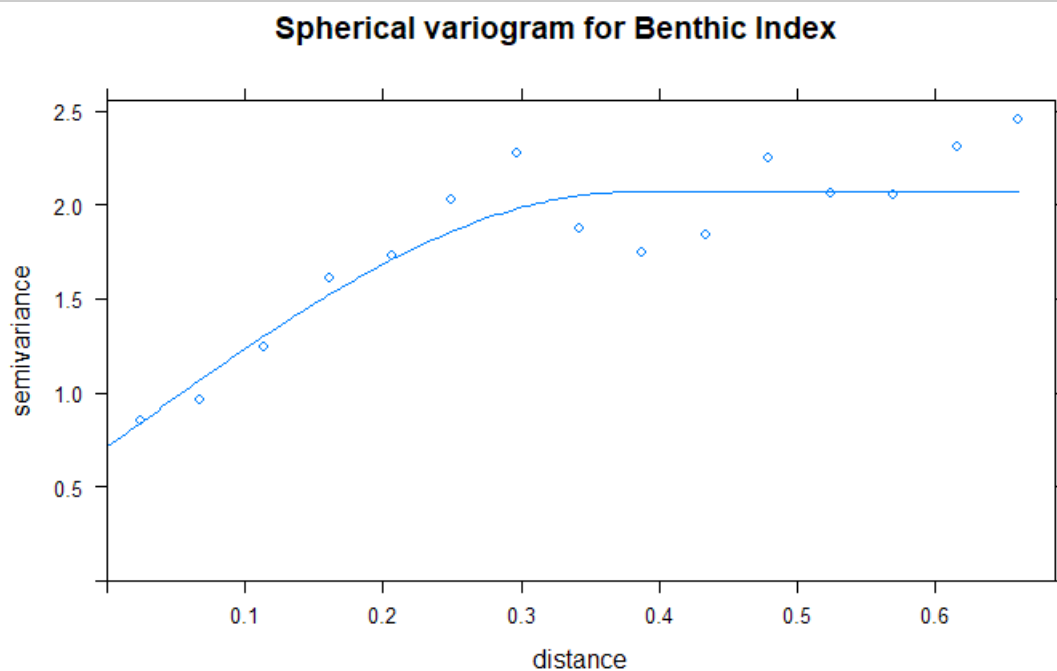
Gaussian

```
benthic.vg.fit.gau <- fit.variogram(benthic.vg, model=vgm(1,"Gau", 0.5,1))  
plot(benthic.vg, benthic.vg.fit.gau, main="Gaussian variogram for Benthic Index")
```



Spherical

```
benthic.vg.fit.sph <- fit.variogram(benthic.vg, model=vgm(1,"Sph", 0.5,1))  
plot(benthic.vg, benthic.vg.fit.sph, main="Spherical variogram for Benthic Index")
```

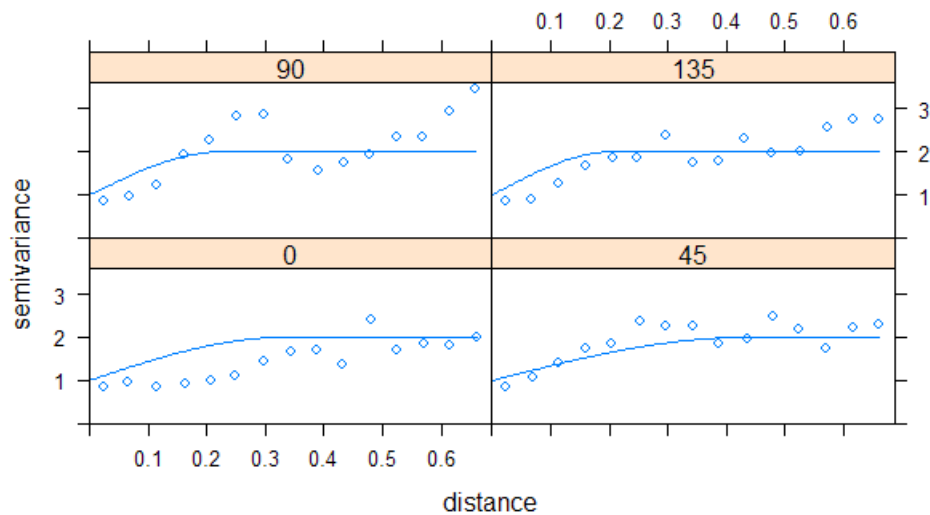


Directional Variogram

A directional variogram can be created with the following code:

```
benthic.dvg <- variogram(Index ~ 1,  
                          data=benthic, alpha=c(0, 45,90,135))  
benthic.dvg.fit <- vgm(1,"Sph", 0.5,1,anis=c(30,0.4))  
plot(benthic.dvg, benthic.dvg.fit,  
     main="Figure 11.5 Directional variograms for Benthic Index")
```

Figure 11.5 Directional variograms for Benthic Index



Exercise 11.1

Table of Contents

| | |
|--|-----|
| Preamble..... | 151 |
| Import Library | 151 |
| Create Assignments | 151 |
| Inspect the Data Set..... | 152 |
| Assign Co-ordinates | 152 |
| Create a variogram..... | 152 |
| Plot the Variogram..... | 152 |
| Create a base plot of that variogram | 152 |
| Create an attractive plot in ggplot..... | 153 |
| Create Variogram Models..... | 154 |
| Exponential Model..... | 154 |
| Gaussian Model: | 155 |
| Spherical Model: | 156 |
| Directional Variogram | 157 |

Preamble

Import Library

The following code will automatically install packages that are not already installed and load them into the library:

```
if(require('pacman')){  
  library('pacman')  
}else{  
  install.packages('pacman')  
  library('pacman')  
}  
  
## Loading required package: pacman  
pacman::p_load(sp, EnvStats, gstat, ggplot2, rmarkdown)
```

Create Assignments

Note that Benthic.df is included in the *EnvStats* package and is not a data frame but an *sp* class, which is very similar to a data frame

```
#Assignments  
benthic <- Benthic.df  
lat <- benthic$Latitude  
lon <- benthic$Longitude
```

Inspect the Data Set

```
head(benthic, 3)
```

```
##   Site.ID Stratum Latitude Longitude Index Salinity Silt
## 1 89/90-1    101  38.4430   -76.4457  2.33      2.0  0.6
## 2 89/90-1    101  38.4232   -76.4237  3.67      2.0  0.8
## 3 89/90-1    101  38.4773   -76.4827  2.00      1.5  0.6
```

Assign Co-ordinates

The benthic data set is not a mere data frame, it is an `sp` object, which is very similar to a data frame. It is necessary to set the co-ordinates of this object though:

```
coordinates(benthic) = ~Longitude + Latitude
```

Create a variogram

A variogram is a description of the structure of spacial data, first this needs to be created before anything can be plotted or modelled:

```
benthic.vg <- variogram(Index ~ 1, data = benthic)
```

Now we can inspect the variogram:

```
head(benthic.vg)
```

```
##      np      dist      gamma dir.hor dir.ver  id
## 1 4545 0.02410565 0.8500134      0      0 var1
## 2 4149 0.06703996 0.9606919      0      0 var1
## 3 3883 0.11396396 1.2417457      0      0 var1
## 4 4023 0.16138943 1.6066269      0      0 var1
## 5 5963 0.20615519 1.7325016      0      0 var1
## 6 6732 0.24983031 2.0285331      0      0 var1
```

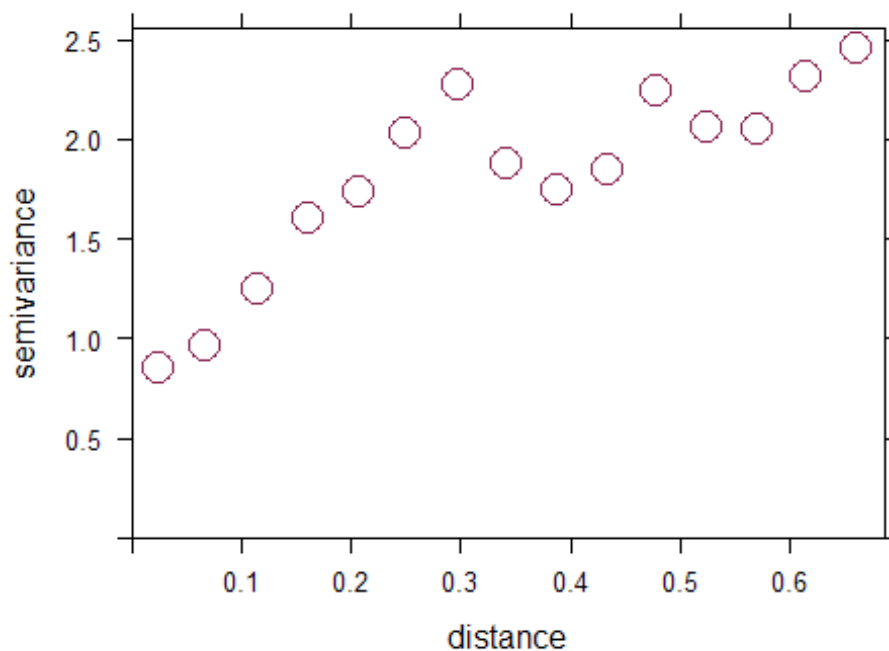
By observing this, we can conclude, that a variogram will have an *x*-axis of `dist` and a *y*-axis of `gamma`.

Plot the Variogram

Create a base plot of that variogram

```
plot(benthic.vg, main="Figure 11.1 Empirical variogram for Benthic Index",
     ,
     col = 'violetred4', cex = 2)
```

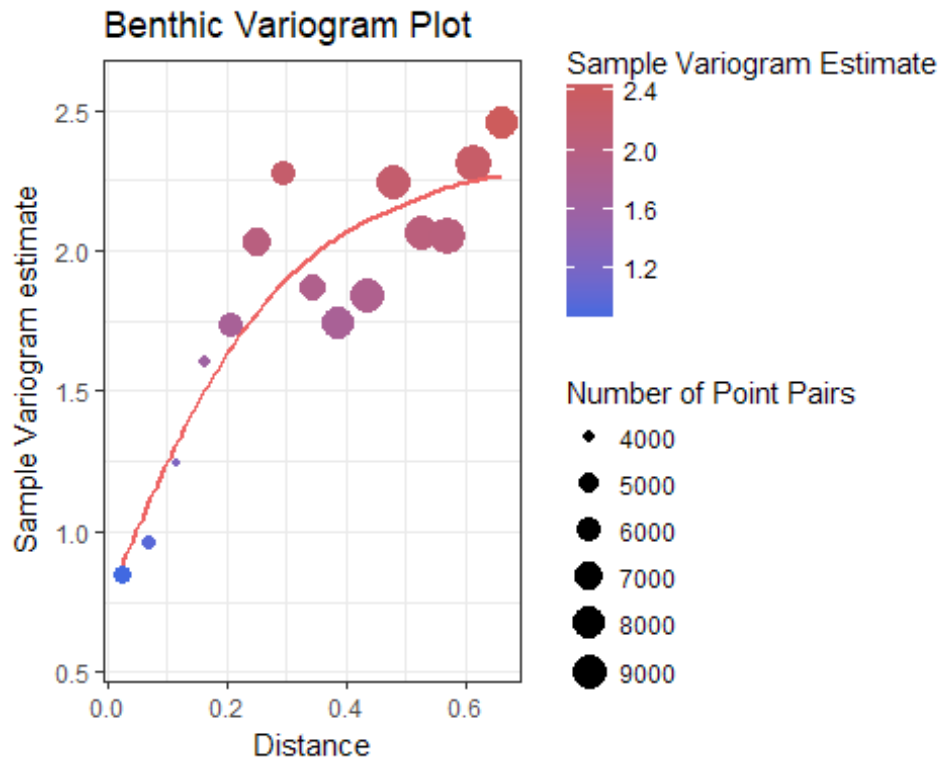

Figure 11.1 Empirical variogram for Benthic Index



Create an attractive plot in ggplot

The ggplot2 package makes it reasonable easy to add a non-parametric model using the *loess* method:

```
ggplot(data = benthic.vg, aes(x = dist, y = gamma, col = gamma, size = np)) +
  geom_point() +
  scale_color_continuous(low = 'royalblue', high = 'indianred') +
  labs(size = "Number of Point Pairs", col = "Sample Variogram Estimate",
       y = "Sample Variogram estimate", x = "Distance",
       title = 'Benthic Variogram Plot') +
  theme_bw() +
  geom_smooth(se = TRUE, fill = 'purple', alpha = 0,
             size = 1, span = 2, col = 'indianred2')
## `geom_smooth()` using method = 'loess'
```



Create Variogram Models

Variogram models are objects of the class `variogramModel` and cannot be drawn with `ggplot2`, hence only base plots can be used to draw a parametric model over the variogram plot.

Exponential Model

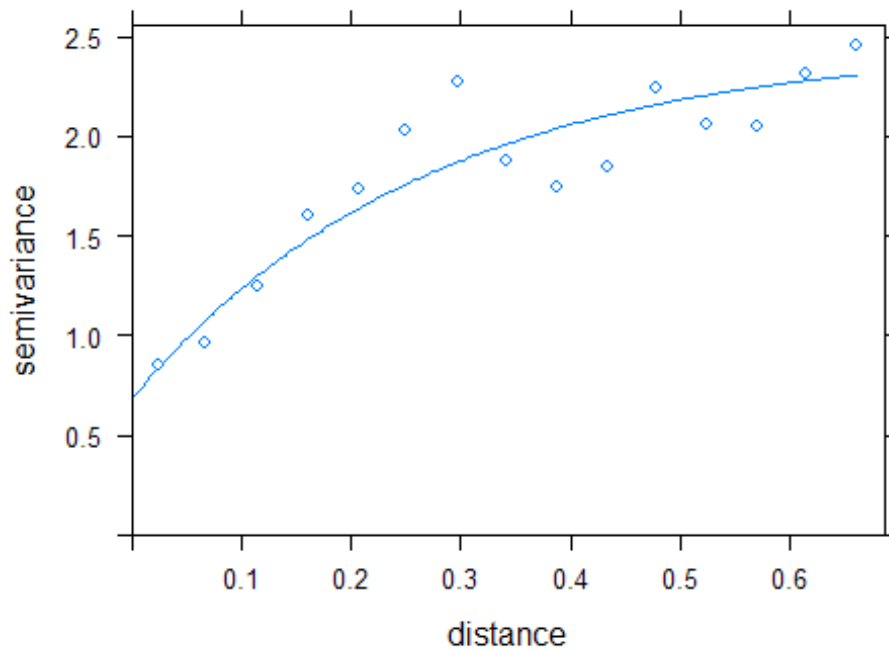
An exponential model is of the form:

$$C(h) = \sigma^2 \times e^{\frac{-h}{r}}$$

and can be modelled:

```
benthic.vg.fit.exp <- fit.variogram(benthic.vg,
                                   model=vgm(1,"Exp", 0.5,1))
plot(benthic.vg, benthic.vg.fit.exp,
     main="Exponential variogram for Benthic Index")
```

Exponential variogram for Benthic Index



Gaussian Model:

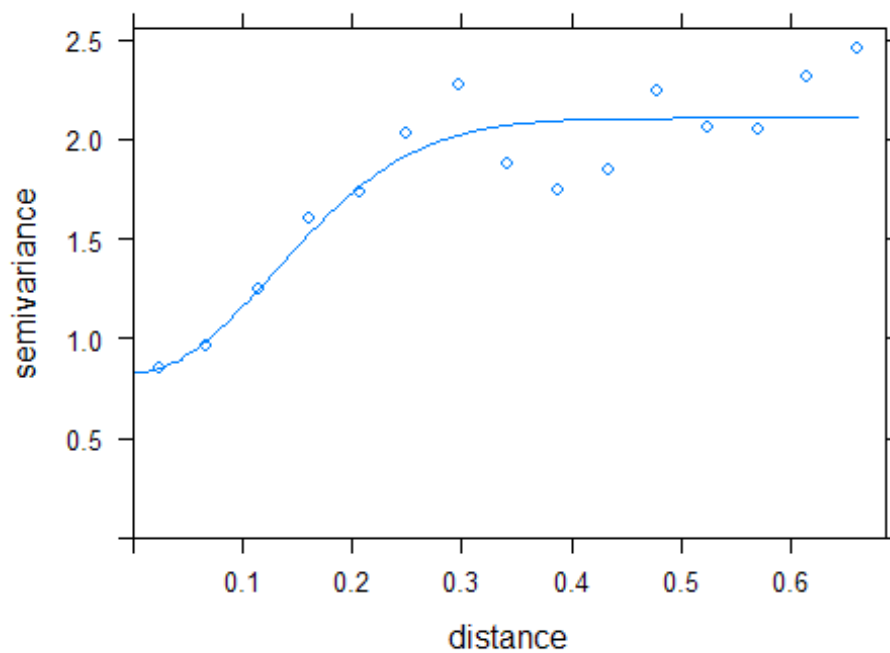
A Gaussian model is of the form:

$$C(h) = \sigma^2 \times e^{\left(\frac{-h}{r}\right)^2}$$

and can be modelled:

```
benthic.vg.fit.gau <- fit.variogram(benthic.vg,  
                                   model=vgm(1,"Gau", 0.5,1))  
plot(benthic.vg, benthic.vg.fit.gau,  
     main="Gaussian variogram for Benthic Index")
```

Gaussian variogram for Benthic Index



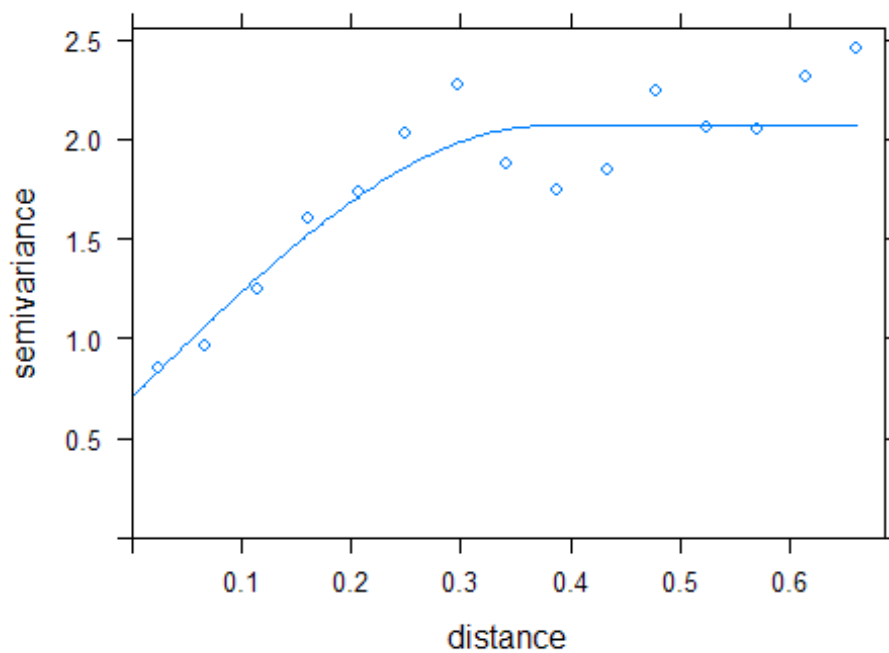
Spherical Model:

A spherical model is of the form:

$$C(h) = \sigma^2 \times \left(1 - \frac{3h}{2r} + \frac{h^2}{2r^3}\right)$$

```
benthic.vg.fit.sph <- fit.variogram(benthic.vg,  
                                   model=vgm(1,"Sph", 0.5,1))  
plot(benthic.vg, benthic.vg.fit.sph,  
     main="Spherical variogram for Benthic Index")
```

Spherical variogram for Benthic Index



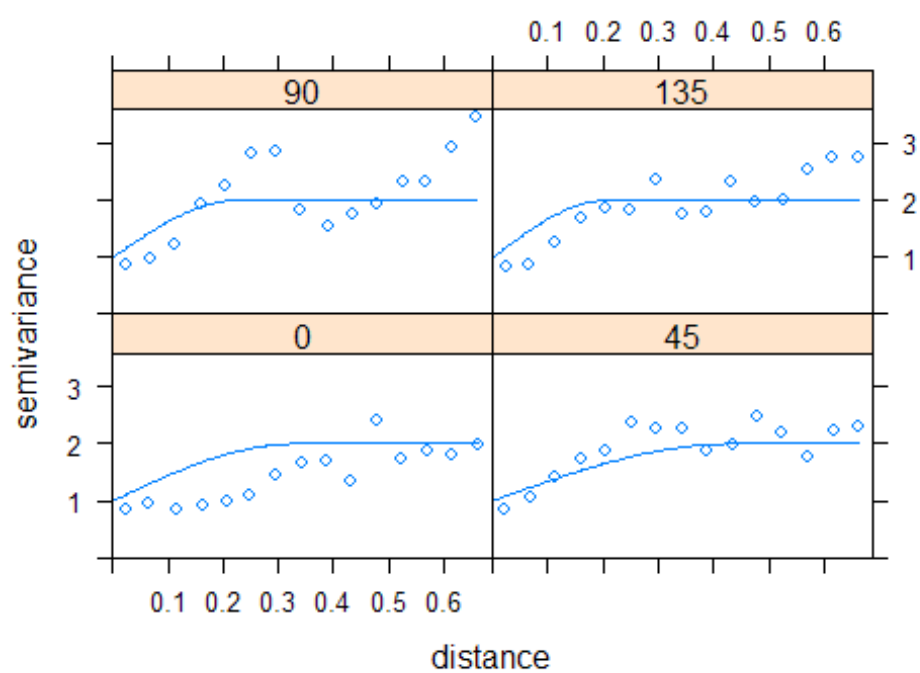
Directional Variogram

Because a directional variogram is still of the class `variogramModel`, it is not possible (to the best of my knowledge) to use `ggplot2` in order to plot it.

The `alpha` option allows a vector of directional degrees to be specified, in this case, 0, 45, 90 and 135.

```
benthic.dvg <- variogram(Index ~ 1,
                        data=benthic, alpha=c(0, 45, 90, 135))
benthic.dvg.fit <- vgm(1, "Sph", 0.5, 1, anis=c(30, 0.4))
plot(benthic.dvg, benthic.dvg.fit,
     main="Figure 11.5 Directional variograms for Benthic Index")
```

Figure 11.5 Directional variograms for Benthic Index



Predicting Geostatistical Data

Week 13 Material (9 Oct 2017); Due (16 Oct 2017)

Spatial prediction is about predicting the value at some location given the values at other locations.

Generalised Leas Squares

If:

- The model for the geostatistical data comes from a probability distribution with fixed parameters (i.e. is parametric), and
- The errors are not correlated

The standard least squares method can be used to estimate the coefficients.

Then values at a location can be predicted from co-ordinates. However caution should be exercised when extrapolating outside the range of the predictors.

Kriging

The Kriging method was developed for mining in the 1960s.

Universal kriging is a method of predicting values, whereby the random error ε is also predicted.

Ordinary kriging, (usually just called kriging), assumes a constant trend surface

For any location s_0 , the kriging prediction (i.e. ordinary kriging) of the response variable is given by:

$$\hat{Z}(S_0) = \sum_{i=1}^n [\lambda_i \cdot z(s_i)]$$

Where:

$$\lambda : \sum_{i=1}^n [\lambda_i] = 1$$

i.e. λ is the average value of all the observed values.

sp Package

The `sp` package uses S4 to define its methods and classes.

An `sp` object is similar to a data frame and is an object ideal for storing spatial data.

gstat

The `gstat` package provides tools to work with spatial data.

Exercise 12.1

Table of Contents

| | |
|--|-----|
| Preamble | 161 |
| Import Library | 161 |
| Create Assignments | 161 |
| Inspect the Data Set | 161 |
| Latitude and Longitude | 161 |
| Assign the Co-ordinates | 162 |
| Preliminary Step: Create a kriging prediction Model | 162 |
| Create a variogram Model | 162 |
| Create a Variogram | 162 |
| Fit an exponential model to that variogram | 162 |
| Create a Grid of Domain values to use as predictor values | 163 |
| Create a sequence of incrementally increasing for the X and Y axis | 163 |
| Combine into a domain base | 163 |
| Create the Kriging prediction object | 163 |
| Part 1: Create a heatmap | 163 |
| Create it in ggplot2 | 164 |
| Map overlay | 166 |
| Part 2: Contour Map | 167 |
| Base Package | 168 |
| Using ggplot2 | 168 |
| Map Overlay | 170 |
| Map Overlay with Heatmap | 171 |
| Part 3: Surface Plots | 172 |
| Base | 172 |
| Plotly | 173 |

The goal here is to create a prediction model for the value of benthic over the Chesapeake bay region.

This prediction model will be graphically portrayed with a:

1. Heatmap
 - i) with base plot
 - ii) with ggplot2 over an actual map
2. Contour plot

- i) with base plot
 - ii) With `ggplot2` over an actual map
- 3. 3d Surface plot
 - i) with base plot
 - ii) with a dynamic `plotly` surface plot

Preamble

Import Library

The following code will automatically install packages that are not already installed and load them into the library:

```
if(require('pacman')){
  library('pacman')
}else{
  install.packages('pacman')
  library('pacman')
}

## Loading required package: pacman

## Warning: package 'pacman' was built under R version 3.5.2

pacman::p_load(sp, EnvStats, gstat, ggplot2, rmarkdown, reshape2, ggmap, R
ColorBrewer, parallel, dplyr, plotly)
```

Create Assignments

Note that `Benthic.df` is included in the `EnvStats` package and is not a data frame but an `sp` class, which is very similar to a data frame

```
#Assignments
benthic <- Benthic.df
lat  <- benthic$Latitude
lon  <- benthic$Longitude
index <- benthic$Index
```

Inspect the Data Set

```
head(benthic, 3)
```

```
##   Site.ID Stratum Latitude Longitude Index Salinity Silt
## 1 89/90-1     101  38.4430  -76.4457  2.33      2.0  0.6
## 2 89/90-1     101  38.4232  -76.4237  3.67      2.0  0.8
## 3 89/90-1     101  38.4773  -76.4827  2.00      1.5  0.6
```

Latitude and Longitude

- Latitude is a line that runs east to west that measures how far north of the equator a location is
 - Latitude is assigned to the y-axis
- Longitude is a line that runs north to south that measures east/west
 - Longitude is assigned to the x-axis

Assign the Co-ordinates

The *Benthic* Data set is an `sp` class objects and requires for its co-ordinates to be set:

```
coordinates(benthic) = ~Longitude + Latitude
```

Preliminary Step: Create a kriging prediction Model

In order to create a kriging prediction model two things are required:

- Fitted Variogram Model (e.g. an exponential model)
- A domain values over which to predict values

Create a variogram Model

Create a Variogram

If a 4th degree polynomial model was used to model the benthic values over space, a variogram can be made from that model thusly:

(The quadratic model is a required parameter of the kriging process, in exercise 12.2 a quadratic model will be used, the specific mathematics of this though is somewhat beyond me, however the kriging method uses the residuals from that surface model in order to make the predictions.)

```
x <- lon
y <- lat

benthic.vg <- variogram(object = index ~
  x + y +
  x^2 + x*y + y^2 +
  x^3 + x^2*y + x*y^2 + y^3 +
  x^4 + x^3*y + x^2*y^2 + x*y^3 + y^4,
  data = benthic)
```

```
head(benthic.vg)
```

| ## | np | dist | gamma | dir.hor | dir.ver | id |
|------|------|------------|-----------|---------|---------|------|
| ## 1 | 4545 | 0.02410565 | 0.8506164 | 0 | 0 | var1 |
| ## 2 | 4149 | 0.06703996 | 0.9393502 | 0 | 0 | var1 |
| ## 3 | 3883 | 0.11396396 | 1.1513831 | 0 | 0 | var1 |
| ## 4 | 4023 | 0.16138943 | 1.4599238 | 0 | 0 | var1 |
| ## 5 | 5963 | 0.20615519 | 1.5641015 | 0 | 0 | var1 |
| ## 6 | 6732 | 0.24983031 | 1.8992994 | 0 | 0 | var1 |

Fit an exponential model to that variogram

```
benthic.vg.fit.exp <- fit.variogram(benthic.vg, model=vgm(1,"Exp", 0.5,1))
```

Thus the variogram model needed for the kriging prediction has been ascertained

Create a Grid of Domain values to use as predictor values

Create a sequence of incrementally increasing for the X and Y axis

This will essentially act as the resolution of the overlay, Most monitors are about 140 ppi, hence assuming the plot will be a 9 inch square, about 800 x 800 pixels should look right, but 200 x 200 will be the used due to performance.

```
xgrid      <- seq(min(lon), max(lon), length.out = 40)
ygrid      <- seq(min(lat), max(lat), length.out = 40)
```

Combine into a domain base

This domain base however has to be a spatial object, this can be achieved by defining co-ordinates in the data frame

```
xy.surface <- expand.grid(lon = xgrid, lat = ygrid)
coordinates(xy.surface) = ~lon + lat
head(xy.surface, 3)

## SpatialPoints:
##      lon      lat
## 1 -77.3157 38.0018
## 2 -77.2794 38.0018
## 3 -77.2431 38.0018
## Coordinate Reference System (CRS) arguments: NA
```

Create the Krige prediction object

```
benthic.pred <- krige(formula = Index ~1,
                      locations = benthic,
                      newdata = xy.surface,
                      model= benthic.vg.fit.exp)

## [using ordinary kriging]

head(benthic.pred)

##      coordinates var1.pred var1.var
## 1 (-77.3157, 38.0018)  3.144851 2.321630
## 2 (-77.2794, 38.0018)  3.120608 2.295719
## 3 (-77.2431, 38.0018)  3.086702 2.269852
## 4 (-77.2068, 38.0018)  3.041955 2.243451
## 5 (-77.1705, 38.0018)  2.985132 2.215685
## 6 (-77.1342, 38.0018)  2.914961 2.185591
```

Now there is a prediction model, if that prediction model is extrapolated over our domain (xy.surface) the various plots can be created

Part 1: Create a heatmap

In order to create a heatmap, it is first necessary to take the sp prediction object and make it into pixel data, otherwise the heatmap won't look quite right:

```
class(benthic.pred)
```

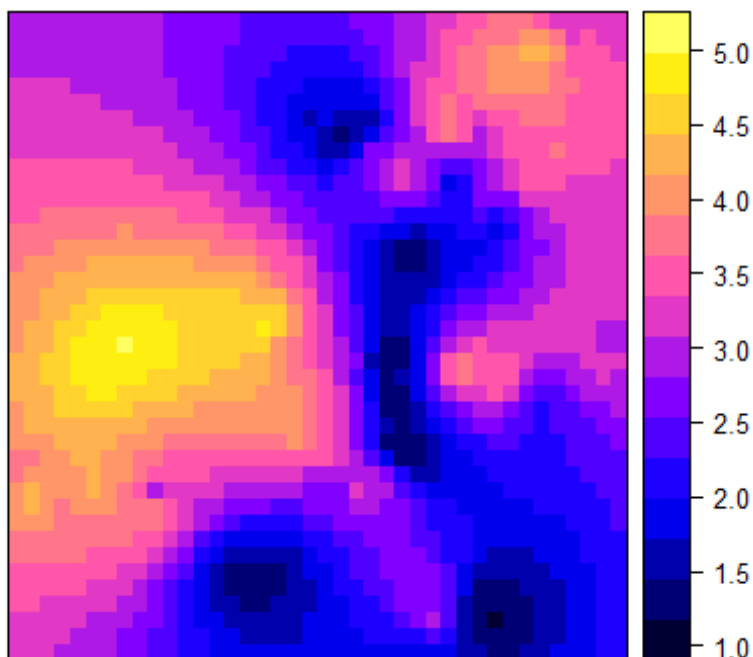
```
## [1] "SpatialPointsDataFrame"
## attr(,"package")
## [1] "sp"

gridded(benthic.pred) = TRUE
class(benthic.pred)

## [1] "SpatialPixelsDataFrame"
## attr(,"package")
## [1] "sp"

spplot(benthic.pred["var1.pred"],main="Predictions of Benthic Index - ordinary kriging")
```

Predictions of Benthic Index - ordinary kriging



Create it in ggplot2

First it is necessary to create a data frame from the `sp` object:

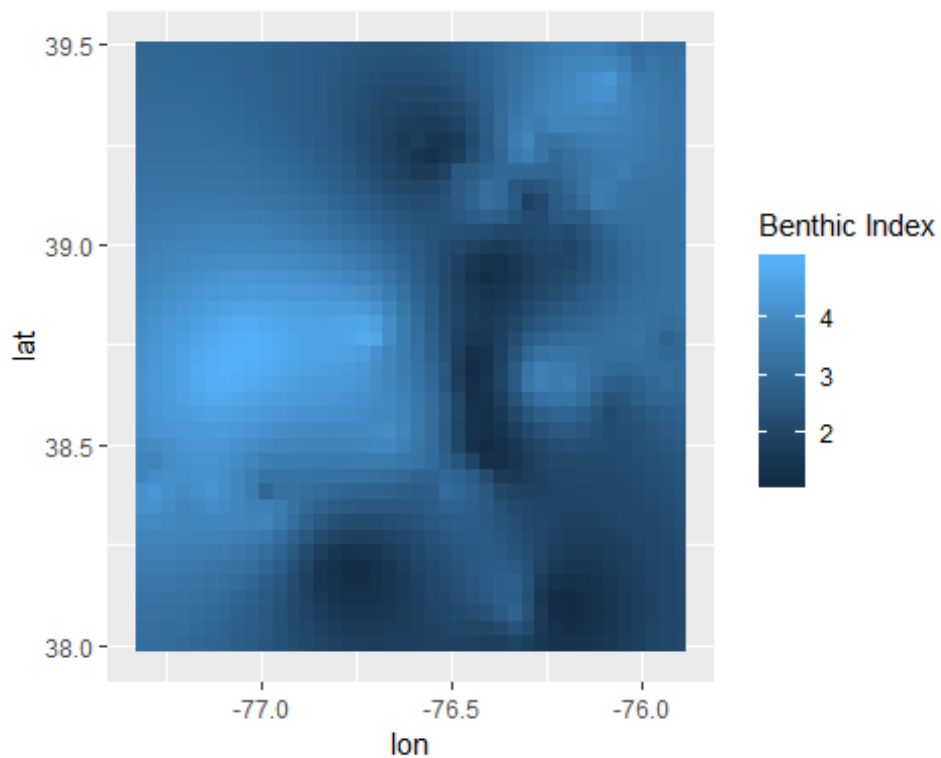
```
benthic.pred.df      <- cbind(benthic.pred@coords, benthic.pred@data)
benthic.pred.df.tidy <- melt(benthic.pred.df)

## No id variables; using all as measure variables

head(benthic.pred.df)

##      lon      lat var1.pred var1.var
## 1 -77.3157 38.0018  3.144851 2.321630
## 2 -77.2794 38.0018  3.120608 2.295719
## 3 -77.2431 38.0018  3.086702 2.269852
## 4 -77.2068 38.0018  3.041955 2.243451
## 5 -77.1705 38.0018  2.985132 2.215685
## 6 -77.1342 38.0018  2.914961 2.185591
```

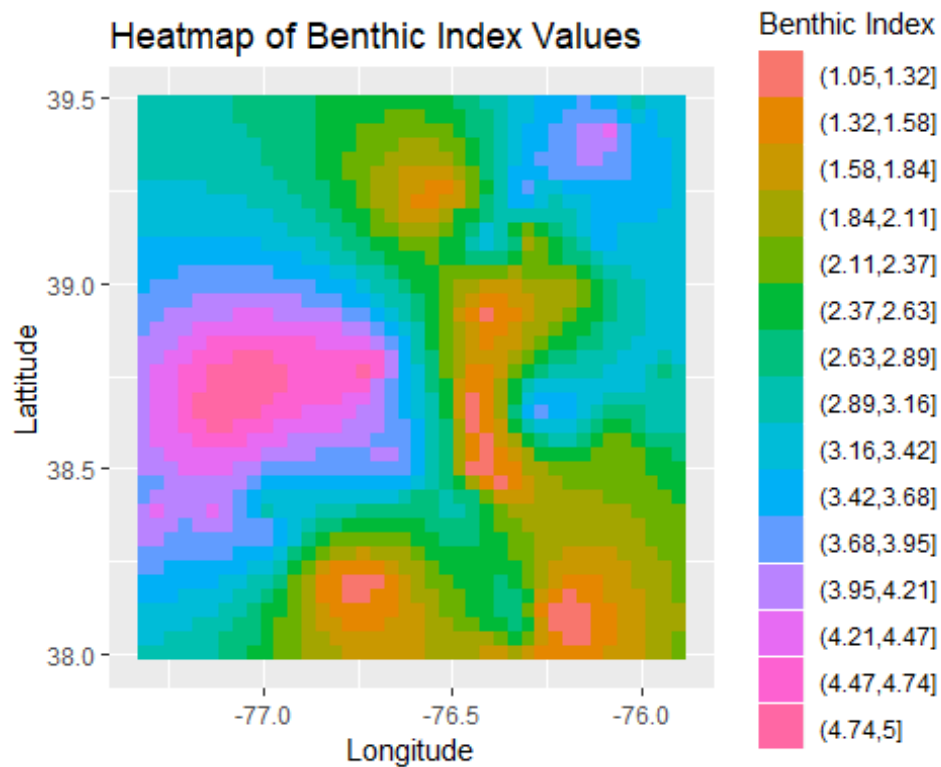
```
ggplot() +
  geom_tile(data = benthic.pred.df, aes(lon, lat, fill = var1.pred)) +
  labs(fill = "Benthic Index") #
```



This plot would be easier to read if it had discrete intervals rather than a continuous scale:

```
benthic.pred.df$bin <- cut(benthic.pred.df$var1.pred,
                           breaks = as.vector(seq(from = 0, to = 5, length.out = 20)))

ggplot() +
  geom_tile(data = benthic.pred.df, aes(lon, lat, fill = bin)) +
  labs(fill = "Benthic Index",
       title = "Heatmap of Benthic Index Values",
       x = "Longitude",
       y = "Latitude")
```



Map overlay

Now by adjusting the colour scale and opacity, this can be used as an overlay for the map from exercise 10.1:

```
benthic.pred.df$bin <- cut(benthic.pred.df$var1.pred,
                           breaks = as.vector(seq(from = 0, to = 5, length.out = 9)))
```

#Get a map

```
bbox <- make_bbox(benthic.pred.df$lon, benthic.pred.df$lat, f = 0.01)
map <- get_map(location = bbox, maptype = 'toner')
```

maptype = "toner" is only available with source = "stamen".

resetting to source = "stamen"...

Map from URL : <http://tile.stamen.com/toner/9/146/194.png>

Map from URL : <http://tile.stamen.com/toner/9/147/194.png>

Map from URL : <http://tile.stamen.com/toner/9/148/194.png>

Map from URL : <http://tile.stamen.com/toner/9/146/195.png>

Map from URL : <http://tile.stamen.com/toner/9/147/195.png>

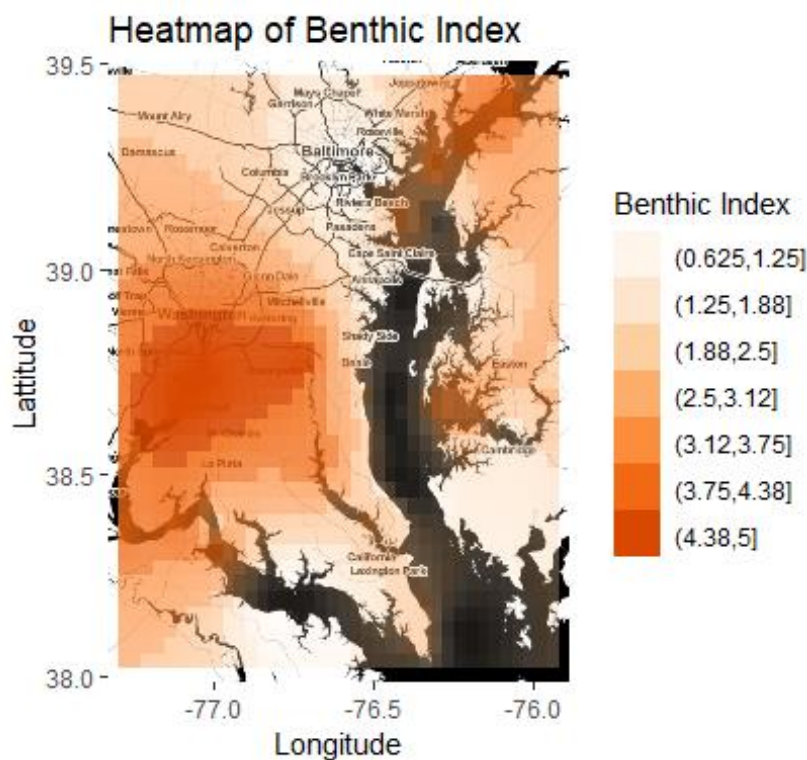
Map from URL : <http://tile.stamen.com/toner/9/148/195.png>

Map from URL : <http://tile.stamen.com/toner/9/146/196.png>

Map from URL : <http://tile.stamen.com/toner/9/147/196.png>

```
## Map from URL : http://tile.stamen.com/toner/9/148/196.png
## Map from URL : http://tile.stamen.com/toner/9/146/197.png
## Map from URL : http://tile.stamen.com/toner/9/147/197.png
## Map from URL : http://tile.stamen.com/toner/9/148/197.png

#Create the colours
cols <- brewer.pal(n = length(levels(benthic.pred.df$bin)), name = "Oranges")
#Create the Plot
ggmap(ggmap = map) +
  geom_tile(data = benthic.pred.df, aes(lon, lat, fill = bin, alpha = var1
.pred)) +
  scale_fill_manual(values = cols) +
  labs(fill = "Benthic Index",
       x = "Longitude",
       y = "Latitude",
       title = "Heatmap of Benthic Index") +
  guides(alpha=FALSE)
```



Part 2: Contour Map

In order to view the surface plot, contours can also be used.

Recall from Week 11/Exercise 10, that contour and surface plots work a little differently to ordinary plots.

A domain grid of equally spaced x and y values is required and a matrix of z values that can be overlaid onto that domain grid in order to provide the surface values required to create the surface plot.

The method from Lecture 10/Wk. 11 involved predicting over the square domain of `xy.surface`, this won't work here because instead of the `predict` function, the `krige` function was used, hence it is necessary to convert the output from the `krige` function, into the output that would have been given by the `predict` function, which is the required z surface matrix.

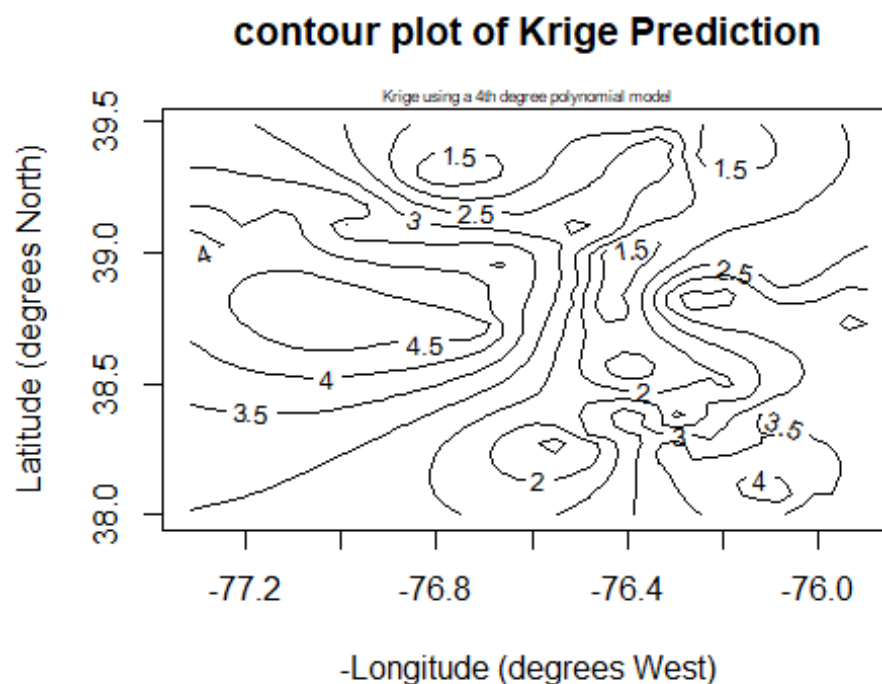
This can be achieved with the `data.matrix` function:

```
z_surface_matrix <- data.matrix(benthic.pred[1])
```

Base Package

Using base R packages, a contour plot can be created with the `contour` function:

```
contour(xgrid, ygrid, z_surface_matrix,  
        xlab="-Longitude (degrees West)", ylab="Latitude (degrees North)",  
        ,  
        main = "contour plot of Krige Prediction",  
        labcex = 0.8)  
mtext("Krige using a 4th degree polynomial model", cex = 0.5)
```



Using ggplot2

In order to create a contour plot in ggplot 2, it will be necessary to use 'tidy' data, which is a table of data such that each column corresponds to a variable:


```

rownames(z_surface_matrix) <- xgrid
colnames(z_surface_matrix) <- ygrid
benthic.pred.melt <- melt(z_surface_matrix,
                          varnames = c("Longitude", "Latitude"),
                          value.name = "Benthic_Index")

head(benthic.pred.melt)

##   Longitude Latitude Benthic_Index
## 1  -77.3157   38.0018      2.990251
## 2  -77.2794   38.0018      2.982310
## 3  -77.2431   38.0018      2.971759
## 4  -77.2068   38.0018      2.958286
## 5  -77.1705   38.0018      2.941581
## 6  -77.1342   38.0018      2.921350

```

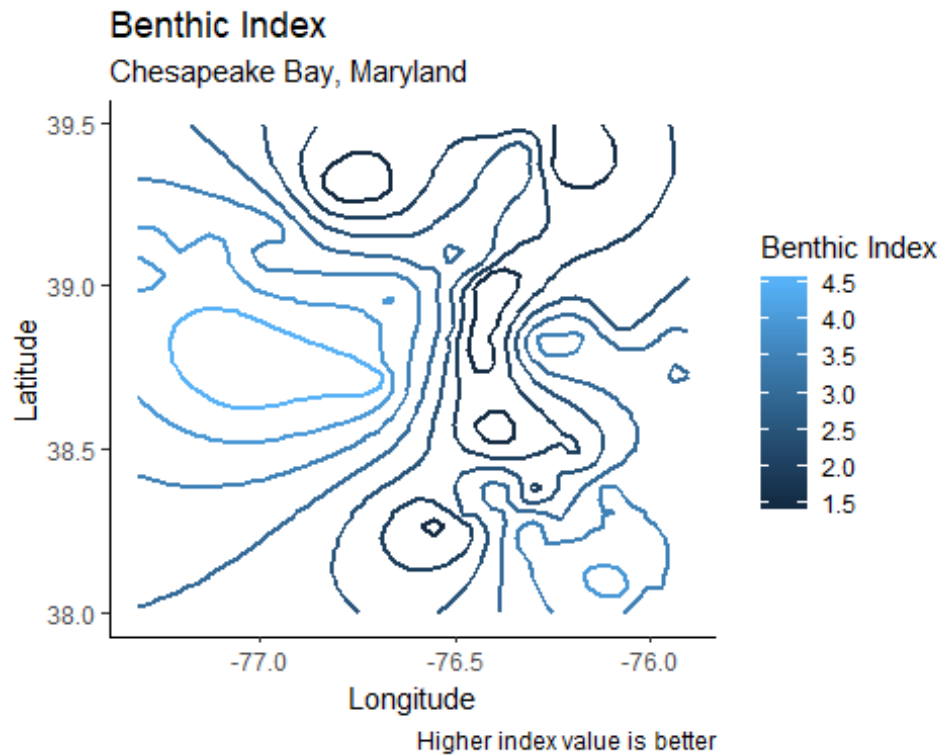
Observe that this is another way of getting the data frame that was required in part 1 (benthic.pred.df), it is included here as another method and for the sake of completion, also it matches the method used in week 10/lecture 11.

Now using ggplot2 a contour plot, can be created:

```

ggplot(data = benthic.pred.melt, aes(x = Longitude,
                                     y = Latitude,
                                     z = Benthic_Index,
                                     colour = ..level..),
       size = 0.5) +
  labs(x = "Longitude", y = 'Latitude',
       title = 'Benthic Index',
       subtitle = "Chesapeake Bay, Maryland",
       caption = "Higher index value is better",
       col = 'Benthic Index', size = 'Benthic Index') +
  geom_contour(lwd = 1) +
  theme_classic()

```

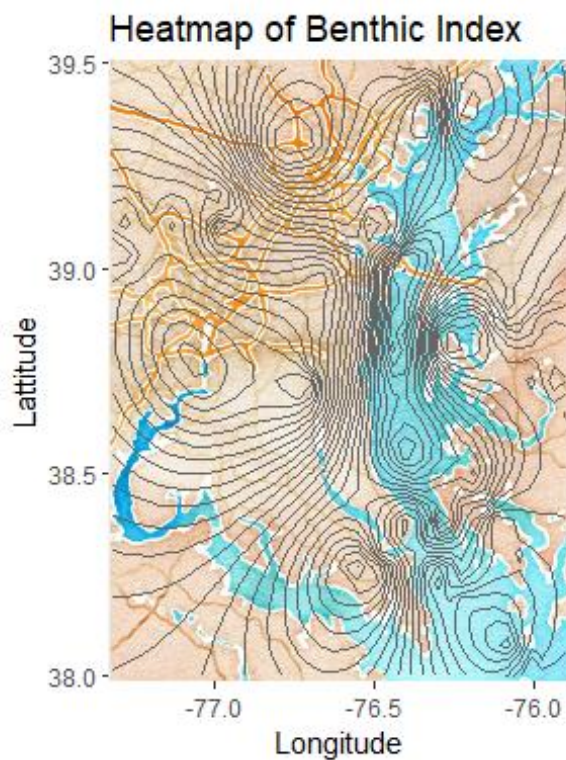


Map Overlay

The advantage to using ggplot2, is the map overlay can be taken advantage of:

```
map <- get_map(location = bbox, maptype = 'watercolor')  
## maptype = "watercolor" is only available with source = "stamen".  
## resetting to source = "stamen"....  
## Map from URL : http://tile.stamen.com/watercolor/9/146/194.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/147/194.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/148/194.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/146/195.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/147/195.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/148/195.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/146/196.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/147/196.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/148/196.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/146/197.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/147/197.jpg  
## Map from URL : http://tile.stamen.com/watercolor/9/148/197.jpg
```

```
ggmap(ggmap = map) +
  labs(fill = "Benthic Index",
       x = "Longitude",
       y = "Latitude",
       title = "Heatmap of Benthic Index") +
  guides(alpha=FALSE) +
  geom_contour(data = benthic.pred.melt, aes(x = Longitude,
                                             y = Latitude,
                                             z = Benthic_Index),
              col = 'grey40',
              binwidth = 0.15,
              size = 0.70) +
  scale_fill_manual(values = cols)
```

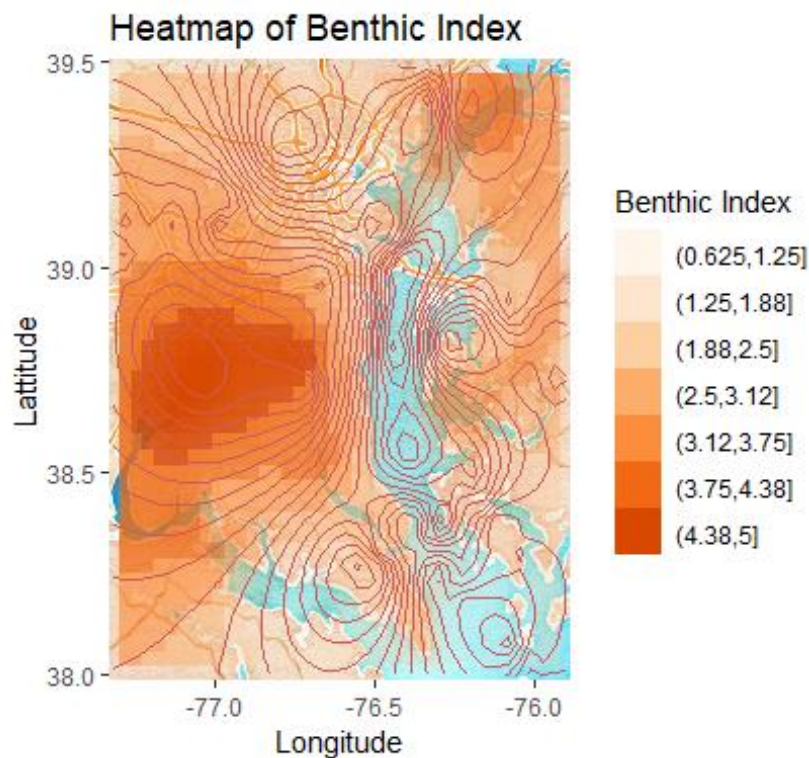


Map Overlay with Heatmap

This can be combined with the heatmap layer:

```
ggmap(ggmap = map) +
  labs(fill = "Benthic Index",
       x = "Longitude",
       y = "Latitude",
       title = "Heatmap of Benthic Index") +
  guides(alpha=FALSE) +
  geom_tile(data = benthic.pred.df, aes(lon, lat, fill = bin, alpha = va
r1.pred)) +
  geom_contour(data = benthic.pred.melt, aes(x = Longitude,
                                             y = Latitude,
                                             z = Benthic_Index),
              col = 'indianred',
              binwidth = 0.2,
```

```
size = 0.6) +
scale_fill_manual(values = cols)
```



Part 3: Surface Plots

A surface plot allows a three dimensional view of the model

Base

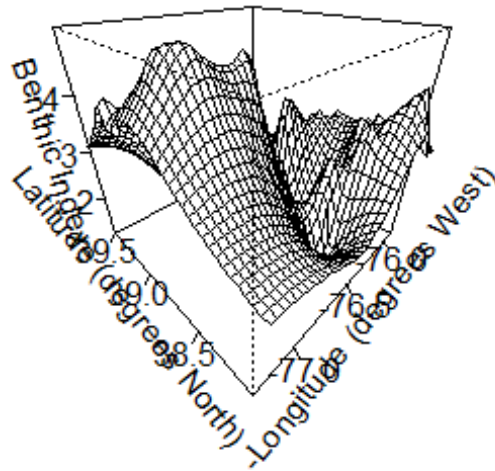
In base package, a surface plot can be made:

```
persp(xgrid, ygrid, z_surface_matrix,
      xlim = c(-77.3, -75.9), #the values domain needs to be adjusted
      above
      ylim = c(38.1, 39.5),    #Make sure to set an appropriate do
      main
      # zlim = c(0, 6),
      theta = -45, phi = 30, d = 0.5,
      xlab="-Longitude (degrees West)",
      ylab="Latitude (degrees North)",
      zlab="Benthic Index", ticktype = "detailed")

## Warning in persp.default(xgrid, ygrid, z_surface_matrix, xlim = c(-77.3
, :
## surface extends beyond the box

title(main=paste("Surface Plot of Benthic Index", "Loess Smoothing", sep
="\n"))
```

Surface Plot of Benthic Index Loess Smoothing



Plotly

Plotly has the advantage of being fully interactive and not as difficult to constrain within a plot/box area:

```
p <- plot_ly(x = ygrid, y = xgrid, z = z_surface_matrix) %>%
  add_surface() %>%
  layout(
    title = "Benthic Index in Chesapeake Bay",
    scene = list(
      xaxis = list(title = "Longitude",
        range = c(38.2, 38.9)), #You shouldn't need to edit
the
      yaxis = list(title = "Latitude",
e
        range = c(-77, -75.9)),
      zaxis = list(title = "Benthic Index")
    ))

#p
#This output is HTML, it cannot be rendered inside a word document
```

The surface plot really reinforces the fact that the bay has the lowest (i.e. best) benthic index value.