# Time Series Analysis MS4218

## Lecture 1

***Introduction***

**Kevin Burke**

kevin.burke@ul.ie

# Course

- The course will be based on “Time Series Analysis with Applications in R” (by Jonathan D. Cryer, Kung-Sik Chan).
- All material will be uploaded to http://sulis.ul.ie
- 3-hour block lecture which will be used as a mixture of lectures, labs and tutorials.
- **Working independently outside of the lecture block is required**.

# Assessment

The assessment breakdown is:

- R Project: 40% (to be submitted in Week 13)
- Final: 60%.

Grading bands:

- **A1**: 90 - 100
- **A2**: 80 - 89
- **B1**: 70 - 79
- **B2**: 60 - 69
- **B3**: 55 - 59
- **C1**: 50 - 54
- **C2**: 45 - 49
- **C3**: 40 - 44
- **D1**: 35 - 39
- **D2**: 30 - 34
- **F** : 0 - 29

# How to Install R

1. Go to http://cran.r-project.org/ and click on “Download R for Windows” at the top of the main page.
2. On the next page click “install R for the first time”.
3. At the top of the next page click “Download R for Windows”.
4. Run the downloaded executable file to install R on your computer.

# Using R - Basic Example

Now that you have installed R, click on the R icon to open it.

Once open, click on “File” in the top left corner and then “New script”.

Copy and paste the code below into the script that you have opened:

```
x = c(1,1,2,4,3,2,1,4,5,3,6,9,1,2,15)
mean(x)
sd(x)
```

Within this script file in R, highlight the copied code. Press “Ctrl + R” to run it.

This gives the *mean* and *standard deviation* (more on these later) for the vector of numbers stored in x - you should get 3.933333 and 3.788454 in the R console.

# Using R - More Information

If you wish to learn more about R, there are many options:

- Within your R script you can use the "?" command to find out more about a given function, e.g., running the code `?mean` will tell you about the `mean` function.

- At the top of the R window you will see a "help" menu. Here you can find information about various aspects of R. In particular, under the heading "Manuals (in PDF)", the "An Introduction to R" and "R Reference Manual" are useful.

- There is extensive information about R online, e.g., google "R tutorial" or "R beginners guide" etc. There are also R help forums where many solutions to common problems can be found.

# Loading Data into R

- R has some inbuilt datasets to practice on. However, we typically wish to analyse external datasets (`.txt` and `.csv` files work best with R).
- First save your dataset to a particular location on your PC, e.g., let's assume that `testdata.txt` is stored in the following location: `C:\Users\Kevin Burke\Documents\TimeSeries`
- Open R and set your "work directory" by typing into your script file:
  `setwd("C:\\Users\\Kevin Burke\\Documents\\TimeSeries")`
  (note the use of double-backslash and quotation marks in the above command)
- Load in the dataset using
  `read.table("testdata.txt",header=T)` or
  `read.csv("testdata.csv",header=T)`.

# TSA Package

- The most commonly used package for analysing time series data is the TSA pacakge; this will be used throughout the course.
- To install this package type `install.packages("TSA")`.
- Now, to load this package type `library("TSA")`.

# `ts` data class

- The TSA pacakge requires data in the "`ts`" class.

- Some examples (first generate `x <- rnorm(20)`):
  - Annual data starting in 2010:
    `xnew <- ts(x, freq=1, start=2010)`
  - Quarterly data starting in Q1 of 2010:
    `xnew <- ts(x, freq=4, start=c(2010,1))`
  - Monthly data starting in month 3 of 2010:
    `xnew <- ts(x, freq=12, start=c(2010,3))`
  - Weekly data starting in week 40 of 2010:
    `xnew <- ts(x, freq=52, start=c(2010,40))`
  - Daily data starting on day 2 of week 1:
    `xnew <- ts(x, freq=7, start=c(1,2))`

# Time Series Data

Data obtained from observations collected sequentially over time.

- **Finance**: interest/inflation rates, stock price, NASDAQ index, annual GDP.
- **Business/Manufacturing**: weekly sales/profit, quantity demanded, daily output, number of defective batches.
- **Weather**: temperature, wind speed, rainfall, river height.
- **Agriculture**: crop yield, livestock, soil erosion
- **Biology**: heart rate, body mass index, annual height/weight, daily energy expended, calorie intake.

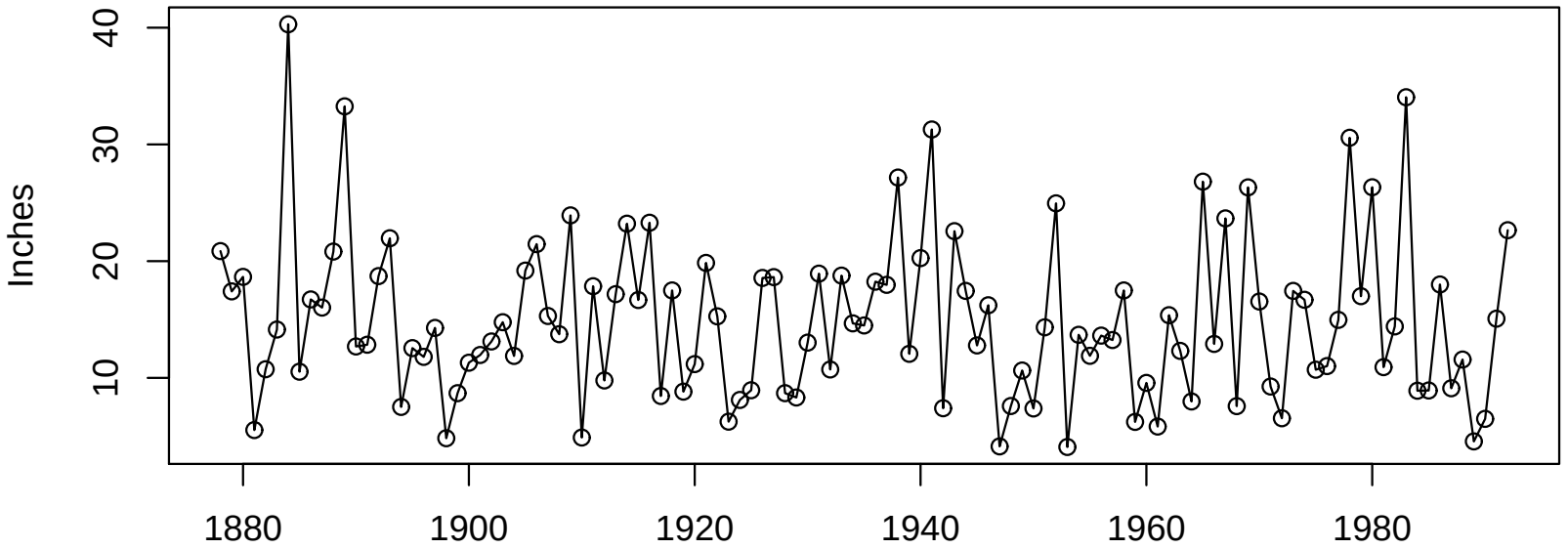
# Datasets

The `TSA` package contains a number of datasets (for full list see `data(package="TSA")`). Following Cyrer&Chan, we will focus on:

- `larain`: annual rainfall (inches) over 100 years in Los Angeles, California.
- `color`: measure of colour in consecutive batches from an industrial chemical process.
- `hare`: annual Canadian hare numbers over 31 years.
- `tempdub`: average monthly temperature over 12 years in Dubuque, Iowa.
- `oilfilters`: monthly sales of John Deere oil filters.

To load one of these datasets type, for example, `data(larain)`.

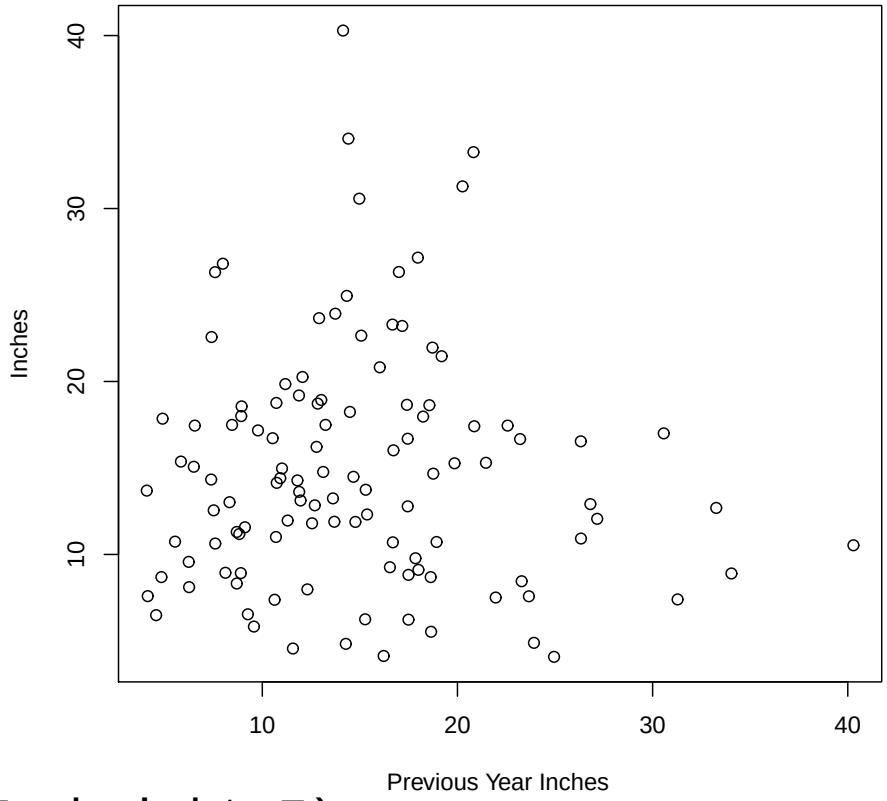
# Time Series Plot: LA Annual Rainfall




```
dev.new(width=8, height=4)
data(larain)
plot(larain,ylab='Inches',xlab='Year',type='o')
```

Hard to see any clear pattern.

# Scatterplot: Rain Versus Previous Year's Rain

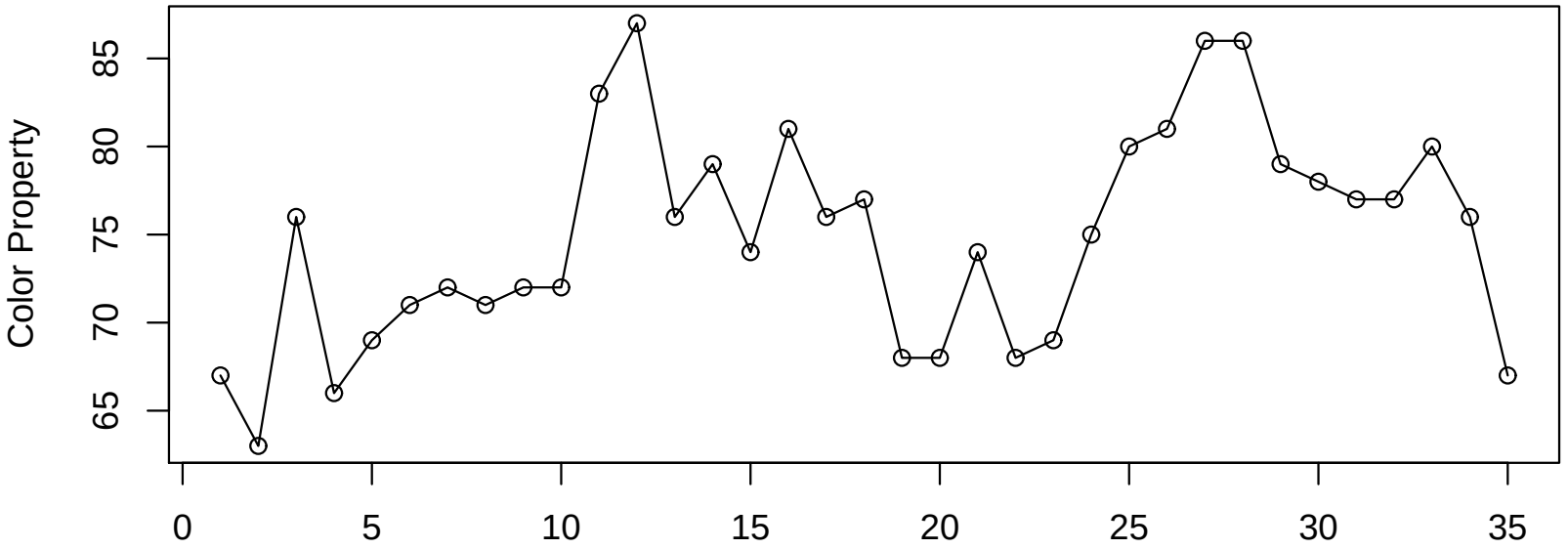



```
dev.new(width=7, height=7)
plot(y=larain,x=zlag(larain,d=1),
ylab='Inches', xlab='Previous Year Inches')
cor(y=larain,x=zlag(larain,d=1), use="pairwise")
```

Very little correlation between last year's rainfall and this year's ($r \approx 0$).

Note: this is called **autocorrelation**, i.e., correlation with itself at different time points.

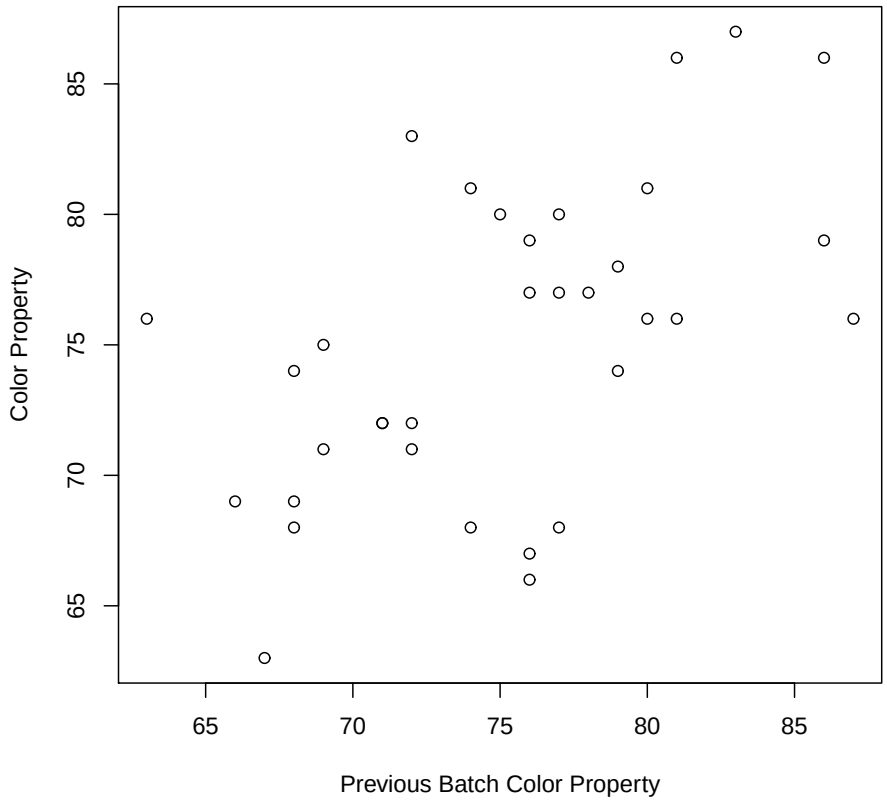
# Time Series Plot: Colour Property in a Chemical Process




```
dev.new(width=8, height=4)
data(color)
plot(color,ylab='Color Property',xlab='Batch',type='o')
```

Neighbouring values in time tend to be similar in size.

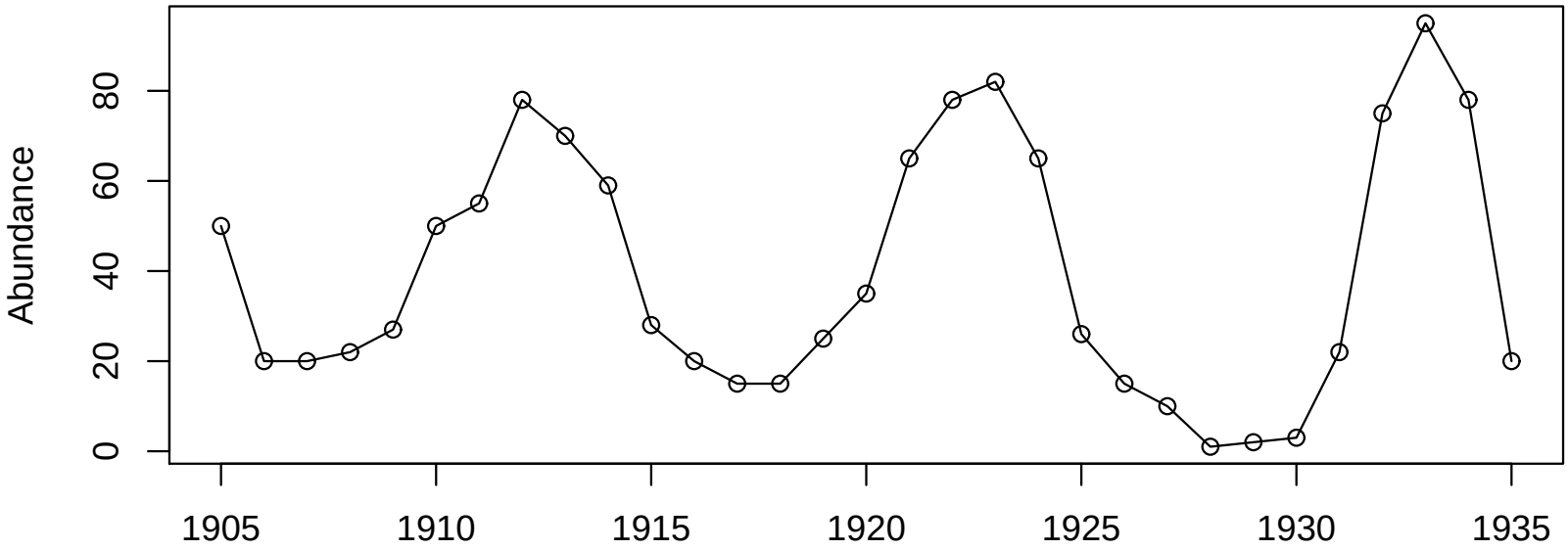
# Scatterplot: Colour Versus Previous Batch's Colour




```
dev.new(width=7, height=7)
plot(y=color,x=zlag(color,d=1),ylab='Color Property',
xlab='Previous Batch Color Property')
cor(y=color,x=zlag(color,d=1), use="pairwise")
```

Moderate correlation ($r \approx 0.5$).

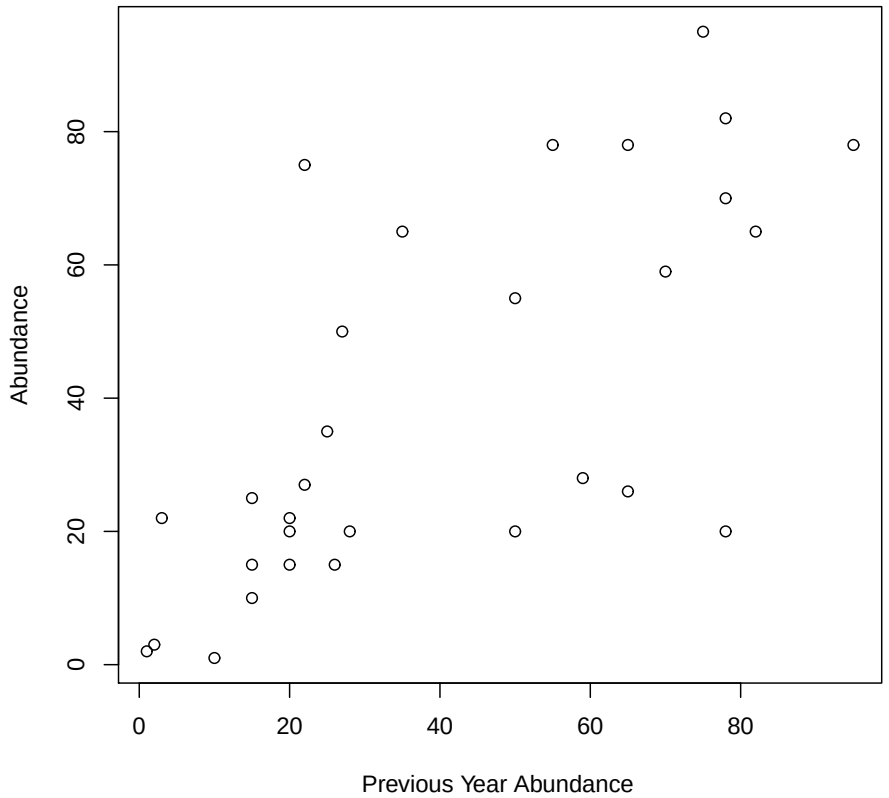
# Time Series Plot: Annual Abundance of Canadian Hare




```
dev.new(width=8, height=4)
data(hare)
plot(hare,ylab='Abundance',xlab='Year',type='o')
```

Neighbouring values are very closely related.

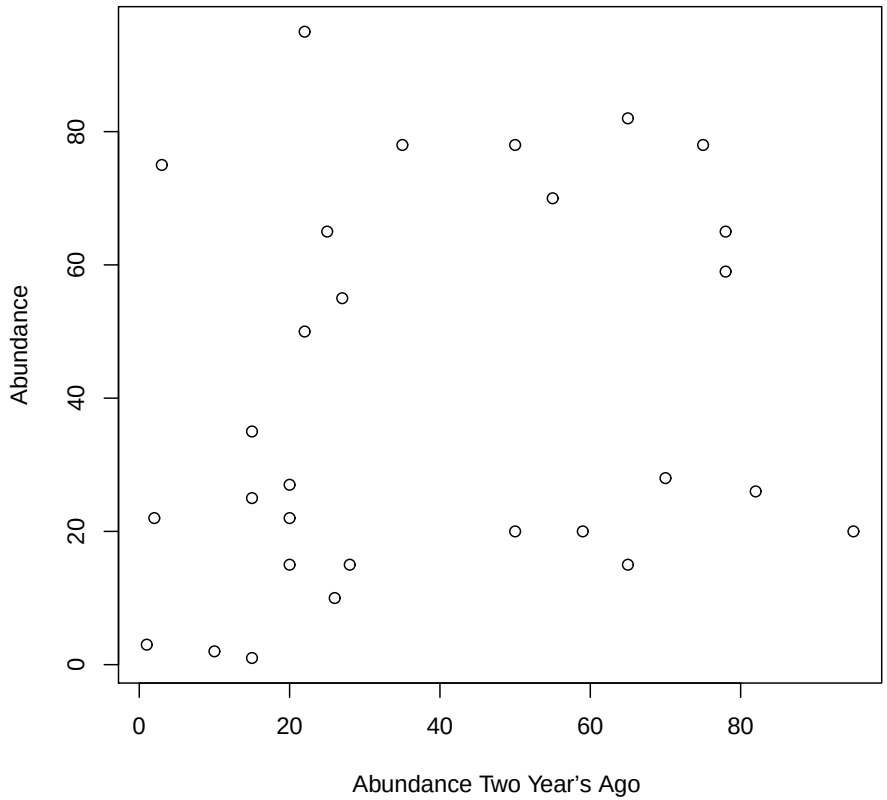
# Scatterplot: Abundance Versus Previous Year's Abundance




```
dev.new(width=7, height=7)
plot(y=hare,x=zlag(hare,d=1),ylab='Abundance',
xlab='Previous Year Abundance')
cor(y=hare,x=zlag(hare,d=1), use="pairwise")
```

Relatively strong correlation ($r \approx 0.7$).

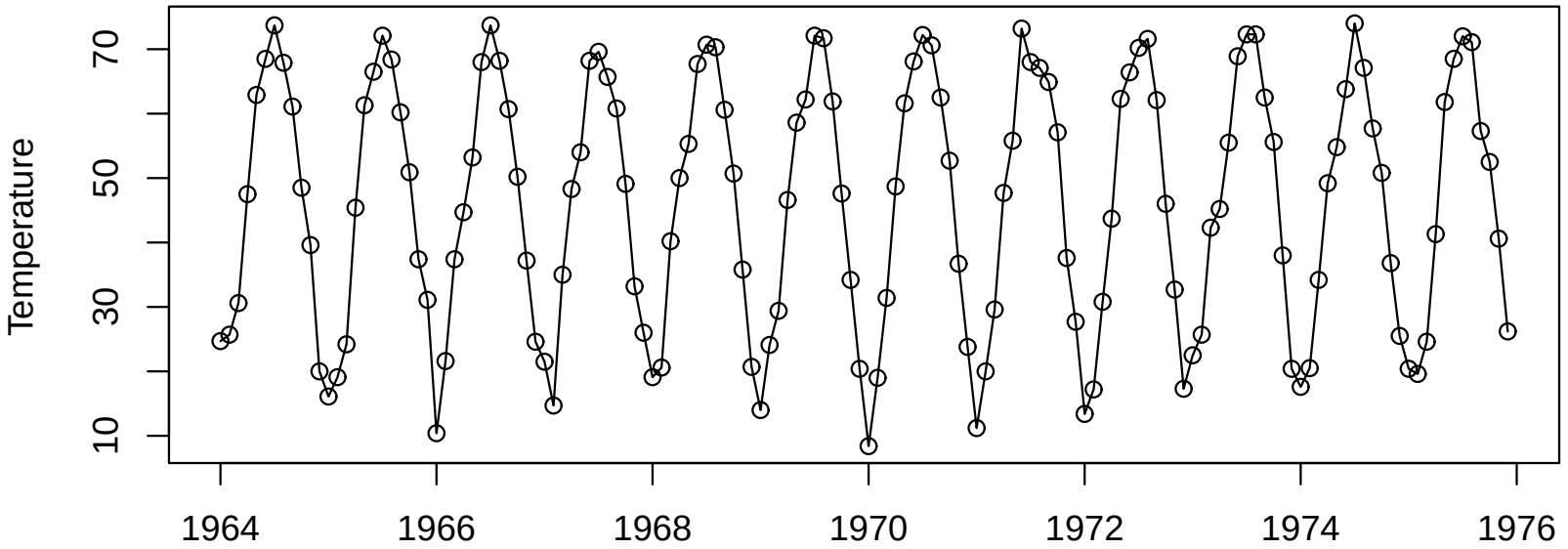
# Scatterplot: Abundance Versus Abundance Two Year's Ago




```
dev.new(width=7, height=7)
plot(y=hare,x=zlag(hare,d=2),ylab='Abundance',
xlab='Abundance Two Year's Ago')
cor(y=hare,x=zlag(hare,d=2), use="pairwise")
```

Less correlated at **lag two**, i.e., two years apart, ($r \approx 0.4$).

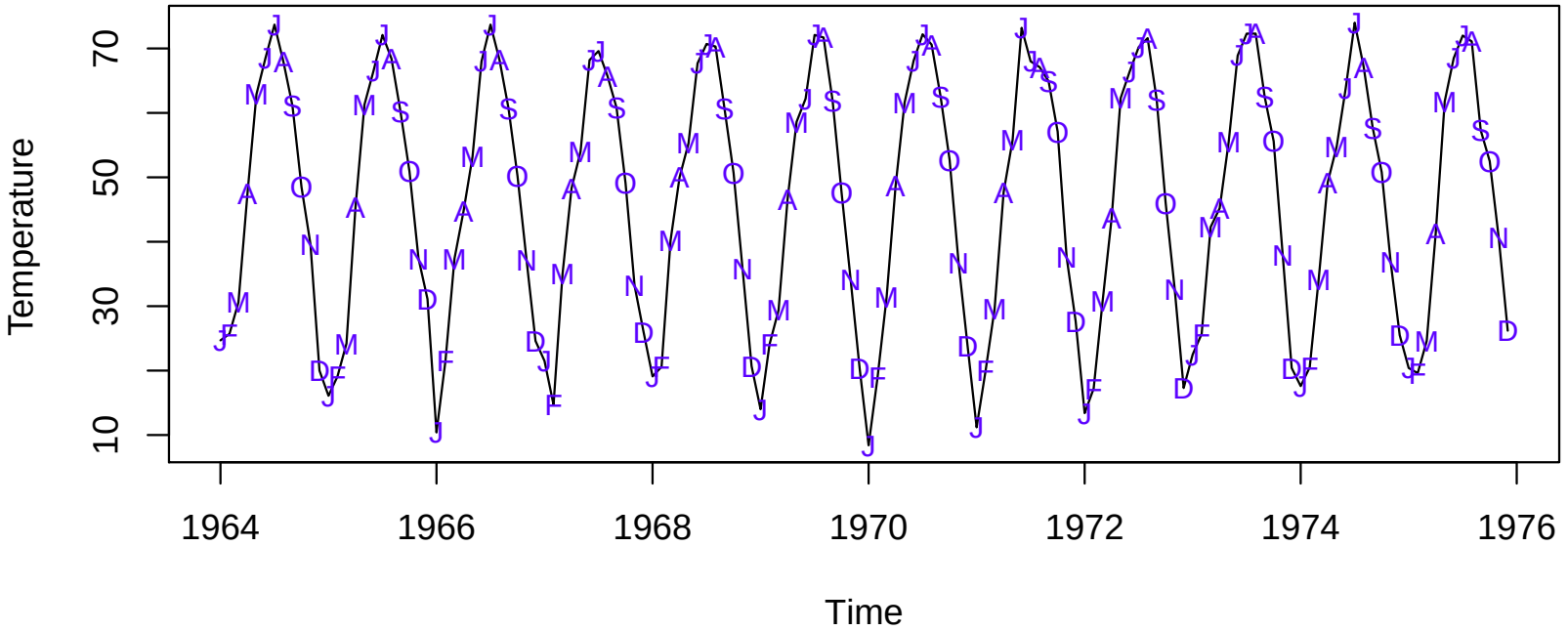
# Time Series Plot: Average Monthly Temperature in Dubuque




```
dev.new(width=8, height=4)
data(tempdub)
plot(tempdub,ylab='Temperature',type='o')
```

Very regular pattern called (monthly) **seasonality**: observations twelve months apart are related.

# Time Series Plot: With Month Labels




```
plot(tempdub,ylab='Temperature',type='l')
points(y=tempdub,x=time(tempdub),
pch=as.vector(season(tempdub)), col=4, cex=0.8 )
```

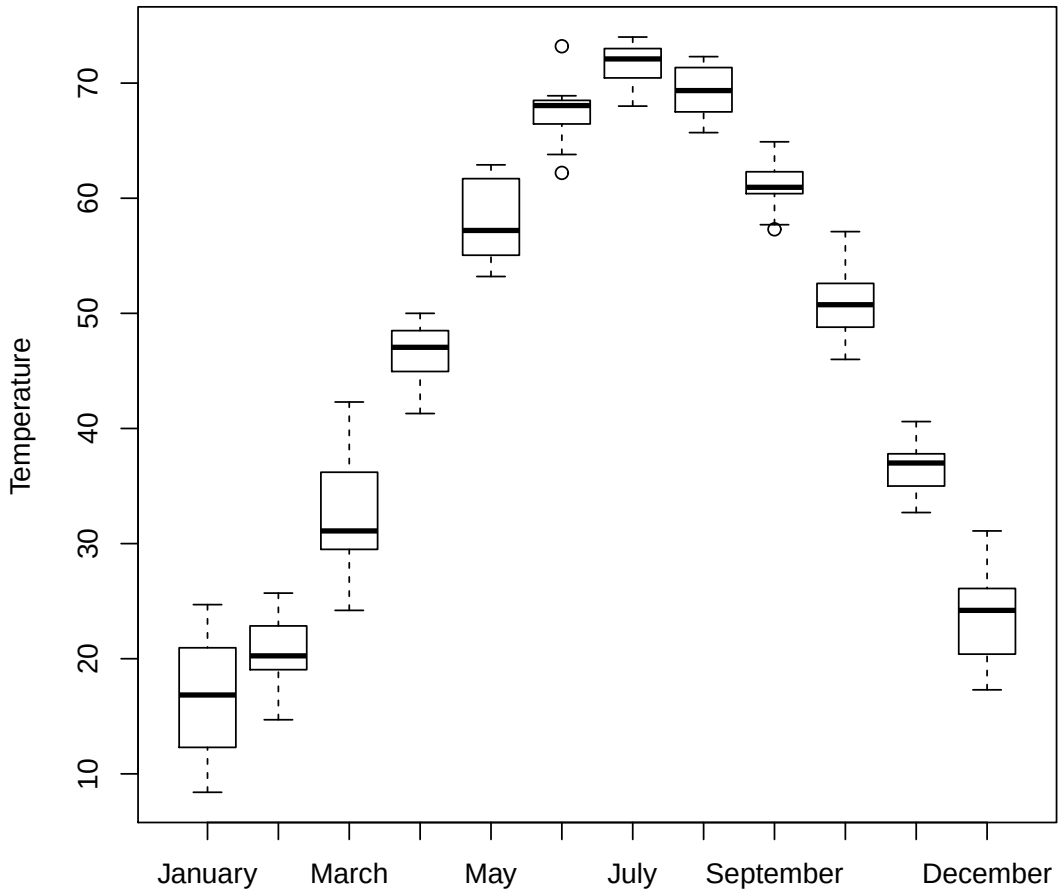
- `time(tempdub)` extracts the time points
- `as.vector(season(tempdub))` extracts the season labels, i.e., the months

# Monthly Means using aggregate

```
aggregate(tempdub~season(tempdub),
data=tempdub, mean)
```

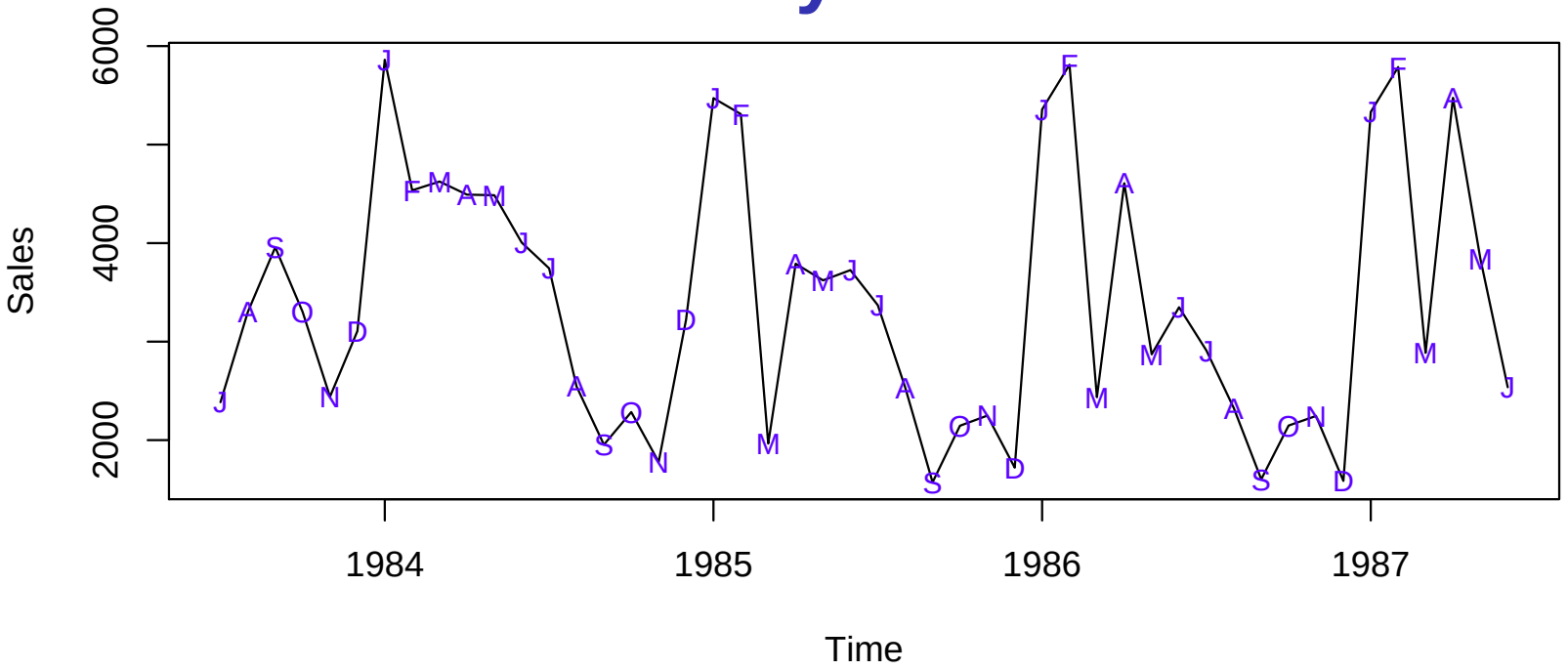
```
   season(tempdub)  tempdub
1          January 16.60833
2         February 20.65000
3            March 32.47500
4            April 46.52500
5              May 58.09167
6             June 67.50000
7             July 71.71667
8           August 69.33333
9        September 61.02500
10         October 50.97500
11        November 36.65000
12        December 23.64167
```

# Monthly Boxplots




```
dev.new(width=7, height=7)
boxplot(tempdub~season(tempdub), ylab="Temperature")
```

# Time Series Plot: Monthly Oil Filter Sales




```
dev.new(width=8, height=4)
data(oilfilters)
plot(oilfilters,ylab='Sales',type='l')
points(y=oilfilters,x=time(oilfilters),
pch=as.vector(season(oilfilters)), col=4, cex=0.8 )
```

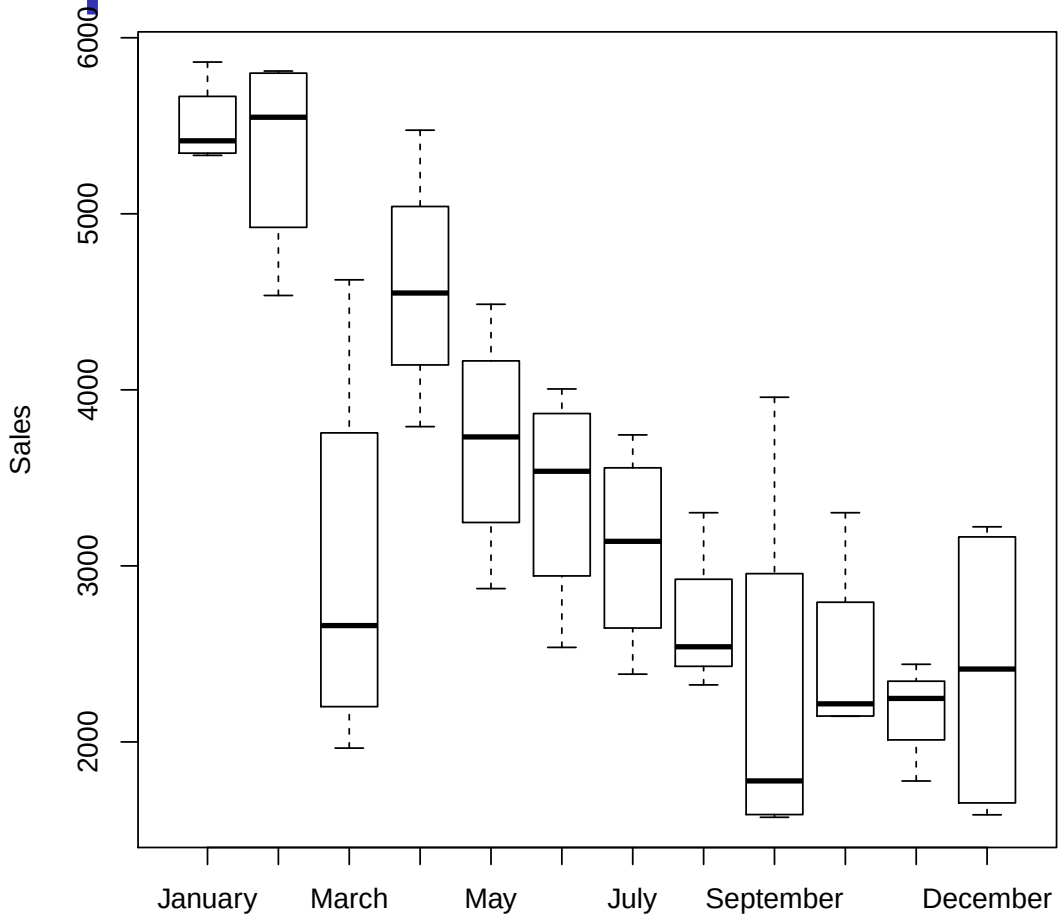
- January/February high, September – December low.

# Monthly Means using aggregate

```
aggregate(oilfilters~season(oilfilters),
data=oilfilters, mean)
```

```
   season(oilfilters) oilfilters
1             January    5505.75
2            February    5361.00
3               March    2978.00
4               April    4591.50
5                 May    3705.50
6                June    3404.25
7                July    3102.00
8              August    2676.75
9           September    2271.75
10            October    2470.25
11           November    2178.25
12           December    2409.00
```

# Monthly Boxplots




```
dev.new(width=7, height=7)
boxplot(oilfilters~season(oilfilters), ylab="Sales")
```

# Modelling Time Series Data

Box and Jenkins (1976) three stages of model building:

1. **Identification:** decide on initial models for the data based on plotting the time series, computing summary statistics, investigating the autocorrelation structure and applying knowledge of the subject area.
2. **Fitting:** estimate model parameters using maximum likelihood, least squares or method of moments.
3. **Checking:** assess how well the model fits the data by investigating residuals

If the model provides a good fit, then it is useful and can be used to predict future values of the time series. Otherwise, we must return to step 1. and consider other models.

# The Principle of Parsimony

The principle of parsimony is useful to bear in mind when building models.

1. Oxford Definition: “extreme unwillingness to use resources”
2. Occam’s Razor: “Entities must not be multiplied beyond necessity”
3. Albert Einstein: “everything should be made as simple as possible but not simpler”

In other words, aim to use the simplest model (smallest number of parameters) that provides a good fit.

Achieving a slightly better fit with many more parameters is generally not worthwhile. However, if a complex model is required, that is fine.

# Random Variables

- **Random variable**: $X$ is a numeric quantity whose value is determined by a probabilistic process.
- **Support**: $\mathcal{X}$ is the set of all possible values of $X$.
- **Observation**: $x$, i.e., a realisation of $X$.
- **Mass function** ($X$ is **discrete**): produces probabilities via $p(x) = \Pr(X = x)$ where $\sum_{x_i \in \mathcal{X}} p(x_i) = 1$.
- **Density function** ($X$ is **continuous**): a function $f(x)$ (*not* a probability) such that $\int_{\mathcal{X}} f(x)\,dx = 1$.

# Random Variables

- **Joint density**: $f(x,y)$ where $\int_{\mathcal{Y}} \int_{\mathcal{X}} f(x,y)\,dx\,dy = 1$
- **Independence**: $f(x,y) = f(x)f(y)$
- **Marginal density**: $f(x) = \int_{\mathcal{Y}} f(x,y)\,dy = 1$ and similarly $f(y) = \int_{\mathcal{X}} f(x,y)\,dy = 1$
- **Conditional density**: $f(x \mid y) = \frac{f(x,y)}{f(y)}$

# Expectation

- **Expected Value**: $E(X) = \int x\, f(x)\, dx$
- **Function of $X$**: $E[g(X)] = \int g(x)\, f(x)\, dx$
- **Linear property**: $E(aX + b) = aE(X) + b$
- **Function of $X$ and $Y$**: $E[g(X,Y)] = \int \int g(x,y)\, f(x,y)\, dx\, dy$
- **Linear property**: $E(aX + bY + c) = aE(X) + bE(Y) + c$ and, more generally,

$$E(a_1X_1 + \cdots + a_nX_n) = a_1E(X_1) + \cdots + a_nE(X_n)$$

$$E\left(\sum_{i=1}^{n} a_iX_i\right) = \sum_{i=1}^{n} a_iE(X_i)$$

# Variance

A *positive* measure of spread around the mean.

- **Variance**: $\text{Var}(X) = E[X - EX]^2 = E(X^2) - (EX)^2$
- **Linear function**: $\text{Var}(aX + b) = a^2 \text{Var}(X)$.

# Covariance

Measure (positive or negative) of linear relationship between $X$ and $Y$.

- $\text{Cov}(X, Y) = E[(X - EX)(Y - EY)] = E(XY) - (EX)(EY)$

  Note that $X, Y$ independent $\Rightarrow \text{Cov}(X, Y) = 0$.
  However $\text{Cov}(X, Y) = 0 \not\Rightarrow X, Y$ independent.

- $\text{Cov}(X, X) = \text{Var}(X)$

- $\text{Cov}(X, Y) = \text{Cov}(Y, X)$

- $\text{Cov}(aX + b, cY + d) = ac\,\text{Cov}(X, Y)$

# Covariance of Two Sums

We can show that

$$\begin{aligned}\text{Cov}(X_1 + X_2, Y_1 + Y_2) \\ = \text{Cov}(X_1, Y_1) + \text{Cov}(X_1, Y_2) + \text{Cov}(X_2, Y_1) + \text{Cov}(X_2, Y_2)\end{aligned}$$

and, more generally,

$$\text{Cov}\left(\sum_{i=1}^{n_1} X_i, \sum_{j=1}^{n_2} Y_j\right) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \text{Cov}\left(X_i, Y_j\right).$$

# Variance of a Sum

Note that, from the result on the previous slide, we can get:

$$\begin{aligned}
\text{Var}(X_1 + X_2) &= \text{Cov}(X_1 + X_2, X_1 + X_2) \\
&= \text{Cov}(X_1, X_1) + \text{Cov}(X_1, X_2) + \text{Cov}(X_2, X_1) + \text{Cov}(X_2, X_2) \\
&= \text{Var}(X_1) + \text{Cov}(X_1, X_2) + \text{Cov}(X_1, X_2) + \text{Var}(X_2) \\
&= \text{Var}(X_1) + \text{Var}(X_2) + 2\,\text{Cov}(X_1, X_2),
\end{aligned}$$

i.e., $\text{Var}(X_1 + X_2) \neq \text{Var}(X_1) + \text{Var}(X_2)$ in general.

# Variance of a Sum

More generally we have:

$$\text{Var}\left(\sum_{i=1}^{n} X_i\right) = \text{Cov}\left(\sum_{i=1}^{n} X_i, \sum_{j=1}^{n} X_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} \text{Cov}\left(X_i, X_j\right)$$

$$= \sum_{i=j} \text{Cov}\left(X_i, X_j\right) + \sum_{i>j} \text{Cov}\left(X_i, X_j\right) + \sum_{i<j} \text{Cov}\left(X_i, X_j\right)$$

$$= \sum_{i=j} \text{Cov}\left(X_i, X_j\right) + 2\sum_{i>j} \text{Cov}\left(X_i, X_j\right)$$

$$= \sum_{i=1}^{n} \text{Var}(X_i) + 2\sum_{i=2}^{n}\sum_{j=1}^{i-1} \text{Cov}\left(X_i, X_j\right).$$

# Correlation

Measure (positive or negative) of linear relationship between $X$ and $Y$.

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\,\text{Var}(Y)}}$$

scaled such that $\text{Corr}(X, Y) \in [-1, 1]$, i.e., $|\text{Corr}(X, Y)| \leq 1$.

Note: $|\text{Corr}(X, Y)| \leq 1$ is a result of the *Cauchy-Schwarz inequality*:

$$|\text{Cov}(X, Y)| \leq \sqrt{\text{Var}(X)\,\text{Var}(Y)}$$

or $$[\text{Cov}(X, Y)]^2 \leq \text{Var}(X)\,\text{Var}(Y)$$