Finding Data Patterns with R Markdown
Part 2

Professor Jian Zhang
SMSAS, University of Kent
E-mail: j.zhang-79@kent.ac.uk

1 Predictive data analysis

Data analysts typically use a technique called linear regression, which finds the line that best fits the data so we can make predictions based on that line. How could we make more accurate predictions by using other data? We could try to collect more data and incorporate that into our model, like considering the effect of overall economic growth on rising college tuition. More data and better techniques helps us to predict the future better, but nothing can guarantee a perfectly accurate prediction.

A linear regression is a statistical model that analyzes the relationship between a response variable (often called y) and one or more variables and their interactions (often called x or explanatory variables). You make this kind of relationship in your head all the time, for example, when you calculate the age of a child based on their height, you are assuming the older they are, the taller they will be. A linear regression can be calculated in R with the command lm.The lm command takes the variables in the format:

```
 lm([target] ~ [predictor], data = [data source])
```

We use this command to predict the Amr, Eur, Afr and Asi based on the value of Gml using the historical data. A good way to test the quality of the fit of the model is to look at the residuals or the differences between the real values and the predicted values. The idea here is that the sum of the residuals is approximately zero or as low as possible. One measure very used to test how good your model is is the coefficient of determination or R^2 defined by the proportion of the total variability explained by the regression model. The results show that R^2 for fitted regression lines are 0.8137, 0.9082, 0.4499, and 0.7558 respectively, indicating these fitting are of high goodness-of-fit quality.

```
library(tidyverse)
library(seasonal)
library(fpp2)
GM_life <- read_csv("GM-Life Expectancy.csv")
head(GM_life)
ts.plot(ts(GM_life[,3],start=1800,end=2024),xlab="year",ylab="Expectancy",
        main = "Life Expectancy")
Fertility <- read_csv("Total fertility rate_simplified.csv")
head(Fertility)
x<-matrix(0,dim(Fertility)[2],4)
Fertility1<-Fertility[,1:dim(Fertility)[2]]
x<-t(Fertility1[,1:dim(Fertility)[2]])
ts.plot(ts(cbind(x[,1],
        x[,2],
        x[,3],x[,4]),start=1800,end=2024),
        gpars = list(xlab = "Year",
                     ylab = "Fertility",
                     main = "Fertility in four regions",
                     lwd = rep(2,4),
                     lty = c(1:4),
                     col = c("darkred","darkblue","darkgreen","grey")
                     )
        )
legend("topright", c("Amr","Eur","Afr","Asi"), lwd = rep(2,4), lty = c(1:4), col = c("darkred"

# ggplot
# Convert the data into a data frame
dat<-data.frame("Amr"=x[1:225,1],"Eur"=x[1:225,2], "Afr"=x[1:225,3], "Asi"=x[1:225,4], "Gml"=G

ggplot(data=dat) +
  aes(x =Gml, y = Amr) +
  geom_point(colour = "#0c4c8a") +
  theme_minimal()+
  geom_smooth(method='lm')

ggplot(data=dat) +
  aes(x =Gml, y = Eur) +
  geom_point(colour = "#0c4c8a") +
  theme_minimal()+
  geom_smooth(method='lm')

ggplot(data=dat) +
  aes(x =Gml, y = Afr) +
  geom_point(colour = "red") +
  theme_minimal()+
 geom_smooth(method='lm')
```

```
#pairwise scatter plot
pairs(dat)

#Simple linear regression analysis using the command lm()


summary(lm(dat$Amr~dat$Gml))

summary(lm(dat$Asi~dat$Gml))
```