Regression (Part 1)

Forecasting: principles and practice (2nd edition)

book by Rob Hyndman and George Athanasopoulos
slides adopted from Peter Fuleky and updated by Joseph Alba

Simple linear regression

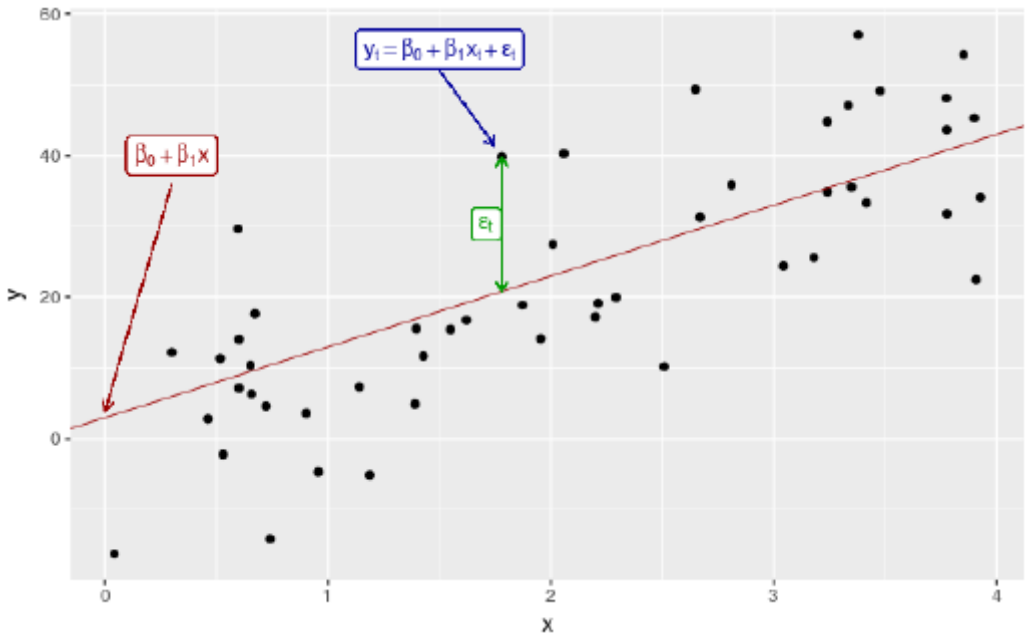
The basic concept is that we forecast variable y assuming it has a linear relationship with variable x.

$$y = \beta_0 + \beta_1 x + \varepsilon.$$

The model is called *simple* regression as we only allow one predictor variable x. The *forecast variable* y is sometimes also called the regressand, dependent or explained variable. The *predictor variable* x is sometimes also called the regressor, independent or explanatory variable.

The parameters β_0 and β_1 determine the intercept and the slope of the line respectively. The intercept β_0 represents the predicted value of y when $x = 0$. The slope β_1 represents the predicted increase in y resulting from a one unit increase in x.

Example



Linear regression

We can think of each observation y_t consisting of the systematic or explained part of the model, $\beta_0 + \beta_1 x_t$, and the random *error*, ε_t.

The error ε_t

captures anything that may affect y_t other than x_t. We assume that these errors:

- have mean zero; otherwise the forecasts will be systematically biased.
- are not autocorrelated; otherwise the forecasts will be inefficient as there is more information to be exploited in the data.
- are unrelated to the predictor variable; otherwise there would be more information that should be included in the systematic part of the model.

It is also useful to have the errors normally distributed with constant variance in order to produce prediction intervals and to perform statistical inference.

Another important assumption in the simple linear model is that x is not a random variable.

If we were performing a controlled experiment in a laboratory, we could control the values of x (so they would not be random) and observe the resulting values of y.

With observational data (including most data in business and economics) it is not possible to control the value of x, and hence we make this an assumption.

Least squares estimation

In practice, we have a collection of observations but we do not know the values of β_0 and β_1. These need to be estimated from the data. We call this *fitting a line through the data*.

There are many possible choices for β_0 and β_1, each choice giving a different line.

The least squares principle provides a way of choosing β_0 and β_1 effectively by minimizing the sum of the squared errors. The values of β_0 and β_1 are chosen so that that minimize

$$\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_t)^2.$$

Using mathematical calculus, it can be shown that the resulting **least squares estimators** are

$$\hat{\beta}_1 = \frac{\sum_{t=1}^{T}(y_t - \bar{y})(x_t - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2}$$

and

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where $\bar{x}$ is the average of the x observations and $\bar{y}$ is the average of the y observations. The estimated line is known as the *regression line*.

Fitted values and residuals

The forecast values of y obtained from the observed x values are called *fitted values*: $\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_t$, for $t = 1, \ldots, T$. Each $\hat{y}_t$ is the point on the regression line corresponding to x_t.

The difference between the observed y values and the corresponding fitted values are the *residuals*:

$$e_t = y_t - \hat{y}_t = y_t - \hat{\beta}_0 - \hat{\beta}_1 x_t.$$

The residuals have some useful properties including the following two:

$$\sum_{t=1}^{T} e_t = 0 \quad \text{and} \quad \sum_{t=1}^{T} x_t e_t = 0.$$

Example: US consumption expenditure

Figure below shows time series of quarterly percentage changes (growth rates) of real personal consumption expenditure (y) and real personal disposable income (x) for the US from 1970 Q1 to 2016Q3.

```
library(fpp2)
autoplot(uschange[,c("Consumption","Income")]) +
ylab("% change") + xlab("Year")
```


2.5
0.0
-2.5
% change
1970
1980
1990
2000
2010
Year
series
Consumption
Income

This equation is estimated in R using the tslm (linear model with time series component) function:

```
tslm(Consumption ~ Income, data=uschange)
```

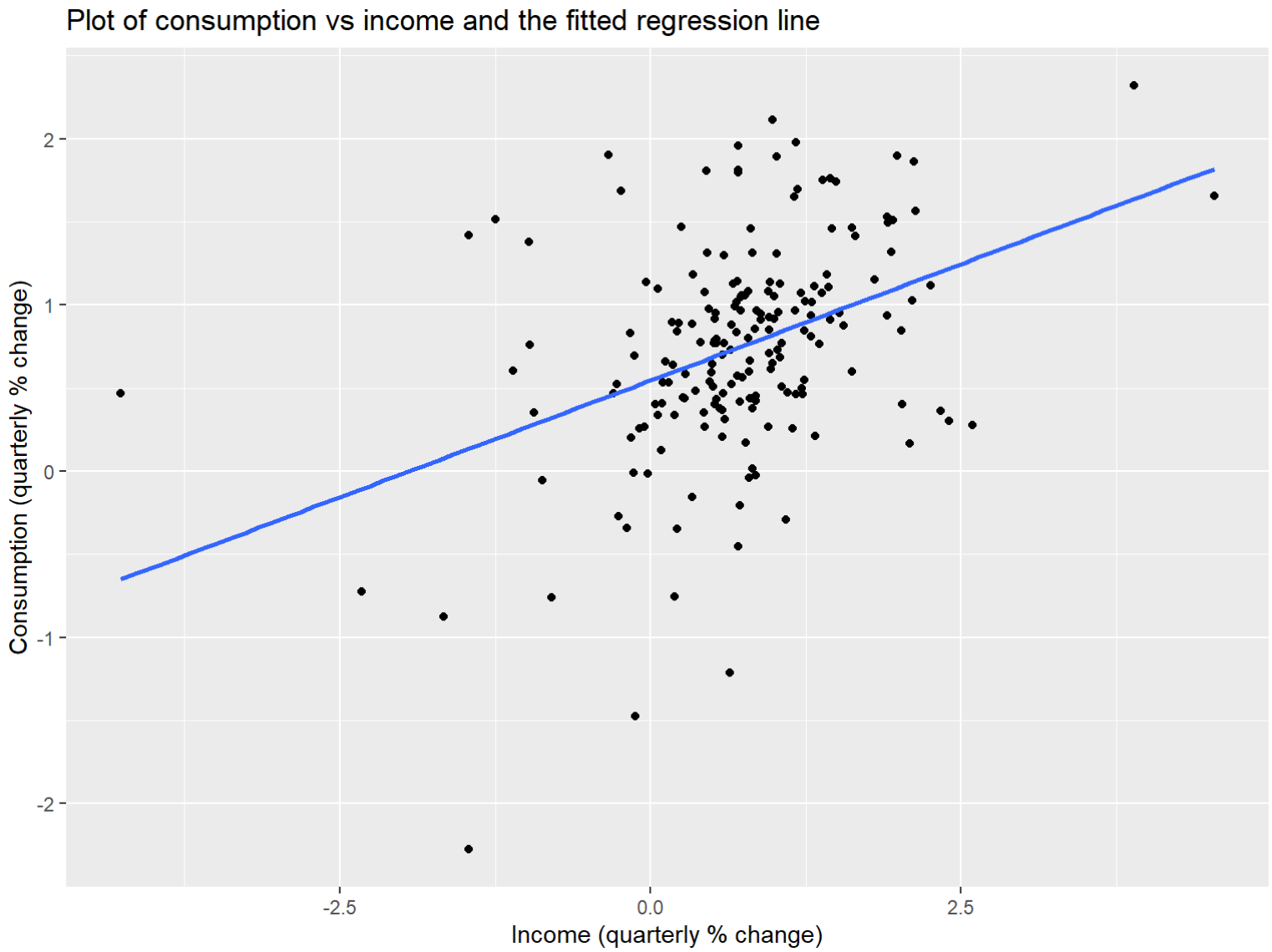
```
##
## Call:
## tslm(formula = Consumption ~ Income, data = uschange)
##
## Coefficients:
## (Intercept)       Income
##      0.5451       0.2806
```

Hence:

$$\hat{y} = 0.545 + 0.28x_t.$$

Graphically

```
uschange %>%
as.data.frame() %>%
ggplot(aes(x=Income, y=Consumption)) +
ggtitle("Plot of consumption vs income and the fitted regression line") +
ylab("Consumption (quarterly % change)") +
xlab("Income (quarterly % change)") +
geom_point() +
geom_smooth(method="lm", se=FALSE)
```


Plot of consumption vs income and the fitted regression line
Consumption (quarterly % change)
2
1
0
-1
-2
-2.5
0.0
2.5
Income (quarterly % change)

Multiple regression

In multiple regression there is one variable to be forecast and several predictor variables. The general form of a multiple regression is

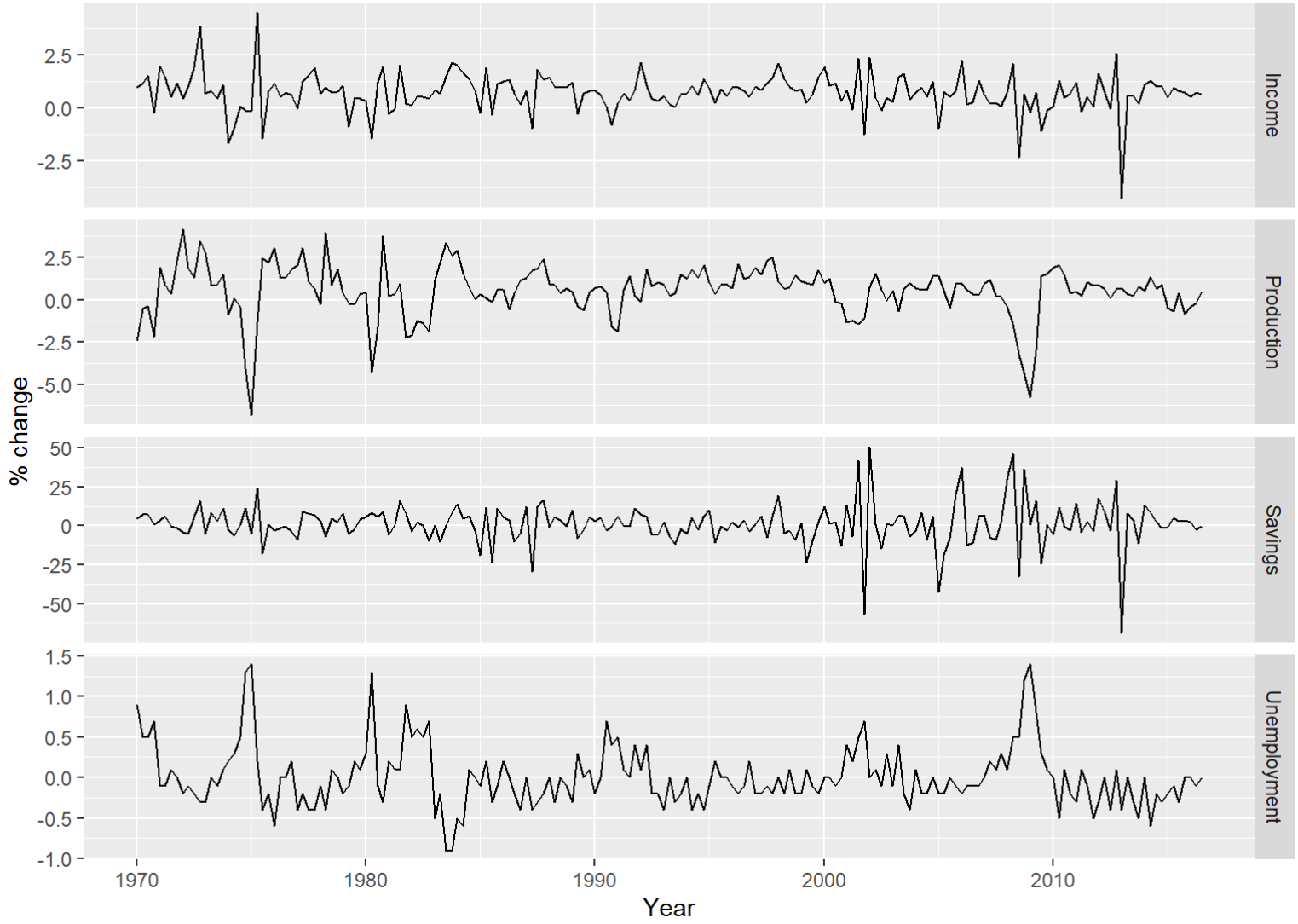
$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + e_t,$$

where y_t is the variable to be forecast and $x_{1,t}, \ldots, x_{k,t}$ are the k predictor variables.

The coefficients $\beta_1, \ldots, \beta_k$ measure the effect of each predictor after taking account of the effect of all other predictors in the model. Thus, the coefficients measure the *marginal effects* of the predictor variables.

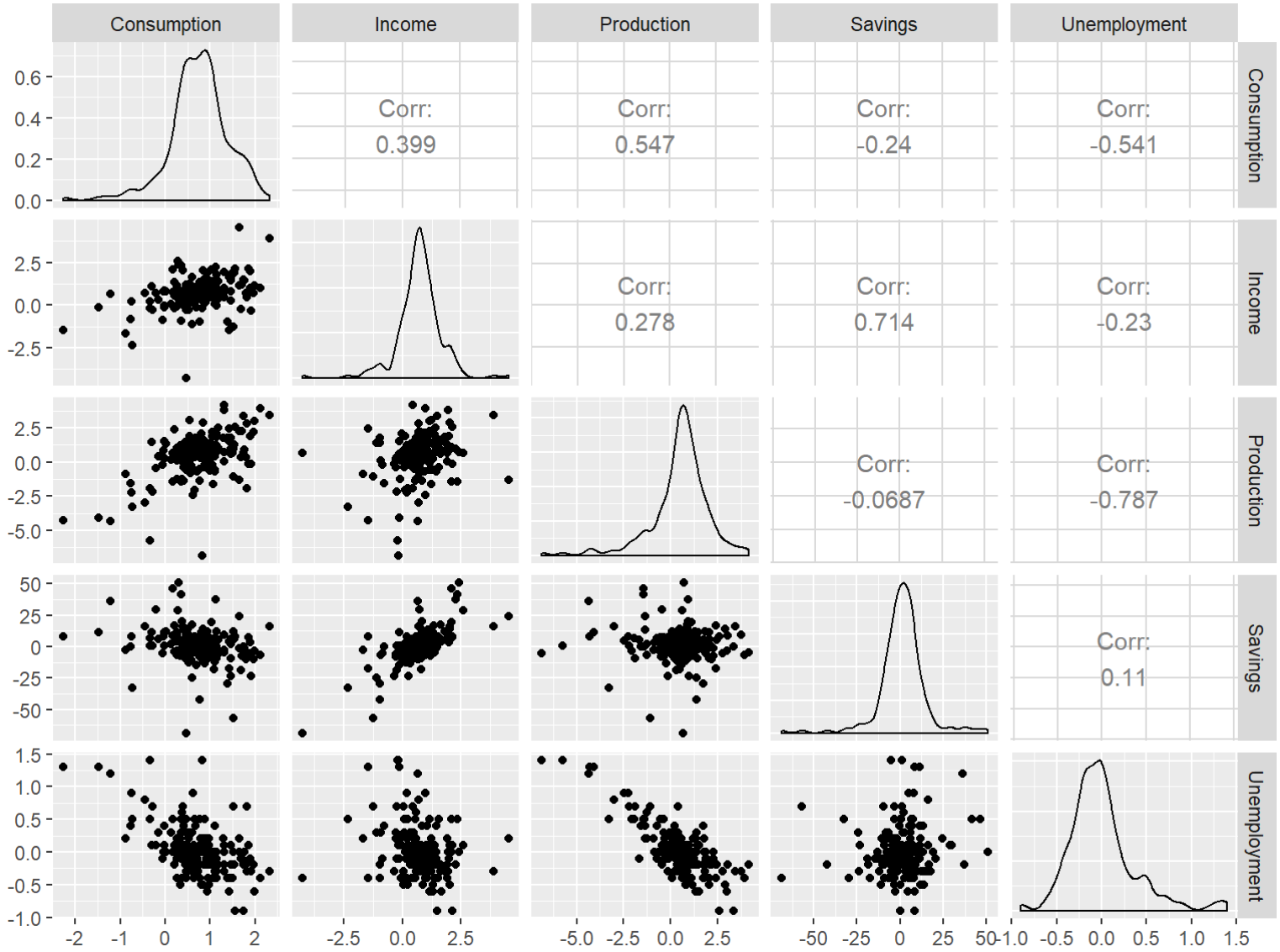
Example: US consumption expenditure

```
autoplot(uschange[,c(2,3,4,5)], facets = TRUE) +
ylab("% change") + xlab("Year")
```


Income
Production
Savings
Unemployment
% change
Year
2.5
0.0
-2.5
-5.0
50
25
0
-25
-50
1.5
1.0
0.5
-0.5
-1.0
1970
1980
1990
2000
2010

- The figure shows additional predictors that may be useful for forecasting US consumption expenditure.
- These variables are quarterly percentage changes in industrial production and personal savings, and quarterly changes . Building a multiple linear regression
- The model could potentially generate more accurate forecasts as we expect consumption expenditure to not only depend on personal income but on other predictors as well.
- We look at the correlation between consumption and each of the predictors to check the strength of these relationships

```
uschange %>%
as.data.frame() %>%
GGally::ggpairs()
```



Assumptions

The assumptions on errors of the simple linear regression must also hold with multiple regressions.

In addition, one or more of the predictor variables cannot be significantly correlated to each other. If these predictors are correlated, this is known as multicollinearity which may affect the accuracy of the forecast. However, multicollinearity could be addressed and would be less of a problem in forecasting than it would be when making statistical inferences (more on this later).

Least Squares Estimation

The values of the coefficients $\beta_0, \ldots, \beta_k$ are obtained by finding the minimum sum of squares of the errors. That is, we find the values of $\beta_0, \ldots, \beta_k$ which minimize

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \cdots - \beta_k x_{k,t})^2.$$

Finding the best estimates of the coefficients is often called “fitting” the model to the data, or sometimes “learning” or “training” the model. The line shown in figure above was obtained in this way.

Example: US consumption expenditure

The *tslm* function fits a linear regression model to time series data. It is very similar to the *lm* function which is widely used for linear models, but tslm provides additional facilities for handling time series.

```
fit.consMR <- tslm(Consumption ~ Income + Production +
  Unemployment + Savings, data=uschange)
summary(fit.consMR)
```

```
##
## Call:
## tslm(formula = Consumption ~ Income + Production + Unemployment +
##     Savings, data = uschange)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88296 -0.17638 -0.03679  0.15251  1.20553
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.26729    0.03721   7.184 1.68e-11 ***
## Income        0.71449    0.04219  16.934  < 2e-16 ***
## Production    0.04589    0.02588   1.773   0.0778 .
## Unemployment -0.20477    0.10550  -1.941   0.0538 .
## Savings      -0.04527    0.00278 -16.287  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3286 on 182 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7486
## F-statistic: 139.5 on 4 and 182 DF,  p-value: < 2.2e-16
```

The “t value” is the ratio of an estimated coefficient to its standard error and the last column gives the p-value: the probability of the estimated coefficient being as large as it is if there was no real relationship between the consumption and the corresponding predictor. This is useful when studying the effect of each predictor, but is not particularly useful when forecasting.

Fitted values

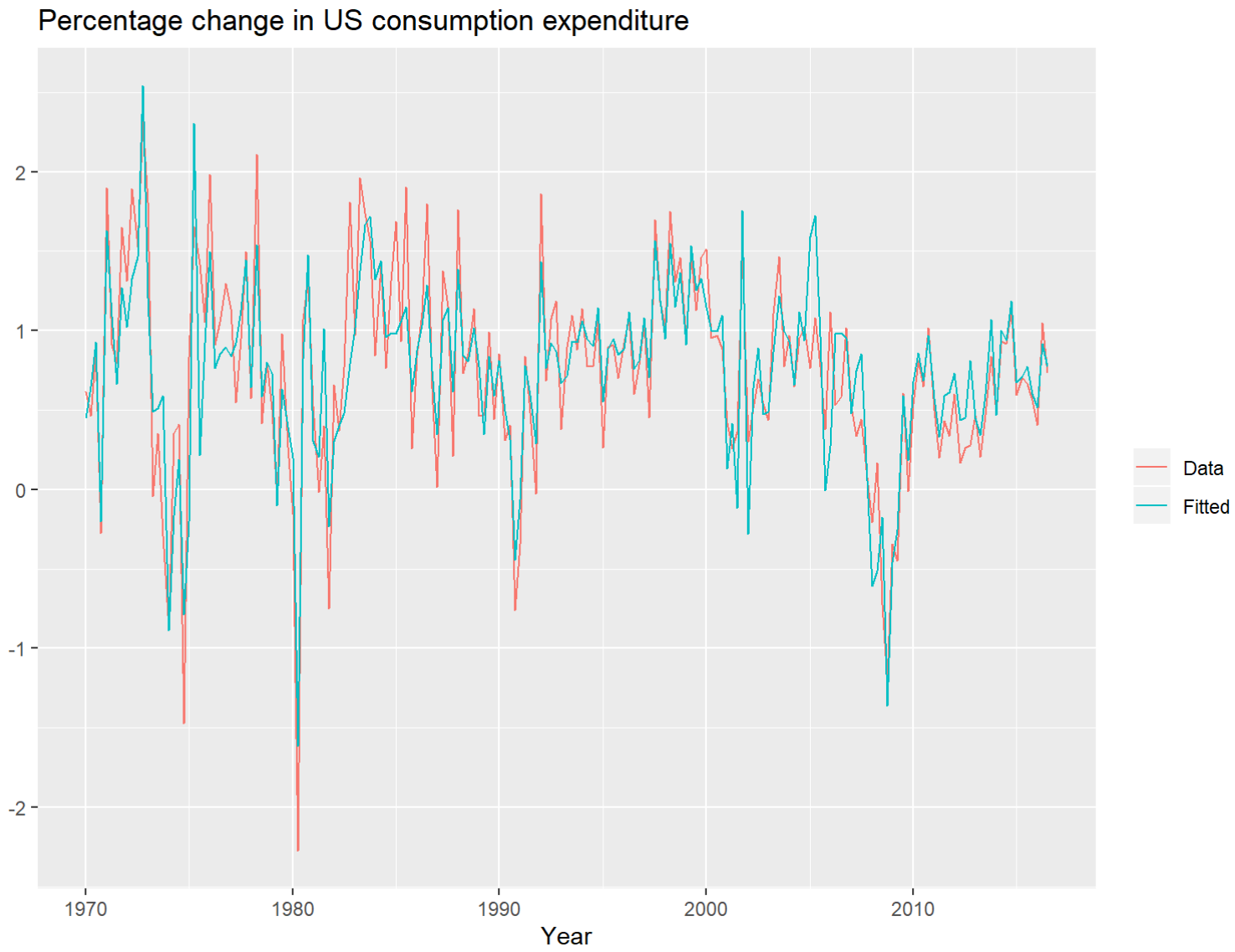
Predictions of y can be obtained by using the estimated coefficients in the regression equation and setting the error term to zero. In general we write,

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t},$$

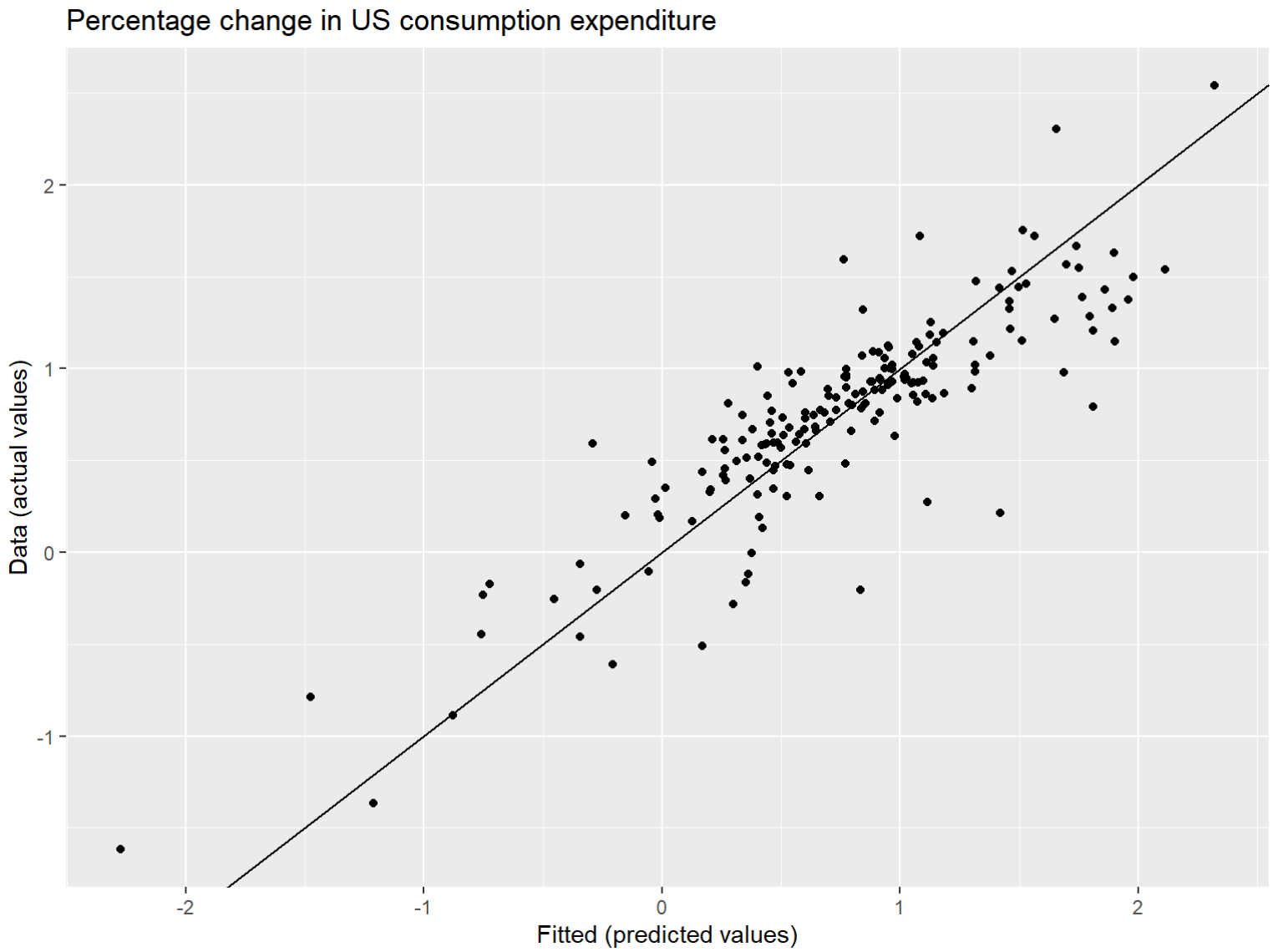
Plugging in the values of $x_{1,t}, \ldots, x_{k,t}$ for $t = 1, \ldots, T$ returns predictions of within the training sample, referred to as *fitted* values. Note that these are predictions of the data used to estimate the model not genuine forecasts of future values of y.

The following plots show the actual values compared to the fitted values for the percentage change in the US consumption expenditure series.

```
autoplot(uschange[,'Consumption'], series="Data") +
forecast::autolayer(fitted(fit.consMR), series="Fitted") +
xlab("Year") + ylab("") +
ggtitle("Percentage change in US consumption expenditure") +
guides(colour=guide_legend(title=" "))
```


Percentage change in US consumption expenditure
2
1
0
-1
-2
1970
1980
1990
2000
2010
Year
Data
Fitted

```
cbind(Data=uschange[,"Consumption"], Fitted=fitted(fit.consMR)) %>%
as.data.frame() %>%
ggplot(aes(x=Data, y=Fitted)) +
geom_point() +
xlab("Fitted (predicted values)") +
ylab("Data (actual values)") +
ggtitle("Percentage change in US consumption expenditure") +
geom_abline(intercept=0, slope=1)
```


Percentage change in US consumption expenditure
Data (actual values)
Fitted (predicted values)
2
1
0
-1
-2
-1
0
1
2

Some useful predictors

Trend

A linear trend can be modelled by simply using $x_{1,t} = t$.

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t.$$

where $t = 1, \dots, T$. A trend variable can be specified in the tslm function using the trend predictor.

Dummy Variables

A predictor could be a categorical variable taking only two values (e.g., “yes” and “no”).

This situation can still be handled within the framework of multiple regression models by creating a “dummy variable” taking value 1 corresponding to “yes” and 0 corresponding to “no”. A dummy variable is also known as an “indicator variable”.

If there are more than two categories, then the variable can be coded using several dummy variables (one fewer than the total number of categories).

Seasonal dummy variables

For example, suppose we are forecasting daily electricity demand and we want to account for the day of the week as a predictor. Then the following dummy variables can be created.

Notice that only six dummy variables are needed to code seven categories. That is because the seventh category (in this case Sunday) is specified when the dummy variables are all set to zero.

Many beginners will try to add a seventh dummy variable for the seventh category (*dummy variable trap*).

Day	D1	D2	D3	D4	D5	D6
Sunday	0	0	0	0	0	0
Monday	1	0	0	0	0	0
Tuesday	0	1	0	0	0	0
Wednesday	0	0	1	0	0	0
Thursday	0	0	0	1	0	0
Friday	0	0	0	0	1	0
Saturday	0	0	0	0	0	1
Sunday	0	0	0	0	0	0
.	.	.	.	.	.	.
.	.	.	.	.	.	.

The interpretation of each of the coefficients associated with the dummy variables is that it is a measure of the effect of that category relative to the omitted category. In the above example, the coefficient associated with Monday will measure the effect of Monday compared to Sunday on the forecast variable.

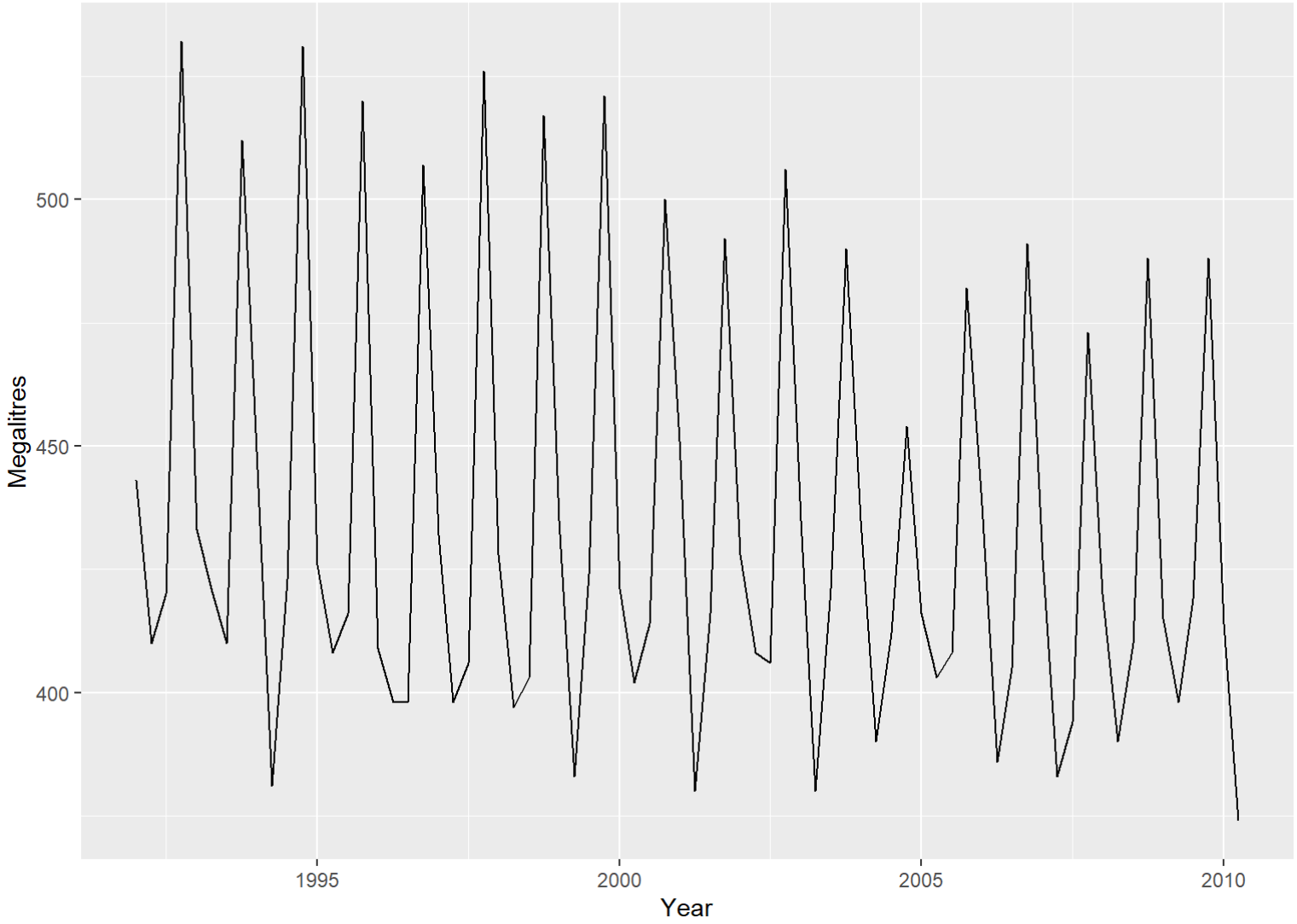
Other uses of dummy variables: *outliers*. If there is an outlier in the data, rather than omit it, you can use a dummy variable to remove its effect. In this case, the dummy variable takes value one for that observation and zero everywhere else.

The *tslm* function will automatically handle this situation if you specify the predictor *season*.

Example: Australian quarterly beer production

Recall the Australian quarterly beer production data shown

```
beer2 <- window(ausbeer, start=1992)
autoplot(beer2) + xlab("Year") + ylab("Megalitres")
```


Megalitres
500
450
400
1995
2000
2005
2010
Year

We want to forecast the value of future beer production. We can model this data using a regression model with a linear trend and quarterly dummy variables:

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + e_t,$$

where $d_{i,t} = 1$ if is in quarter and 0 otherwise. The first quarter variable has been omitted, so the coefficients associated with the other quarters are measures of the difference between those quarters and

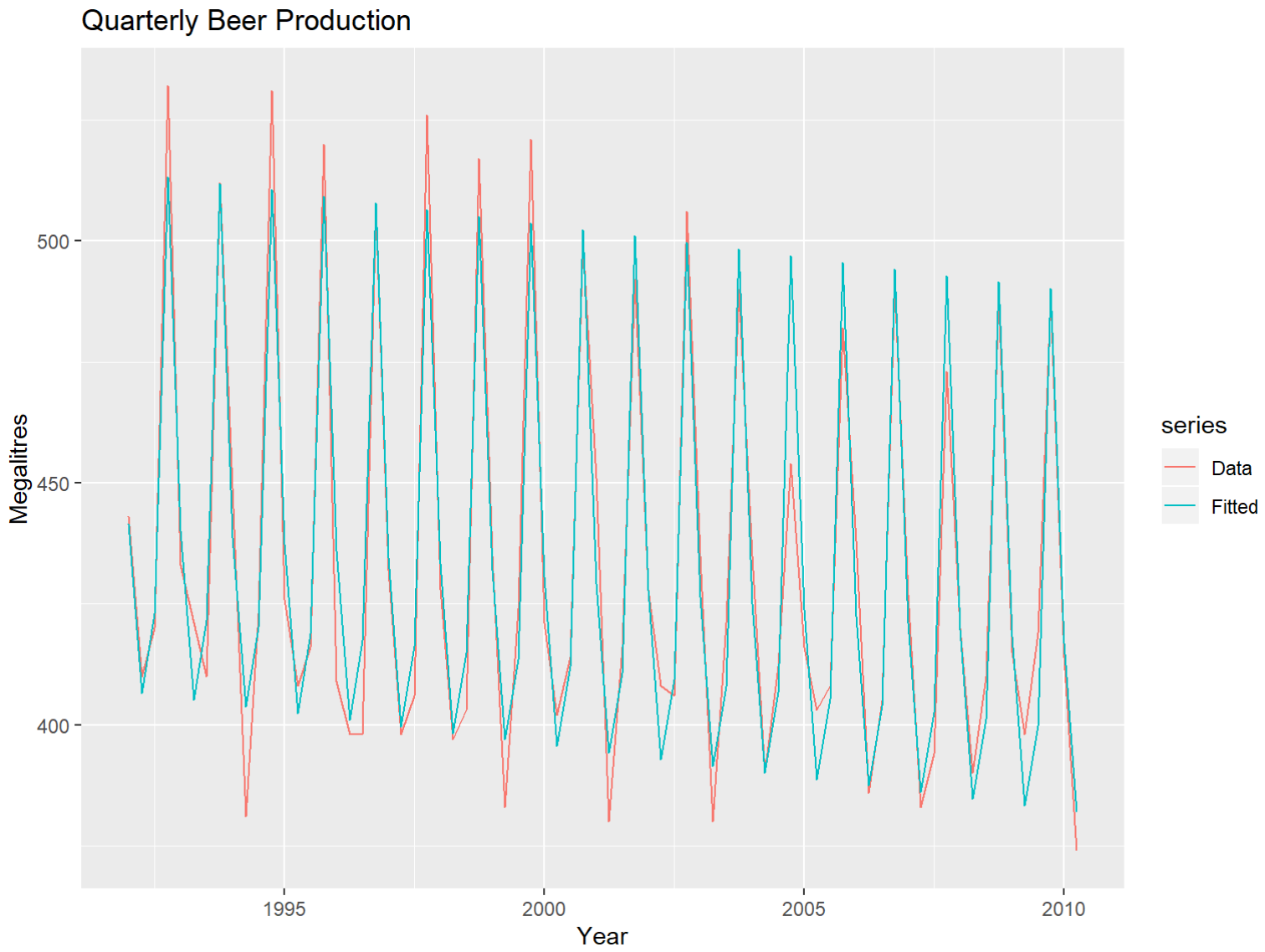
```
fit.beer <- tslm(beer2 ~ trend + season)
summary(fit.beer)
```

Note that trend and season are not objects in the R workspace; they are created automatically by tslm when specified in this way.

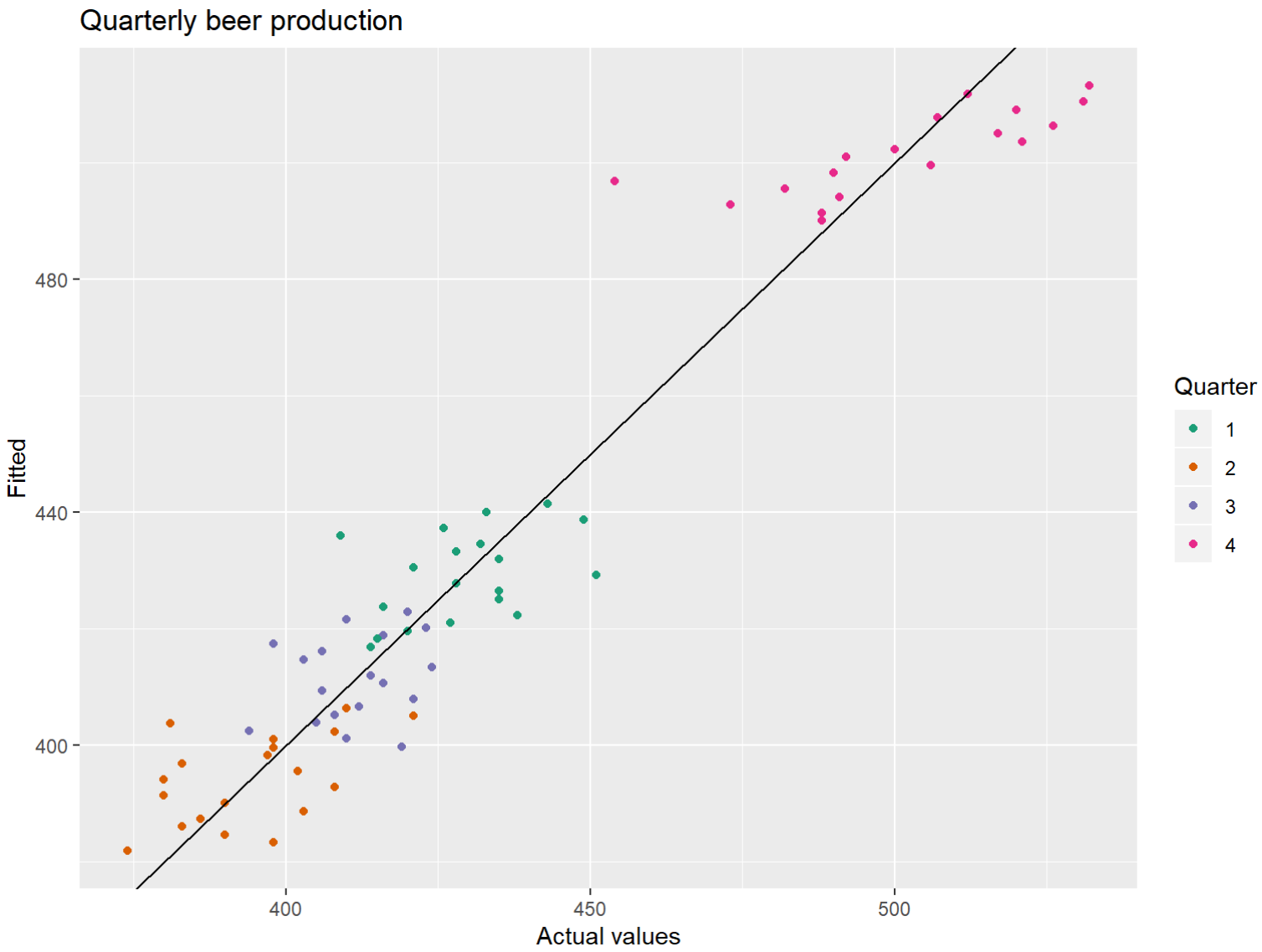
```
##
## Call:
## tslm(formula = beer2 ~ trend + season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.903  -7.599  -0.459   7.991  21.789
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 441.80044    3.73353 118.333  < 2e-16 ***
## trend        -0.34027    0.06657  -5.111 2.73e-06 ***
## season2     -34.65973    3.96832  -8.734 9.10e-13 ***
## season3     -17.82164    4.02249  -4.430 3.45e-05 ***
## season4      72.79641    4.02305  18.095  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9199
## F-statistic: 210.7 on 4 and 69 DF,  p-value: < 2.2e-16
```

There is a downward trend of -0.34 megalitres per quarter. On average, the second quarter has production of 34.7 megalitres lower than the first quarter, the third quarter has production of 17.8 megalitres lower than the first quarter, and the fourth quarter has production of 72.8 megalitres higher than the first quarter.

```
autoplot(beer2, series="Data") +
forecast::autolayer(fitted(fit.beer), series="Fitted") +
xlab("Year") + ylab("Megalitres") +
ggtitle("Quarterly Beer Production")
```


Quarterly Beer Production
Megalitres
500
450
400
1995
2000
2005
2010
Year
series
Data
Fitted

```
cbind(Data=beer2, Fitted=fitted(fit.beer)) %>%
as.data.frame() %>%
ggplot(aes(x=Data, y=Fitted, colour=as.factor(cycle(beer2)))) +
geom_point() +
ylab("Fitted") + xlab("Actual values") +
ggtitle("Quarterly beer production") +
scale_colour_brewer(palette="Dark2", name="Quarter") +
geom_abline(intercept=0, slope=1)
```


Quarterly beer production
Fitted
480
440
400
400
450
500
Actual values
Quarter
1
2
3
4

Fourier series

An alternative to using seasonal dummy variables, especially for long seasonal periods, is to use Fourier terms. Jean-Baptiste Fourier was a French mathematician, born in the 1700s, who showed that a series of sine and cosine terms of the right frequencies can approximate any periodic function. We can use them for seasonal patterns.

If m is the seasonal period, then the first few Fourier terms are given by

$$x_{1,t} = sin(\frac{2\pi t}{m}), x_{2,t} = cos(\frac{2\pi t}{m}), x_{3,t} = sin(\frac{4\pi t}{m}), x_{4,t} = cos(\frac{4\pi t}{m}), \ldots$$

There are many online references like https://www.mathsisfun.com/calculus/fourier-series.html

- If we have monthly seasonality, and we use the first 11 of these predictor variables, then we will get exactly the same forecasts as using 11 dummy variables.
- The advantage of using Fourier terms is that we can often use fewer predictor variables than we need to with dummy variables, especially when m is large. This makes them useful for weekly data, for example, where $m = 52$.
- For short seasonal periods such as with quarterly data, there is little advantage in using Fourier series over seasonal dummy variables.

In R, these Fourier terms are produced using the *fourier* function. For example, the Australian beer data can be modelled like this.

```
fourier.beer <- tslm(beer2 ~ trend + fourier(beer2, K=2))
summary(fourier.beer)
```

- The first argument to *fourier* just allows it to identify the seasonal period m and the length of the predictors to return.
- The second K argument specifies how many pairs of sin and cos terms to include.The maximum number allowed is $K = \frac{m}{2}$ where m is the seasonal period. Because we have used the maximum here, the results are identical to those obtained when using seasonal dummy variables.
- As we will learn in Section 9.5, K controls the smoothness of the seasonal pattern, ie. the higher the K the smoother the seasonal pattern …. (more of this in Section 9.5)

```
##
## Call:
## tslm(formula = beer2 ~ trend + fourier(beer2, K = 2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.903  -7.599  -0.459   7.991  21.789
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              446.87920    2.87321 155.533  < 2e-16 ***
## trend                     -0.34027    0.06657  -5.111 2.73e-06 ***
## fourier(beer2, K = 2)S1-4   8.91082    2.01125   4.430 3.45e-05 ***
## fourier(beer2, K = 2)C1-4  53.72807    2.01125  26.714  < 2e-16 ***
## fourier(beer2, K = 2)C2-4  13.98958    1.42256   9.834 9.26e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.23 on 69 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9199
## F-statistic: 210.7 on 4 and 69 DF,  p-value: < 2.2e-16
```

Intervention variables

It is often necessary to model interventions that may have affected the variable to be forecast.

Spike variable

For example, competitor activity, advertising expenditure, industrial action, and so on, can all have an effect. When the effect lasts only for one period, we use a spike variable. This is a dummy variable which takes value one in the period of the intervention and zero elsewhere. A spike variable is equivalent to a dummy variable for handling an outlier.

Other interventions have an immediate and permanent effect.

Step Variable

If an intervention causes a level shift (i.e., the value of the series changes suddenly and permanently from the time of intervention), then we use a step variable. A step variable takes value zero before the intervention and one from the time of intervention onward.

Piecewise linear trend

Another form of permanent effect is a change of slope. Here the intervention is handled using a piecewise linear trend; a trend that bends at the time of intervention and hence is nonlinear. We will discuss this in Section 5.8.

Trading days

The number of trading days in a month can vary considerably and can have a substantial effect on sales data. To allow for this, the number of trading days in each month can be included as a predictor.

For monthly or quarterly data, the *bizdays* function will compute the number of trading days in each period.

(Note: bizdays is not yet installed. If you wish to use it, you need to install the package)

An alternative that allows for the effects of different days of the week has the following predictors:

$$
\begin{aligned}
x_1 &= \text{number of Mondays in month;} \\
x_2 &= \text{number of Tuesdays in month;} \\
&\vdots \\
x_7 &= \text{number of Sundays in month.}
\end{aligned}
$$

Distributed lags

It is often useful to include advertising expenditure as a predictor. However, since the effect of advertising can last beyond the actual campaign, we need to include lagged values of advertising expenditure (to be discussed with ARIMA models). The following predictors may be use

$$
\begin{aligned}
x_1 &= \text{advertising for previous month;} \\
x_2 &= \text{advertising for two months previously;} \\
&\vdots \\
x_m &= \text{advertising for } m \text{ months previously.}
\end{aligned}
$$

Easter

Easter differs from most holidays because it is not held on the same date each year, and its effect can last for several days. In this case, a dummy variable can be used with value one where the holiday falls in the particular time period and zero otherwise.

For example, with monthly data, when Easter falls in March then the dummy variable takes value 1 in March, when it falls in April, the dummy variable takes value 1 in April, and when it starts in March and finishes in April, the dummy variable is split proportionally between months.

The *easter* function will compute the dummy variable for you.

```
easter.beer <- tslm(beer2 ~ trend + easter(beer2))
summary(easter.beer)
```

Evaluating the regression model

As mentioned, residuals have the useful properties that its average is zero and that the correlation between residuals and the observations for the predictor variable is also zero.

$$\sum_{t=1}^{T} e_t = 0 \quad \text{and} \quad \sum_{t=1}^{T} x_t e_t = 0.$$

After selecting the regression variables and fitting a regression model, it is necessary to plot the residuals to check that the assumptions of the model have been satisfied. There are a series of plots that should be produced in order to check different aspects of the fitted model and the underlying assumptions.

ACF plot of residuals

With time series data, it is usual to find *autocorrelation*. Hence, autocorrelation is also commonly found in the residuals of a fitted regression model with time series data. The estimated model would then violate the assumption of no autocorrelation in errors and the forecast may be inefficient - some other information may be included in the model to obtain a better forecast - but the forecast from the model is still unbiased. However, the prediction intervals will usually be larger. Hence ACF plots of residuals have to be checked.

To test for autocorrelation in residuals of regression models, the *Breusch-Godfrey* test, also referred to as the *LM* (Lagrange Multiplier) test for serial correlation, is used.

Histogram of residuals

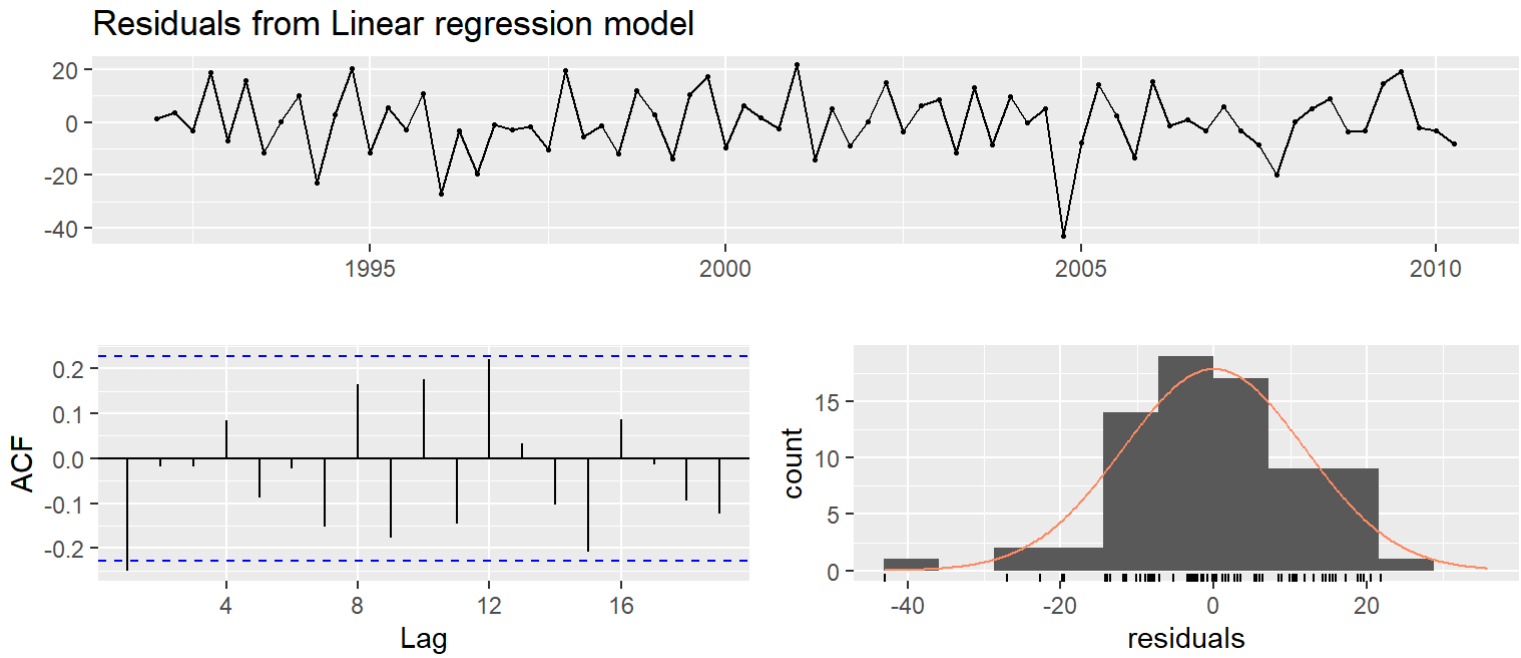
As in time series, we can also check if the residuals from the fitted regression model are normally distributed as this makes the calculation of the prediction intervals easier.

Similarly the function *checkresiduals* will give useful residual diagnostics and will show a time plot, the ACF and the histogram of the residuals from the model fitted to the beer production data as well as the Breusch-Godfrey test for testing up to 8th order autocorrelation.

Example

```
checkresiduals(fit.beer)
```

The plots in the next slide show residuals to be randomly scattered and do not have any systematic pattern. The Breusch-Godfrey test not rejecting the null of no autocorrelation at the 10% level of significance. However, the residual plots also show non-normal distribution.



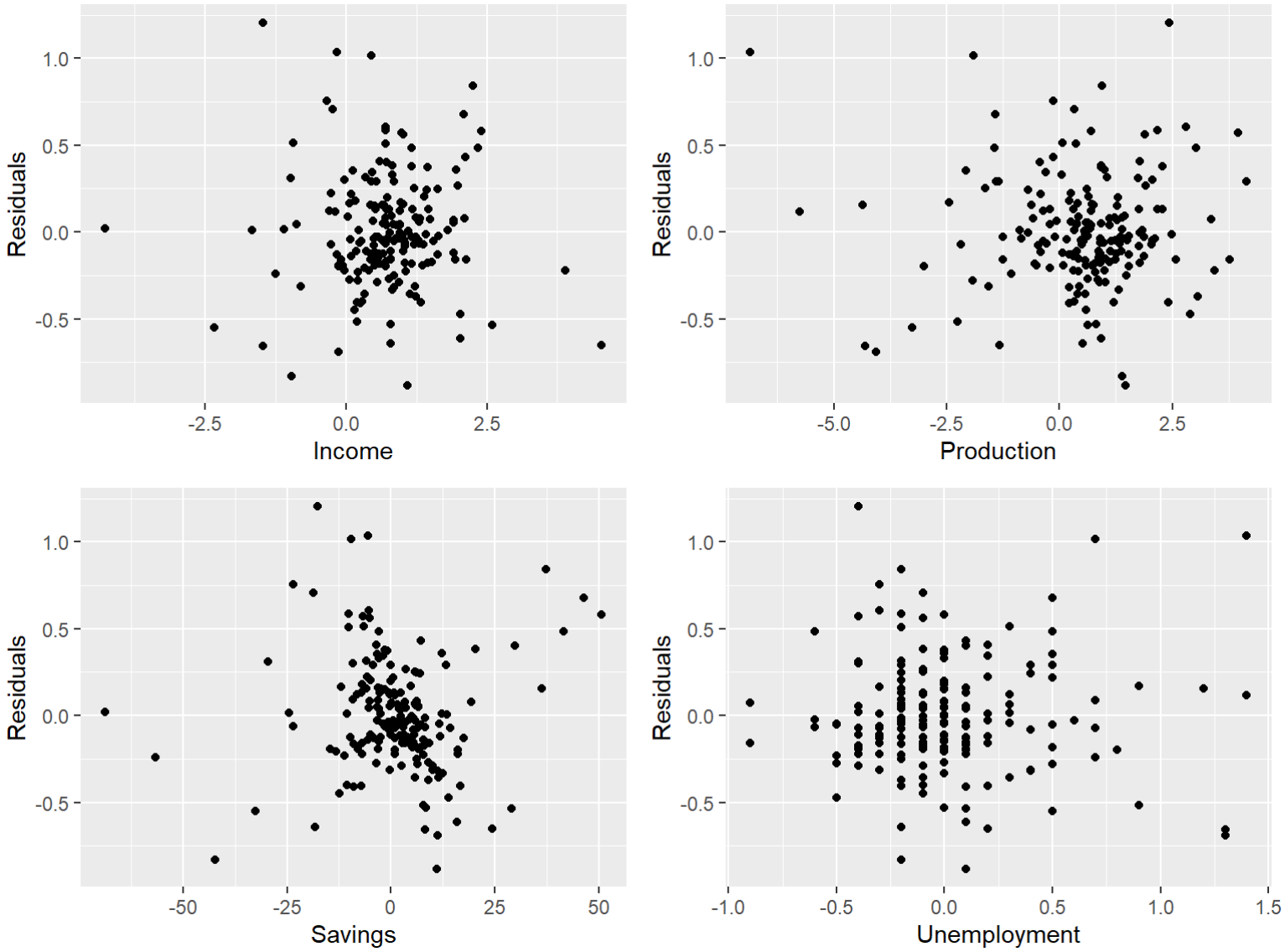

```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 9.3083, df = 8, p-value = 0.317
```

Residual plots against predictors

To plot the residuals from the the multiple regression model for forecasting US consumption plotted against each predictor, we use

```
df <- as.data.frame(uschange)
df[,"Residuals"] <- as.numeric(residuals(fit.consMR))
p1 <- ggplot(df, aes(x=Income, y=Residuals)) + geom_point()
p2 <- ggplot(df, aes(x=Production, y=Residuals)) + geom_point()
p3 <- ggplot(df, aes(x=Savings, y=Residuals)) + geom_point()
p4 <- ggplot(df, aes(x=Unemployment, y=Residuals)) + geom_point()
gridExtra::grid.arrange(p1, p2, p3, p4, nrow=2)
```

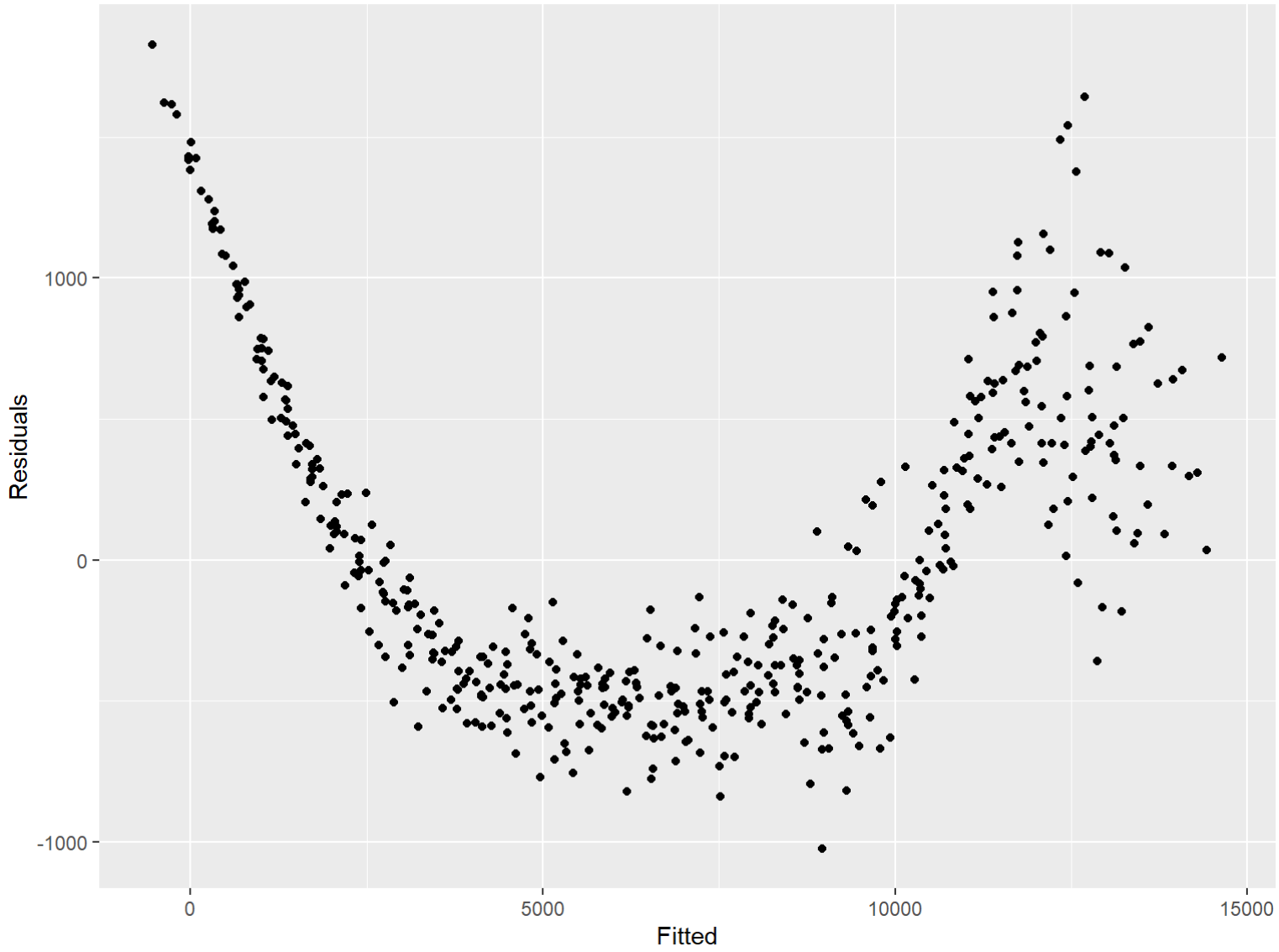
The plots in the next slide seem to be randomly scattered without systematic patterns.


Residuals
1.0
0.5
0.0
-0.5
-2.5
0.0
2.5
Income
Residuals
1.0
0.5
0.0
-0.5
-5.0
-2.5
0.0
2.5
Production
Residuals
1.0
0.5
0.0
-0.5
-50
-25
0
25
50
Savings
Residuals
1.0
0.5
0.0
-0.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Unemployment

Residual plots against fitted values

```
fit <- tslm(elec ~ trend + season)
cbind(Fitted=fitted(fit), Residuals=residuals(fit)) %>%
as.data.frame() %>%
ggplot(aes(x=Fitted, y=Residuals)) + geom_point()
```

The plot shows nonlinear trend and heteroscedastic pattern with variation increasing along the x-axis (Note: heteroscedasticity refers to non-constant variance)


Residuals
1000
0
-1000
0
5000
10000
15000
Fitted

Outliers and influential observations

Observations that take extreme values compared to the majority of the data are called “outliers”. Observations that have a large influence on the estimation results of a regression model are called “influential observations”. Usually, influential observations are also outliers that are extreme in direction.

Two of the many possible reasons for outliers are:

1. the observations wrongly recorded; these must be removed;

2. the observations are simply different, in which case, the observations must be analyzed and the reasons for it must be studied i.e., extraordinary events draughts, Olympics etc. It may not be a good idea to remove the observation

If outliers are to be removed, it is wise to report results with and without the observations.

Example

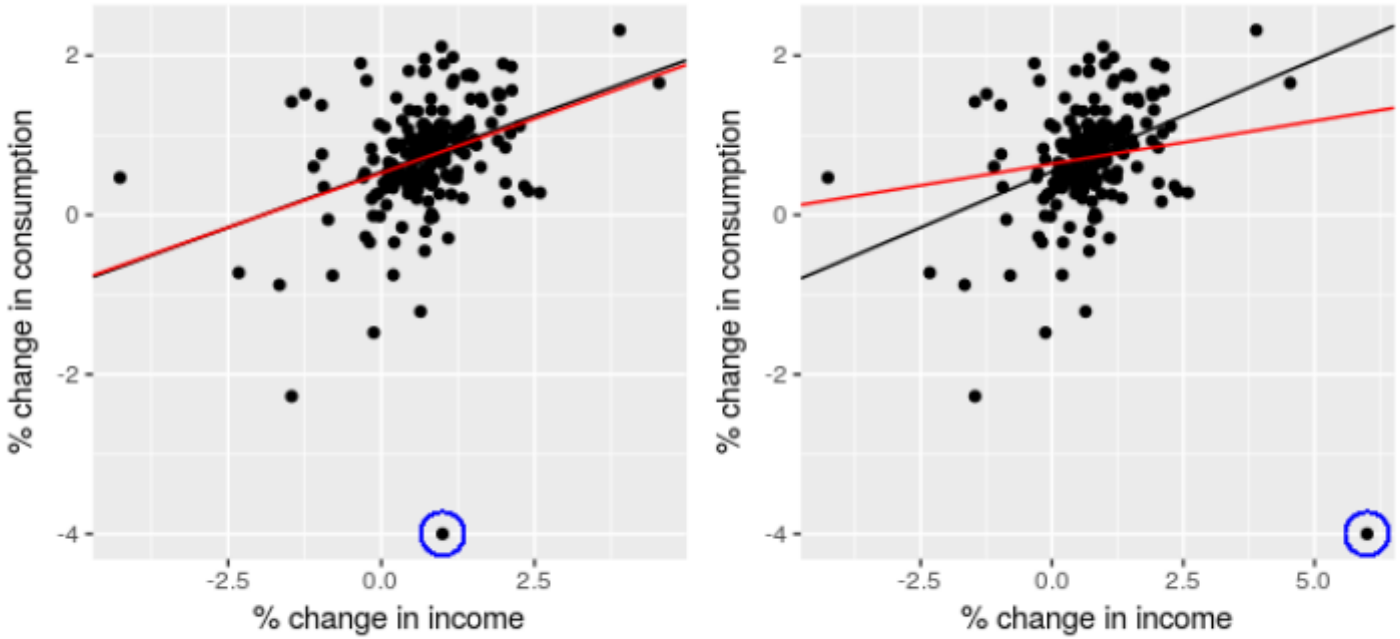


Figure 5.14: The effect of outliers and influential observations on regression

The effect of outliers on a regression.

In the left panel the outlier is only extreme in the direction of y, as the percentage change in consumption has been incorrectly recorded as -4%. The red line is the regression line fitted to the data which includes the outlier, compared to the black line which is the line fitted to the data without the outlier.

In the right panel the outlier now is also extreme in the direction of x with the 4% decrease in consumption corresponding to a 6% increase in income. In this case the outlier is very influential as the red line now deviates substantially from the black line.

Goodness-of-fit

The common measure of goodness of fit is the coefficient of determination or R^2 or the correlation between observed y values and the predicted $\hat{y}$ values:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2};\ 0 \leqslant R^2 \leqslant 1$$

R^2 value usually increases when adding any extra predictor to the model and this could lead to over-fitting. It also does not adjust for *degrees of freedom*.

Adjusted R^2

$$\bar{R}^2 = 1 - (1 - R^2)\frac{T-1}{T-k-1},$$

where T is the number of observations and k is the number of predictors. This is an improvement on R^2 as it will no longer increase with each added predictor.

Example

```
fit.consMR <- tslm(Consumption ~ Income + Production + Unemployment + Savings,
data=uschange)
summary(fit.consMR)
```

```
##
## Call:
## tslm(formula = Consumption ~ Income + Production + Unemployment +
##     Savings, data = uschange)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88296 -0.17638 -0.03679  0.15251  1.20553
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.26729    0.03721   7.184 1.68e-11 ***
## Income        0.71449    0.04219  16.934  < 2e-16 ***
## Production    0.04589    0.02588   1.773   0.0778 .
## Unemployment -0.20477    0.10550  -1.941   0.0538 .
## Savings      -0.04527    0.00278 -16.287  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3286 on 182 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7486
## F-statistic: 139.5 on 4 and 182 DF,  p-value: < 2.2e-16
```

$R^2 = 0.754$ and $\bar{R}^2 = 0.7486$ show the model does a good job explaining most of the variation in the consumption data.

Standard error of the regression

Another measure shown above is standard deviation of the residuals or the "residual standard error" which is another measure of how well the model has fitted the data. It is calculated as:

$$\hat{\sigma} = \sqrt{\frac{1}{T-k-1}\sum_{t=1}^{T} e_t^2}.$$

k refer to the number of predictors and "1" refers to the the intercept. As in the $\bar{R}^2$, it accounts for the number of estimated parameters in computing the residuals.

The standard error is related to the size of the average error that the model produces. We can compare this error to the sample mean of y or with the standard deviation of y to gain some perspective on the accuracy of the model. The evaluation of the standard error can be highly subjective as it is scale dependent (note: refer to R^2).

It is required when generating prediction intervals, discussed in Section 5.6

Spurious regression

Time series data are often "non-stationary," which means the values of the time series do not fluctuate around a constant mean or with a constant variance (Chapter 8)

Regressing non-stationary time series can lead to spurious regressions.

Example

Consider air passengers in Australia regressed against rice production in Guinea as shown in the next slide. Obviously, these are unrelated but appear to be so simply because they both trend upwards in the same manner.

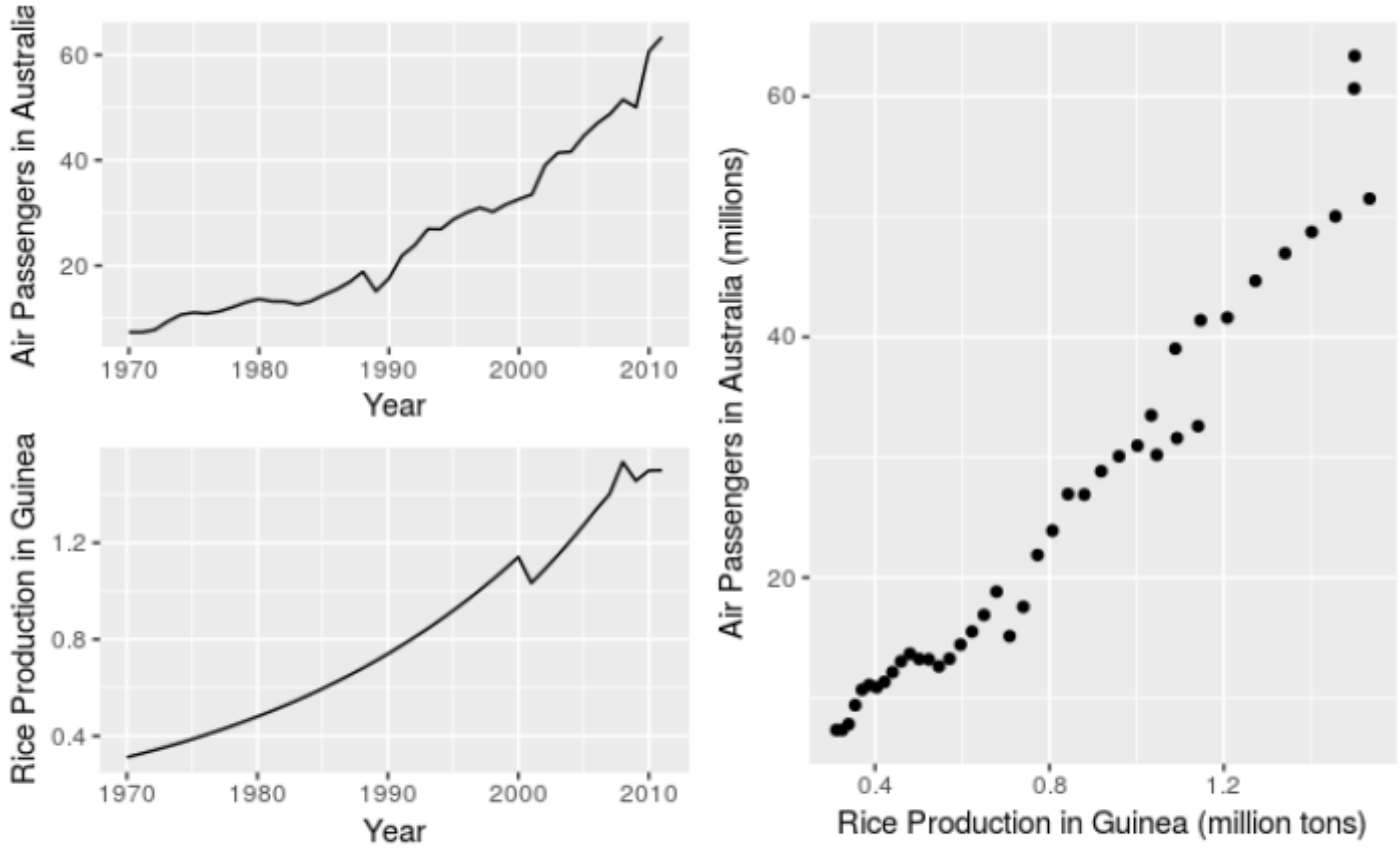


Figure 5.15: Trending time series data can appear to be related, as shown in this example where air passengers in Australia are regressed against rice production in Guinea.

Air passengers in Australia are regressed against rice production in Guinea.

```r
aussies <- window(ausair, end=2011)
fit <- tslm(aussies ~ guinearice)
summary(fit)
```

```
##
## Call:
## tslm(formula = aussies ~ guinearice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9448 -1.8917 -0.3272  1.8620 10.4210
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.493      1.203  -6.229 2.25e-07 ***
## guinearice    40.288      1.337  30.135  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.239 on 40 degrees of freedom
## Multiple R-squared:  0.9578, Adjusted R-squared:  0.9568
## F-statistic: 908.1 on 1 and 40 DF,  p-value: < 2.2e-16
```

Cases of spurious regression might appear to give reasonable short-term forecasts, but they will generally not continue to work into the future.

Selecting predictors

When there are many possible predictors, we need some strategy to select the best predictors to use in a regression model.

Not recommended:

- plot the forecast variable against a particular predictor and if it shows no noticeable relationship, drop it.
- do a multiple linear regression on all the predictors and disregard all variables whose p-values are greater than 0.05.

Instead, use a measure of predictive accuracy: Adjusted R^2, Cross-validation, Akaike's Information Criterion, Corrected Akaike's Information Criterion, Schwarz Bayesian Information Criterion.

Adjusted R^2

As mentioned previously, *adjusted* R^2:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{T-1}{T-k-1},$$

where T is the number of observations and k is the number of predictors. This is an improvement on R^2 as it will no longer increase with each added predictor. Maximizing $\bar{R}^2$ is equivalent to minimizing the following estimate of the variance of the forecast errors:

$$\hat{\sigma}^2 = \frac{\text{SSE}}{T-k-1}.$$

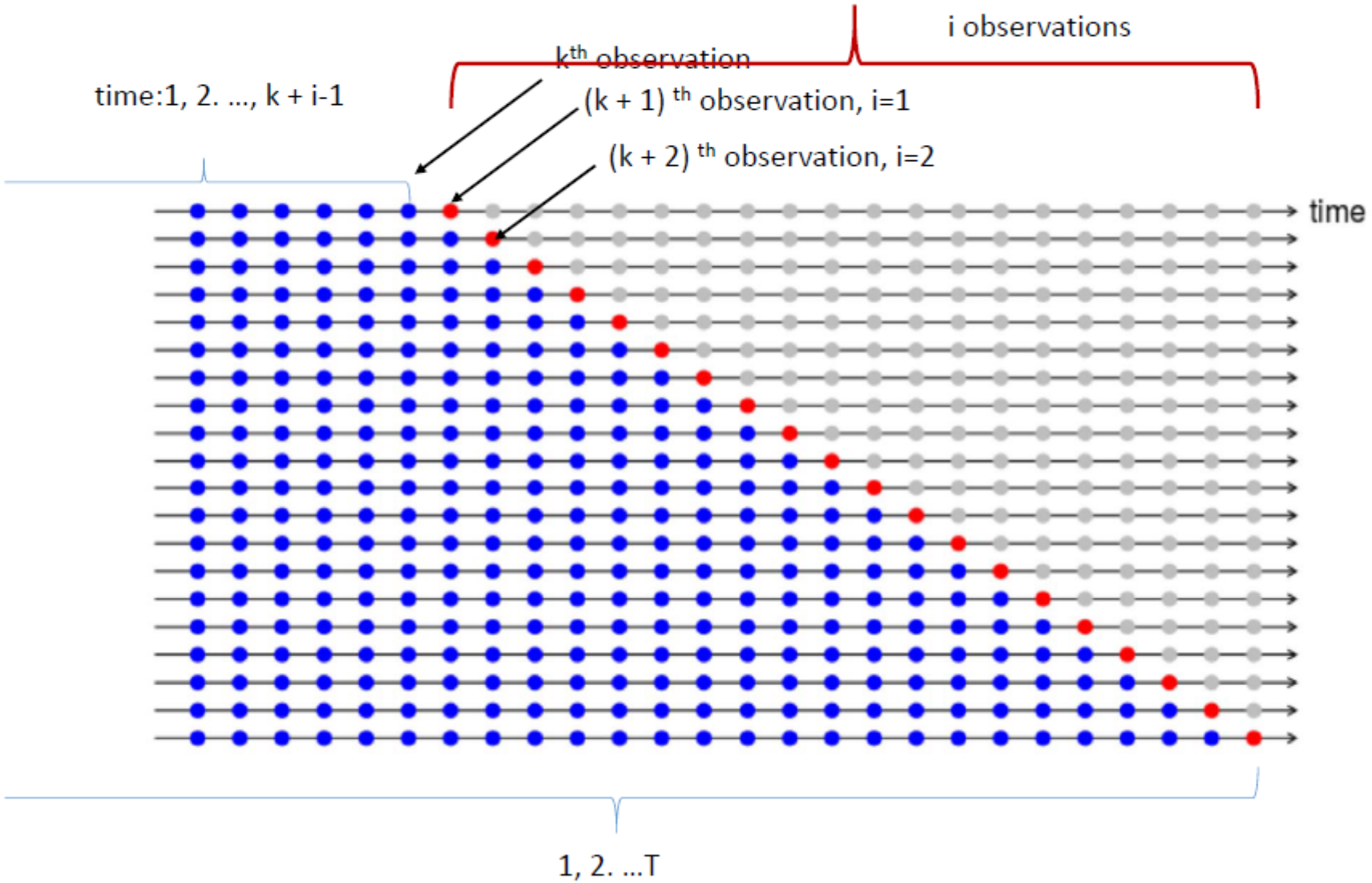
Maximizing $\bar{R}^2$ works quite well as a method of selecting predictors, although it does tend to err on the side of selecting too many predictors.

Cross-validation

Cross-validation is a very useful way of determining the predictive ability of a model. In general, leave-one-out cross-validation for regression can be carried out using the following steps.

1. Remove observation i from the data set, and fit the model using the remaining data. Then compute the forecast error ($e_i^* = y_i - \hat{y}_{k+i}$) for the omitted observation. (This is not the same as the residual because the ith observation was not used in estimating the value of $\hat{y}_{k+i}$.)
2. Repeat step 1 for $i = 1, \ldots, T$.
3. Compute the MSE from $e_1^*, \ldots, e_T^*$. We shall call this the CV.

Under this criterion, the best model is the one with the smallest value of CV.


i observations
k th observation
time:1, 2. ..., k + i-1
(k + 1) th observation, i=1
(k + 2) th observation, i=2
time
1, 2. ...T

Cross validation in regression models

Akaike's Information Criterion

A closely-related method is Akaike's Information Criterion, which we define as

$$\text{AIC} = T\log\left(\frac{\text{SSE}}{T}\right) + 2(k+2),$$

where T is the number of observations used for estimation and k is the number of predictors in the model. The $k+2$ part of the equation occurs because there are $k+2$ parameters in the model — the k coefficients for the predictors, the intercept and the variance of the residuals. The idea here is to penalize the fit of the model (SSE) with the number of parameters that need to be estimated.

The model with the minimum value of the AIC is often the best model for forecasting. For large values of T, minimizing the AIC is equivalent to minimizing the CV value.

Corrected Akaike's IC

For small values of T, the AIC tends to select too many predictors, and so a bias-corrected version of the AIC has been developed:

$$\text{AIC}_\text{c} = \text{AIC} + \frac{2(k+2)(k+3)}{T-k-3}.$$

As with the AIC, the AICc should be minimized.

Schwarz Bayesian IC

$$\text{BIC} = T\log\left(\frac{\text{SSE}}{T}\right) + (k+2)\log(T).$$

The model chosen by BIC is either the same as that chosen by AIC, or one with fewer terms. This is because BIC penalizes the SSE more heavily than the AIC.

Many statisticians like to use BIC because it has the feature that if there is a true underlying model, then with enough data the BIC will select that model. However, in reality there is rarely if ever a true underlying model, and even if there was a true underlying model, selecting that model will not necessarily give the best forecasts (because the parameter estimates may not be accurate).

To obtain these measures in R, use

```
CV(fit.consMR)
```

```
##           CV          AIC         AICc          BIC        AdjR2
##    0.1163477 -409.2980298 -408.8313631 -389.9113781    0.7485856
```

Please remember to load the library fpp2 and create fit.consMR using the tslm (which I did but do not show) before running CV(fit.consMR), i.e.,

```
fit.consMR <- tslm(Consumption ~ Income + Production + Unemployment + Savings,
data=uschange)
```

Example: US consumption

- As shown, to obtain all these measures in R, use CV(fit.consMR) .
- In the multiple regression example for forecasting US consumption we considered four predictors. With four predictors, there are $2^4 = 16$ possible models.
- Now we can check if all four predictors are actually useful, or whether we can drop one or more of them. All 16 models were fitted and the results are summarised in Table 5.1.

- A “1” indicates that the predictor was included in the model, and a “0” means that the predictor was not included in the model. Hence the first row shows the measures of predictive accuracy for a model including all four predictors.
- The results have been sorted according to the AICc and therefore the best models are given at the top of the table, and the worst at the bottom of the table

Table 5.1: All 16 possible models for forecasting US consumption with 4 predictors.

Income	Production	Savings	Unemployment	CV	AIC	AICc	BIC	AdjR2
1	1	1	1	0.116	-409	-409	-390	0.749
1	0	1	1	0.116	-408	-408	-392	0.746
1	1	1	0	0.118	-407	-407	-391	0.745
1	0	1	0	0.129	-389	-389	-376	0.716
1	1	0	1	0.278	-243	-243	-227	0.386
1	0	0	1	0.283	-238	-238	-225	0.365
1	1	0	0	0.289	-236	-236	-223	0.359
0	1	1	1	0.293	-234	-234	-218	0.356
0	1	1	0	0.300	-229	-229	-216	0.334
0	1	0	1	0.303	-226	-226	-213	0.324
0	0	1	1	0.306	-225	-224	-212	0.318
0	1	0	0	0.314	-220	-219	-210	0.296
0	0	0	1	0.314	-218	-218	-208	0.288
1	0	0	0	0.372	-185	-185	-176	0.154
0	0	1	0	0.414	-164	-164	-154	0.052
0	0	0	0	0.432	-155	-155	-149	0.000

16 possible models for forecasting US consumption with 4 predictors.

- The best model contains all four predictors.
- There is clear separation between the models in the first four rows and the ones below.

  -This indicates that Income and Savings are both more important variables than Production and Unemployment. Also,

- The first two rows have almost identical values of CV, AIC and AICc. So we could possibly drop the Production variable and get very similar forecasts. Note that Production and Unemployment are highly (negatively) correlated,
- The most of the predictive information in Production is also contained in the Unemployment variable.

End of Part 1, Chapter 5. To be continued …