THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

Time Series: Autocorrelation

Lecture 18

STA 371G

Predicting oil prices

Date	Oil price ($)
1/1/2013	112.98
1/1/2014	107.94
1/1/2015	55.38
1/1/2016	36.85
1/1/2017	55.05
1/1/2018	?

Predicting oil prices

Date	Oil price ($)
1/1/2013	112.98
1/1/2014	107.94
1/1/2015	55.38
1/1/2016	36.85
1/1/2017	55.05
1/1/2018	?

- What's the best prediction of the price of oil on January 1?

Predicting oil prices

Date	Oil price ($)
1/1/2013	112.98
1/1/2014	107.94
1/1/2015	55.38
1/1/2016	36.85
1/1/2017	55.05
1/1/2018	?

- What's the best prediction of the price of oil on January 1?
- Does next year's price depend on this year's?

Time series

In a time series, data are not necessarily independent. (Often it is not!)

Time series

In a time series, data are not necessarily independent. (Often it is not!)

Time series:

- A sequence of measurements of the same variable collected over time.
- The measurements are made at regular time intervals (most commonly daily, weekly, monthly, quarterly, or yearly).
- The variances are not necessarily constant over time either.

Some examples

- S&P 500 index (or any stock price)
- iPhone sales worldwide
- U.S. unemployment rate
- U.S. inflation rate
- Crime rate in Austin

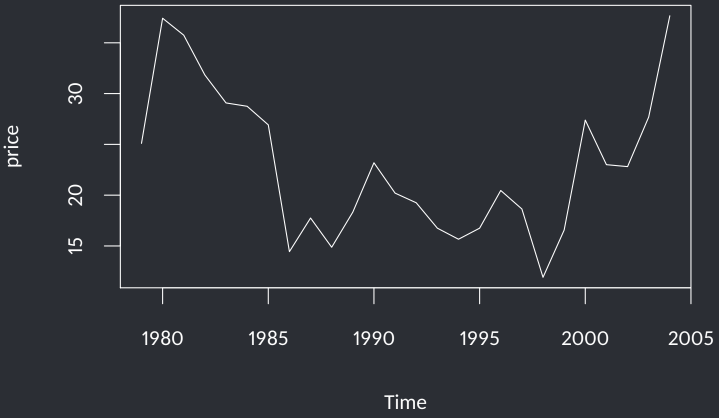
Some examples

- S&P 500 index (or any stock price)
- iPhone sales worldwide
- U.S. unemployment rate
- U.S. inflation rate
- Crime rate in Austin

Any of these could be measured at weekly, monthly, yearly etc. intervals; and each would be a different time series.

Oil Prices 1979-2004

```
# Convert the data into a time series object
price <- ts(oil$price, start=1979, frequency=1)
# Frequency: # of data points per year
plot(price)
```



Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year.

Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year. To predict the oil price in a given year, can we use the previous year's price?

Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year. To predict the oil price in a given year, can we use the previous year's price?

y_t: The oil price at the end of the year t

Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year. To predict the oil price in a given year, can we use the previous year's price?

y_t: The oil price at the end of the year t

t	y_t	y_{t-1}
…	…	…
1999	16.56	11.91
2000	27.39	16.56
2001	23	27.39
2002	22.81	23
…	…	…

Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year. To predict the oil price in a given year, can we use the previous year's price?

y_t: The oil price at the end of the year t

t	y_t	y_{t-1}
...	...	...
1999	16.56	11.91
2000	27.39	16.56
2001	23	27.39
2002	22.81	23
...	...	...

y_{t-1} column is obtained by shifting y_t by 1.

Oil Prices 1979-2004

We argued that oil prices are not independent year-over-year. To predict the oil price in a given year, can we use the previous year's price?

y_t: The oil price at the end of the year t

t	y_t	y_{t-1}
...	...	...
1999	16.56	11.91
2000	27.39	16.56
2001	23	27.39
2002	22.81	23
...	...	...

y_{t-1} column is obtained by shifting y_t by 1.

The lag between y_t and y_{t-1} is one time-step.

Compute one-lag time series

```
# Create lag 1 time series.
priceL1 <- lag(price, k=-1)
# Put them together
price_all <- cbind(price=price, priceL1=priceL1)
price_all[1:5,]

     price priceL1
[1,] 25.10      NA
[2,] 37.42   25.10
[3,] 35.75   37.42
[4,] 31.83   35.75
[5,] 29.08   31.83
```

Compute one-lag time series

```r
# Create lag 1 time series.
priceL1 <- lag(price, k=-1)
# Put them together
price_all <- cbind(price=price, priceL1=priceL1)
price_all[1:5,]

     price priceL1
[1,] 25.10      NA
[2,] 37.42   25.10
[3,] 35.75   37.42
[4,] 31.83   35.75
[5,] 29.08   31.83
```

`priceL1` in the first row is NA because we did not have data from 1978 to put under y_{t-1} column of 1979.

Linear regression model

The simple linear regression model is:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$$

Linear regression model

The simple linear regression model is:

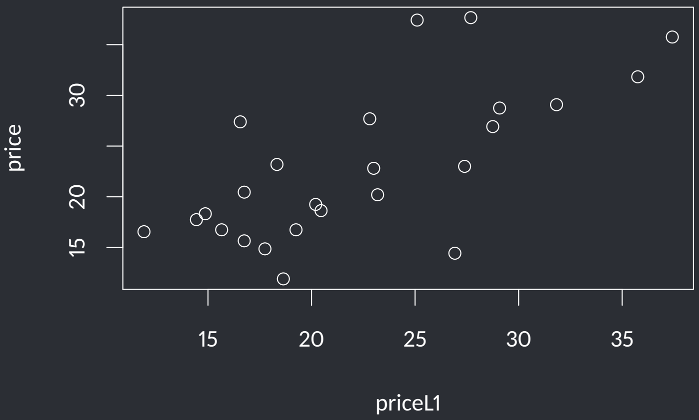
$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t$$

Note that we obtained our predictor from the response itself, lagged 1 time step!

When we use such a model, we expect to see a linear relation between the predictor and the response. Let’s see if there is such a relation!

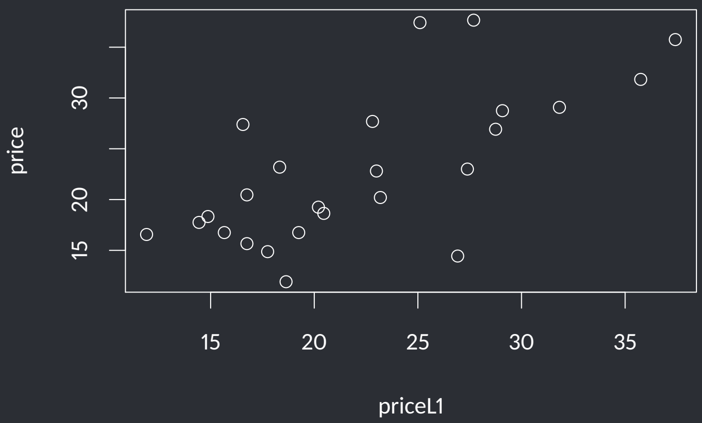
When we use such a model, we expect to see a linear relation between the predictor and the response. Let's see if there is such a relation!

```r
plot(price ~ priceL1, xy.labels=F, xy.lines=F)
```



When we use such a model, we expect to see a linear relation between the predictor and the response. Let's see if there is such a relation!

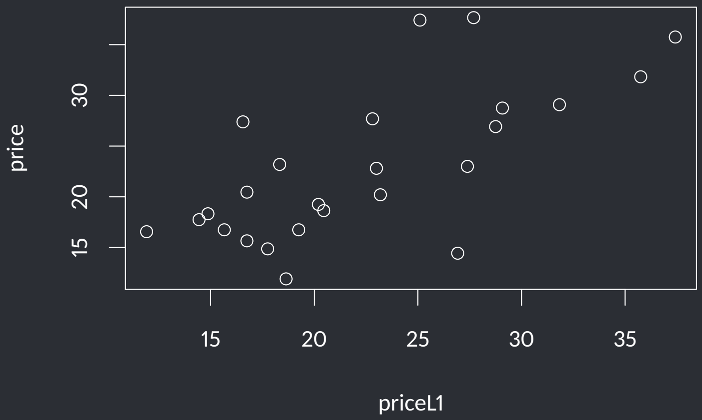
```
plot(price ~ priceL1, xy.labels=F, xy.lines=F)
```



The oil prices seem to be correlated with its first lag!

When we use such a model, we expect to see a linear relation between the predictor and the response. Let's see if there is such a relation!

```
plot(price ~ priceL1, xy.labels=F, xy.lines=F)
```



The oil prices seem to be correlated with its first lag! This is called autocorrelation.

```r
model <- lm(price ~ priceL1, data=price_all)
summary(model)
```

```


Call:
lm(formula = price ~ priceL1, data = price_all)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.9046  -2.9505  -0.8162   1.6303  12.4595 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.8724     3.8389   1.530 0.139722    
priceL1       0.7605     0.1642   4.632 0.000116 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.454 on 23 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.4827,Adjusted R-squared:  0.4602 
F-statistic: 21.46 on 1 and 23 DF,  p-value: 0.0001164
```

```r
model <- lm(price ~ priceL1, data=price_all)
summary(model)
```

```


Call:
lm(formula = price ~ priceL1, data = price_all)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.9046  -2.9505  -0.8162   1.6303  12.4595 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.8724     3.8389   1.530 0.139722    
priceL1       0.7605     0.1642   4.632 0.000116 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.454 on 23 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.4827,Adjusted R-squared:  0.4602 
F-statistic: 21.46 on 1 and 23 DF,  p-value: 0.0001164
```

This is a first-order autoregressive, AR(1), model.

AR(2) model

Let’s try to add one more lag.

```
# Create lag 2 time series.
priceL2 <- lag(price, k=-2)
# Put them together
price_all <- cbind(price=price, priceL1=priceL1, priceL2=priceL2)
price_all[1:5,]
```

```
     price priceL1 priceL2
[1,] 25.10      NA      NA
[2,] 37.42   25.10      NA
[3,] 35.75   37.42   25.10
[4,] 31.83   35.75   37.42
[5,] 29.08   31.83   35.75
```

AR(2) model

Let's try to add one more lag.

```
# Create lag 2 time series.
priceL2 <- lag(price, k=-2)
# Put them together
price_all <- cbind(price=price, priceL1=priceL1, priceL2=priceL2)
price_all[1:5,]
```

```
     price priceL1 priceL2
[1,] 25.10      NA      NA
[2,] 37.42   25.10      NA
[3,] 35.75   37.42   25.10
[4,] 31.83   35.75   37.42
[5,] 29.08   31.83   35.75
```

The model then becomes:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t$$

AR(2) model

```
model <- lm(price ~ priceL1 + priceL2, data=price_all)
summary(model)
```

```
Call:
lm(formula = price ~ priceL1 + priceL2, data = price_all)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.6861  -3.0937   0.7269   2.3375  10.9071 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.1749     3.7363   1.920 0.068505 .  
priceL1       0.8427     0.2073   4.064 0.000557 ***
priceL2      -0.1646     0.2094  -0.786 0.440530    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.911 on 21 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.5411,Adjusted R-squared:  0.4974 
F-statistic: 12.38 on 2 and 21 DF,  p-value: 0.0002807
```

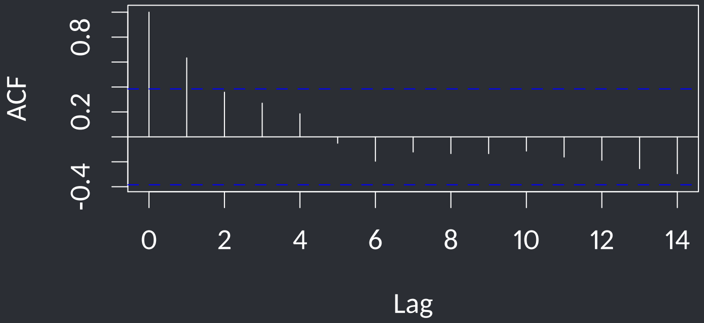
Autocorrelation Function

`priceL2` is not statistically significant.

Autocorrelation Function

`priceL2` is not statistically significant. The Autocorrelation Function (ACF) plots the correlation between the series and each of its lags, to help determine how many lags to include in our model.

```
acf(price)
```



Stationarity assumption

AR models (and many time series models) assume the stationarity of the series.

Stationarity assumption

AR models (and many time series models) assume the stationarity of the series.

A time series is stationary if

- the mean, $E(y_t)$, is the same over time
- the variance, $\text{Var}(y_t)$ is the same over time
- the correlation between y_t and y_{t-h} is the same over time.

Stationarity assumption

To check on the stationarity of a time series, we use the Augmented Dickey-Fuller test.

```
library(tseries)
adf.test(price)



Augmented Dickey-Fuller Test

data:  price
Dickey-Fuller = -0.28178, Lag order = 2, p-value = 0.9844
alternative hypothesis: stationary
```

Stationarity assumption

To check on the stationarity of a time series, we use the Augmented Dickey-Fuller test.

```
library(tseries)
adf.test(price)
```

```


Augmented Dickey-Fuller Test

data:  price
Dickey-Fuller = -0.28178, Lag order = 2, p-value = 0.9844
alternative hypothesis: stationary
```

The null hypothesis is that the series is “explosive” (non-stationary).

Stationarity assumption

To check on the stationarity of a time series, we use the Augmented Dickey-Fuller test.

```
library(tseries)
adf.test(price)



Augmented Dickey-Fuller Test

data:  price
Dickey-Fuller = -0.28178, Lag order = 2, p-value = 0.9844
alternative hypothesis: stationary
```

The null hypothesis is that the series is “explosive” (non-stationary).

Since the p-value is very high, we cannot reject the null hypothesis—this data is not stationary and an AR model is not appropriate.

Stationarity assumption

Trend and seasonality in a time series violate stationarity.

Stationarity assumption

Trend and seasonality in a time series violate stationarity.

However, many time series have either trend or seasonality, and often both!

Stationarity assumption

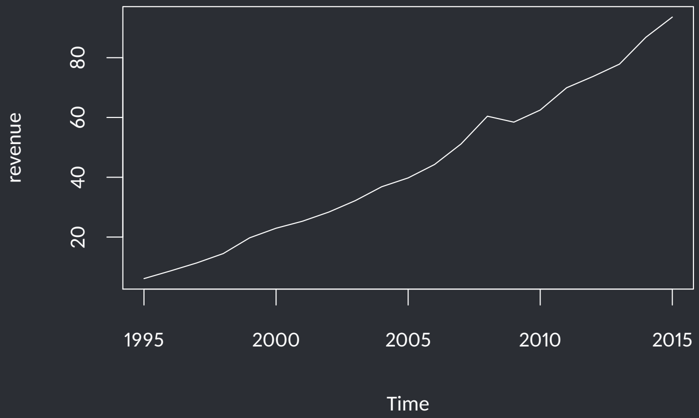
Trend and seasonality in a time series violate stationarity.

However, many time series have either trend or seasonality, and often both!

Let's look at Microsoft's revenue over years...

Increasing trend of Microsoft

```r
# Convert the data into a time series object
# Frequency: # of data points per year (default is 1)
revenue <- ts(microsoft$revenue, start=1995, frequency=1)
plot(revenue)
```



Increasing trend of Microsoft

Let’s also verify the non-stationarity of the data through the ADF test.

Increasing trend of Microsoft

Let's also verify the non-stationarity of the data through the ADF test.

```
adf.test(revenue)
```

```
Augmented Dickey-Fuller Test

data:  revenue
Dickey-Fuller = -0.44992, Lag order = 2, p-value = 0.9771
alternative hypothesis: stationary
```

Increasing trend of Microsoft

Let’s also verify the non-stationarity of the data through the ADF test.

```
adf.test(revenue)



Augmented Dickey-Fuller Test

data:  revenue
Dickey-Fuller = -0.44992, Lag order = 2, p-value = 0.9771
alternative hypothesis: stationary
```

Again, since the p-value is very high, we cannot reject the null hypothesis.

Increasing trend of Microsoft

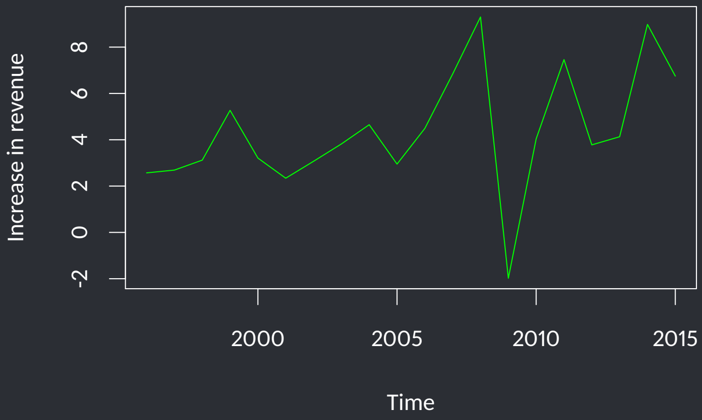
Microsoft's revenue is certainly increasing. But the amount of increase each year seems to be relatively constant.

```r
# Create lag 1 time series.
revenueL1 <- lag(revenue, k=-1)
# Look at the increase (first difference) each year
revenue_increase <- revenue - revenueL1
# Put them together
revenue_all <- cbind(revenue=revenue, revenueL1=revenueL1,
                     revenue_increase=revenue_increase)
revenue_all[1:8,]
```

```
     revenue revenueL1 revenue_increase
[1,]    6.10        NA               NA
[2,]    8.67      6.10             2.57
[3,]   11.36      8.67             2.69
[4,]   14.48     11.36             3.12
[5,]   19.75     14.48             5.27
[6,]   22.96     19.75             3.21
[7,]   25.30     22.96             2.34
[8,]   28.37     25.30             3.07
```

Increasing trend of Microsoft

```
plot(revenue_increase, col='green', ylab='Increase in revenue')
```



Increasing trend of Microsoft

```
adf.test(revenue_increase)


Augmented Dickey-Fuller Test

data:  revenue_increase
Dickey-Fuller = -3.0968, Lag order = 2, p-value = 0.1545
alternative hypothesis: stationary
```

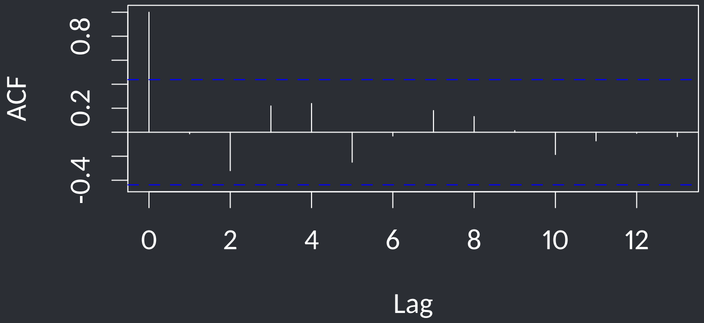
Increasing trend of Microsoft

```
adf.test(revenue_increase)


Augmented Dickey-Fuller Test

data:  revenue_increase
Dickey-Fuller = -3.0968, Lag order = 2, p-value = 0.1545
alternative hypothesis: stationary
```

Still cannot reject the null hypothesis and further transformation is required. But let’s move on to model the yearly increase in revenue.

Autocorrelation function using the increase in revenue

```r
acf(revenue_increase)
```



Autocorrelation function using the increase in revenue

What should we expect?

- There does not seem to be a very strong autocorrelation in the revenue increase time series.
- The autocorrelation with the second lag is higher than the first one.

```r
revenue_increaseL1 <- lag(revenue_increase, k=-1)
revenue_increaseL2 <- lag(revenue_increase, k=-2)
rev_inc_all <- cbind(revenue_increase = revenue_increase,
                     revenue_increaseL1 = revenue_increaseL1,
                     revenue_increaseL2 = revenue_increaseL2)
rev_inc_all[1:5,]
```

```
     revenue_increase revenue_increaseL1 revenue_increaseL2
[1,]             2.57                 NA                 NA
[2,]             2.69               2.57                 NA
[3,]             3.12               2.69               2.57
[4,]             5.27               3.12               2.69
[5,]             3.21               5.27               3.12
```

```r
model_rev_inc <- lm(revenue_increase ~ revenue_increaseL1
                      + revenue_increaseL2, data=rev_inc_all)
summary(model_rev_inc)
```

```


Call:
lm(formula = revenue_increase ~ revenue_increaseL1 + revenue_increaseL2,
    data = rev_inc_all)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9423 -1.6555 -0.1413  1.0997  5.1529

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.59190    1.70754   3.860  0.00154 **
revenue_increaseL1 -0.08454    0.24354  -0.347  0.73331
revenue_increaseL2 -0.41570    0.26843  -1.549  0.14231
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.613 on 15 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.1391,Adjusted R-squared:  0.02435
F-statistic: 1.212 on 2 and 15 DF,  p-value: 0.3251
```

This model could be used to predict the increase in the revenue, instead of the revenue itself.

The predicted increase could be added on top of the revenue at $t - 1$ to predict the revenue in t.

But the overall R^2 is still very low, so it’s not going to give us a great prediction.

Another option: predict Y from t

We can also just predict revenue from time using a simple linear regression:

```
Call:
lm(formula = revenue ~ year, data = microsoft)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2324 -2.8046 -0.2805  1.5983  6.7318

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -8542.5594   232.8877  -36.68   <2e-16 ***
year            4.2826     0.1162   36.87   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.223 on 19 degrees of freedom
Multiple R-squared:  0.9862,Adjusted R-squared:  0.9855
F-statistic:  1359 on 1 and 19 DF,  p-value: < 2.2e-16
```

Another option: predict *Y* from *t*

We can also just predict revenue from time using a simple linear regression:

```
Call:
lm(formula = revenue ~ year, data = microsoft)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2324 -2.8046 -0.2805  1.5983  6.7318

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -8542.5594   232.8877  -36.68   <2e-16 ***
year            4.2826     0.1162   36.87   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.223 on 19 degrees of freedom
Multiple R-squared:  0.9862,Adjusted R-squared:  0.9855
F-statistic:  1359 on 1 and 19 DF,  p-value: < 2.2e-16
```

This gives a good prediction (R^2 is high!). But independence is violated so we can't rely on the veracity of the *p*-values.