THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

# Time Series: Autocorrelation

## Lecture 18

STA 371G

# Predicting beer production over time

Goal: Predict beer production in the US (in millions of gallons)

# An autoregressive model for beer production

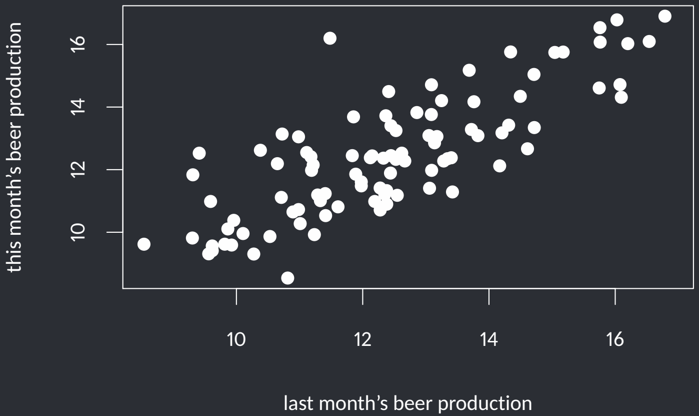
If we believe last month's beer figures are a good prediction for next month's beer figures, we might choose an AR(1) model:

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

This means that the next month's beer figures depend only on the current month's beer figures, plus random noise.

Let's look at whether this seems a reasonable assumption

```
# Convert the data into a time series object
beer <- ts(beer.df$beer, start=c(1970,7), frequency=12)
# Create a lag-1 version
beerL1 <- lag(beer, k=-1)
#combine them in a data frame
beer_all <- cbind(beer=beer,beerL1=beerL1)
# Frequency: # of data points per year
plot(beer~beerL1,data=beer_all,xlab = "last month\'s beer production",ylab="this 
duction",pch=19)
```



```r
model <- lm(beer ~ beerL1, data=beer_all)
summary(model)
```

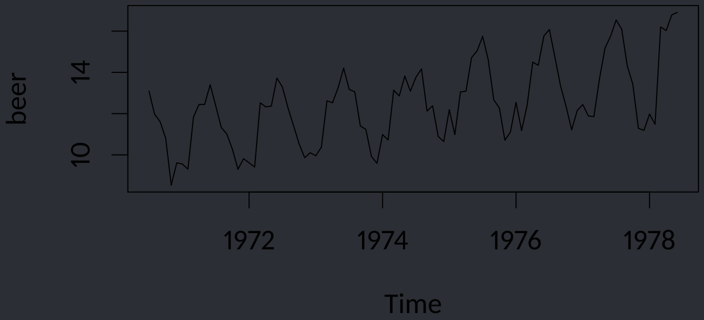
```


Call:
lm(formula = beer ~ beerL1, data = beer_all)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.6018 -0.8428 -0.1902  0.8539  4.5109 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.25720    0.80557   2.802  0.00618 ** 
beerL1       0.82136    0.06417  12.800  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.183 on 93 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.6379,Adjusted R-squared:  0.634 
F-statistic: 163.8 on 1 and 93 DF,  p-value: < 2.2e-16
```

Last month's beer production is statistically significant. The $R^2$ is good, but not amazing…

perhaps we can do better!

Hmmm... there seems to be more of a pattern here than in the oil price data from last week!

```r
plot(beer)
```



# Types of temporal variation

There are several types of temporal variation we might want to predict!

Cyclic variation is *unpredictable* up and down movement, of the sort we saw last week.

- Length of cycle may vary
- Often caused by multiple interacting factors.
- Example: Stock prices vary due to recessions, depressions and recoveries.
- Example: Sales may be affected by fashions.

## Types of temporal variation

There are several types of temporal variation we might want to predict!

Cyclic variation is *unpredictable* up and down movement, of the sort we saw last week.

- Length of cycle may vary
- Often caused by multiple interacting factors.
- Example: Stock prices vary due to recessions, depressions and recoveries.
- Example: Sales may be affected by fashions.

If we don't know those factors, often the best we can do is predict based on the last time point... i.e. an autoregressive model.

# Trend

A trend is a persistent, overall, upwards or downwards pattern in the data.

- Example: Effects due to population growth (e.g. demand on health services).
- Example: “Moore’s Law” – processor speeds double every two years.
- Could be linear or non-linear – the trend may continue at a constant rate, or accelerate, or level off.

We definitely saw a trend in the beer production numbers!

## Seasonal

- Seasonal variation is a regular pattern of up and down fluctuation.
- The length of the cycle is the same (e.g. yearly, monthly)
- Caused by effects such as weather, holidays.
- It is predictable.
- Example: Toy sales increase in December.
- Example: Ice cream sales affected by the weather.

There was clear seasonality in the beer production numbers!

# Modeling a trend

Let's first deal with the upward trend. We can try predicting production from time using a simple linear regression:

```
summary(lm(beer ~ month_count, data=beer.df))
```

```
Call:
lm(formula = beer ~ month_count, data = beer.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.87729 -1.48470  0.00505  1.31704  2.80344 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.576746   0.334920  31.580  < 2e-16 ***
month_count  0.038784   0.005996   6.468 4.42e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.628 on 94 degrees of freedom
Multiple R-squared:  0.308,Adjusted R-squared:  0.3006 
F-statistic: 41.84 on 1 and 94 DF,  p-value: 4.415e-09
```

# Modeling a trend

Let's first deal with the upward trend. We can try predicting production from time using a simple linear regression:

```
summary(lm(beer ~ month_count, data=beer.df))
```

```
Call:
lm(formula = beer ~ month_count, data = beer.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.87729 -1.48470  0.00505  1.31704  2.80344 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.576746   0.334920  31.580  < 2e-16 ***
month_count  0.038784   0.005996   6.468 4.42e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.628 on 94 degrees of freedom
Multiple R-squared:  0.308,Adjusted R-squared:  0.3006 
F-statistic: 41.84 on 1 and 94 DF,  p-value: 4.415e-09
```

# Seasonal variation

How can we deal with the seasonal effect?

## Seasonal variation

How can we deal with the seasonal effect?

The simplest solution is to treat month as a categorical variable!

- This means that we assume there is some commonality between September 1970, September 1971, September 1972...
- More general, we might want to treat quarters or days of the weeks as categorical variables.

```
Call:
lm(formula = beer ~ month_count + month, data = beer.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.23574 -0.30407 -0.01724  0.40160  1.65361 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    11.462493   0.255140  44.926  < 2e-16 ***
month_count     0.037887   0.002351  16.114  < 2e-16 ***
monthAugust     0.400097   0.317257   1.261 0.210801    
monthDecember  -2.752326   0.316839  -8.687 2.76e-13 ***
monthFebruary  -2.686726   0.316734  -8.483 7.08e-13 ***
monthJanuary   -2.159214   0.316778  -6.816 1.39e-09 ***
monthJuly       1.164859   0.317405   3.670 0.000428 ***
monthJune       1.227101   0.316734   3.874 0.000213 ***
monthMarch     -0.440613   0.316708  -1.391 0.167874    
monthMay        0.642738   0.316708   2.029 0.045618 *  
monthNovember  -3.026064   0.316917  -9.548 5.23e-15 ***
monthOctober   -1.544927   0.317013  -4.873 5.20e-06 ***
monthSeptember -0.932040   0.317127  -2.939 0.004263 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6334 on 83 degrees of freedom
Multiple R-squared:  0.9075,Adjusted R-squared:  0.8941 
F-statistic: 67.86 on 12 and 83 DF,  p-value: < 2.2e-16
```

- Now that we've modeled seasonality, we've got an $R^2$ of 0.9075 and an adjusted $R^2$ of 0.8941… pretty good!
- We've modeled the trend and the seasonality…

- Now that we've modeled seasonality, we've got an $R^2$ of 0.9075 and an adjusted $R^2$ of 0.8941… pretty good!
- We've modeled the trend and the seasonality…
- … maybe there's also a cyclic effect! Let's take a look at the residuals.

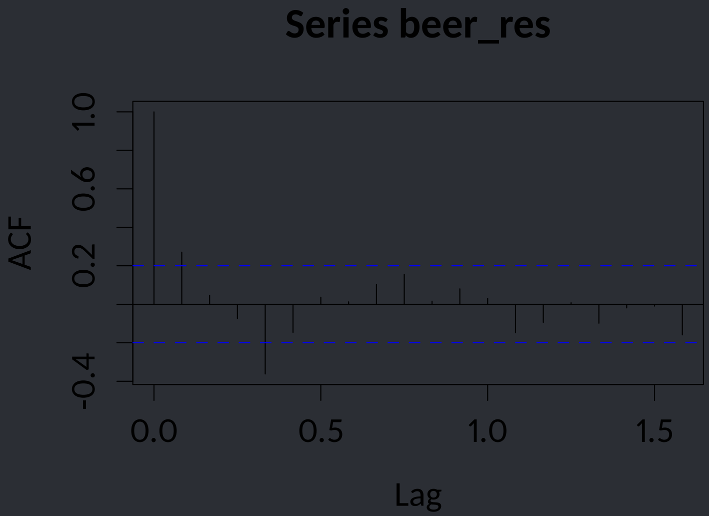
```r
# Convert the residuals into a time series object

beer_res <- ts(model$residuals, start=c(1970,7), frequency=12)
plot(beer_res)
```

Let’s try running an autoregressive model on the residuals

```r
beer_res_L1 <- lag(beer_res, k=-1)
res_all <- cbind(beer_res = beer_res,
                 beer_res_L1 = beer_res_L1)
model <- lm(beer_res~beer_res_L1,data=res_all)
```

```
Call:
lm(formula = beer_res ~ beer_res_L1, data = res_all)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.25571 -0.34325 -0.00875  0.35091  1.86982 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) -0.002829   0.058897  -0.048   0.9618   
beer_res_L1  0.273770   0.099983   2.738   0.0074 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.574 on 93 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:  0.0746,Adjusted R-squared:  0.06465 
F-statistic: 7.498 on 1 and 93 DF,  p-value: 0.007404
```

Statistically significant… but not really practically significant




Autocorrelation function agrees... fairly low autocorrelation at lag one.