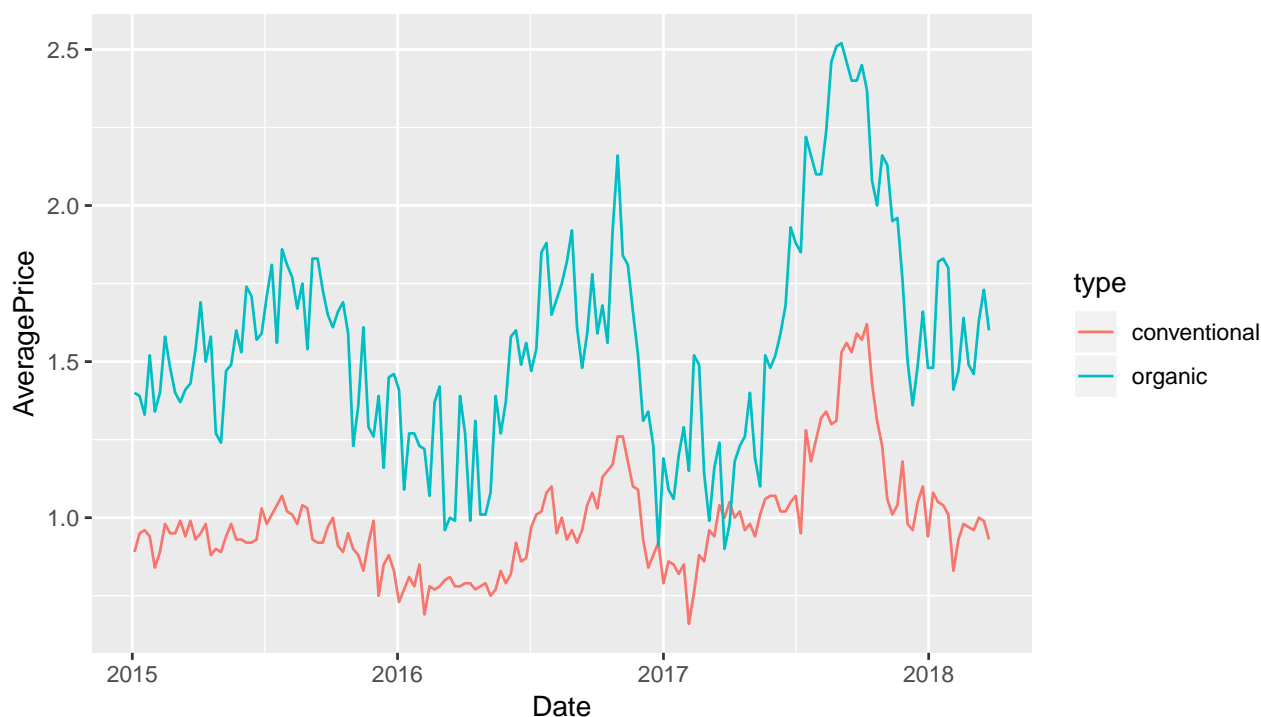


# STAT 436 / 536 - Lecture 17: Key

## Multivariate Time Series

- Multivariate time series data consist of recordings of multiple variables at the same time.
- For instance, consider the prices of conventional and organic avocados in the western region of the United States.

```
avo <- read_csv('http://math.montana.edu/ahoegh/teaching/timeseries/data/avocado_west.csv')  
  
ggplot(data = avo, aes(y = AveragePrice, x = Date)) + geom_line(aes(color = type))
```



## Spurious Regression

- In general, we use regression to assess relationships between a set of variables. However, we need to be cautious in a time-series setting.
- The textbook has an example about Australian electricity and chocolate production sharing an increasing trend. In fact, the two variables have correlation of 0.958. It turns out that the increases in both variables are largely driven by an increasing Australian population, and hence, the relationship is deemed *spurious*.
- More generally in time series analyses, spurious regression results when both series share an underlying stochastic trend. This often takes the form of a random walk (e.g. a unit root).

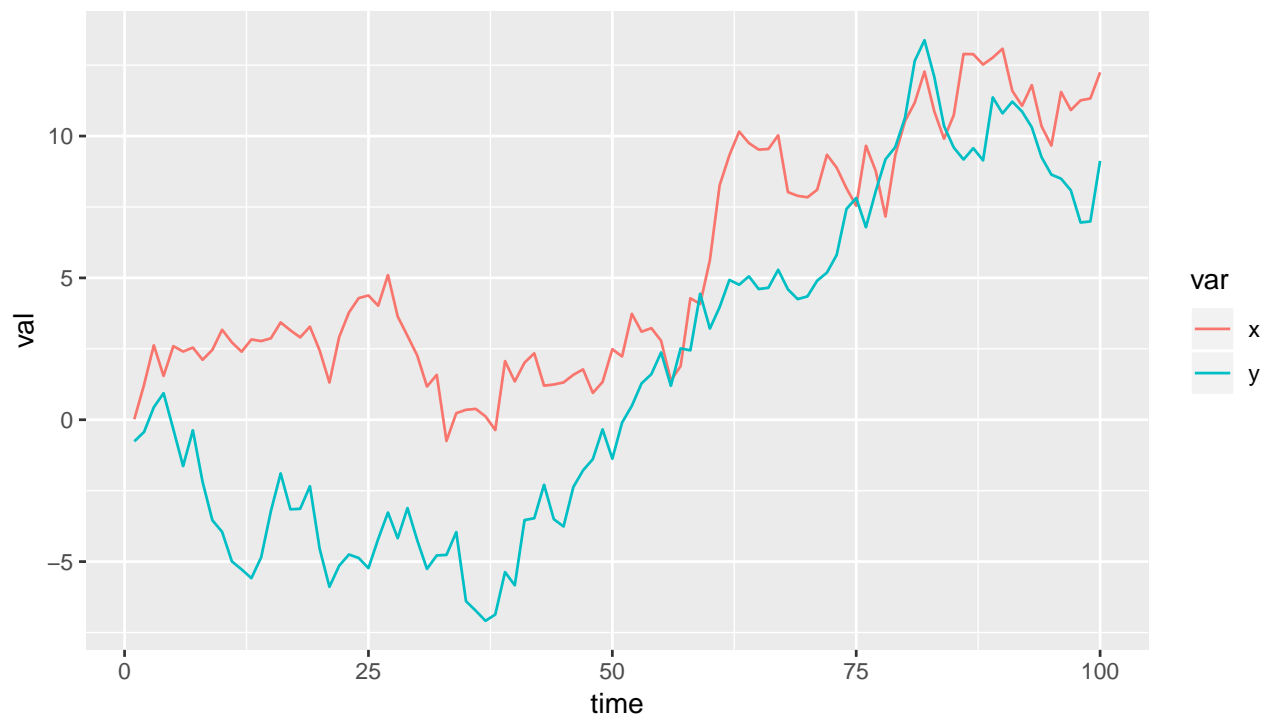
- Consider and example with two simulated random walks.

```
set.seed(10)
x <- rnorm(100)
y <- rnorm(100)

for (i in 2:100){
  x[i] <- x[i-1] + rnorm(1)
  y[i] <- y[i-1] + rnorm(1)
}

comb <- data.frame(time = rep(1:100,2), val=c(x,y), var = rep(c('x','y'), each = 100))

ggplot(data=comb, aes(y=val, x=time)) + geom_line(aes(color = var))
```



```
cor(x,y)
```

```
## [1] 0.9039082
```

- Each time series is a random walk, but the noise terms are uncorrelated. Despite this, there is a high degree of correlation between the two time series due to the stochastic trend.

- Now consider the correlation between the two differenced time series.

```
cor(diff(x), diff(y))
```

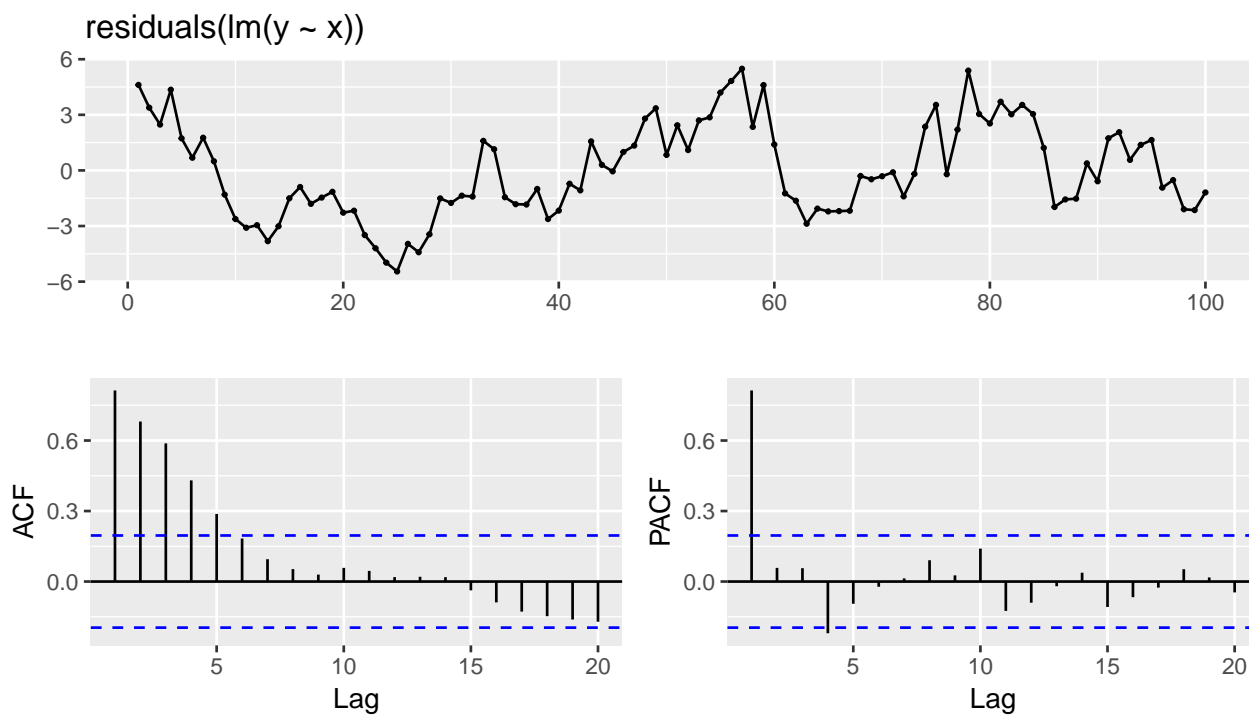
```
## [1] 0.2018941
```

- The result is much smaller correlation. - **Q:** how do you anticipate these results changing for a different seed or a longer time series?

- With spurious regression, we often see that  $t$  or  $z$  scores in regression are quite high and sometimes *statistically significant*. This is directly related to the high correlation values that we have seen in the example above.

- Looking at the residuals can also highlight potential issues with time series regression models.

```
ggtsdisplay(residuals(lm(y~x)))
```



## Unit Root Testing

- In addition to the techniques that we have seen for testing for stationarity we can directly look for unit roots.

- The Dickey-Fuller test is one hypothesis test for assessing whether a unit root exists. Essentially, for a model written as  $x_t = \alpha x_{t-1} + w_t$ , the model tests whether  $\alpha = 1$  (random-walk non-stationary) or  $\alpha < 1$  (stationary).

- Note that the alternative hypothesis is that the series is stationary,

```
adf.test(x)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: x  
## Dickey-Fuller = -2.2343, Lag order = 4, p-value = 0.4796  
## alternative hypothesis: stationary
```

```
adf.test(y)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: y  
## Dickey-Fuller = -2.9218, Lag order = 4, p-value = 0.1948  
## alternative hypothesis: stationary
```

so both of these results would fail to reject the null hypothesis that the series is non-stationary.

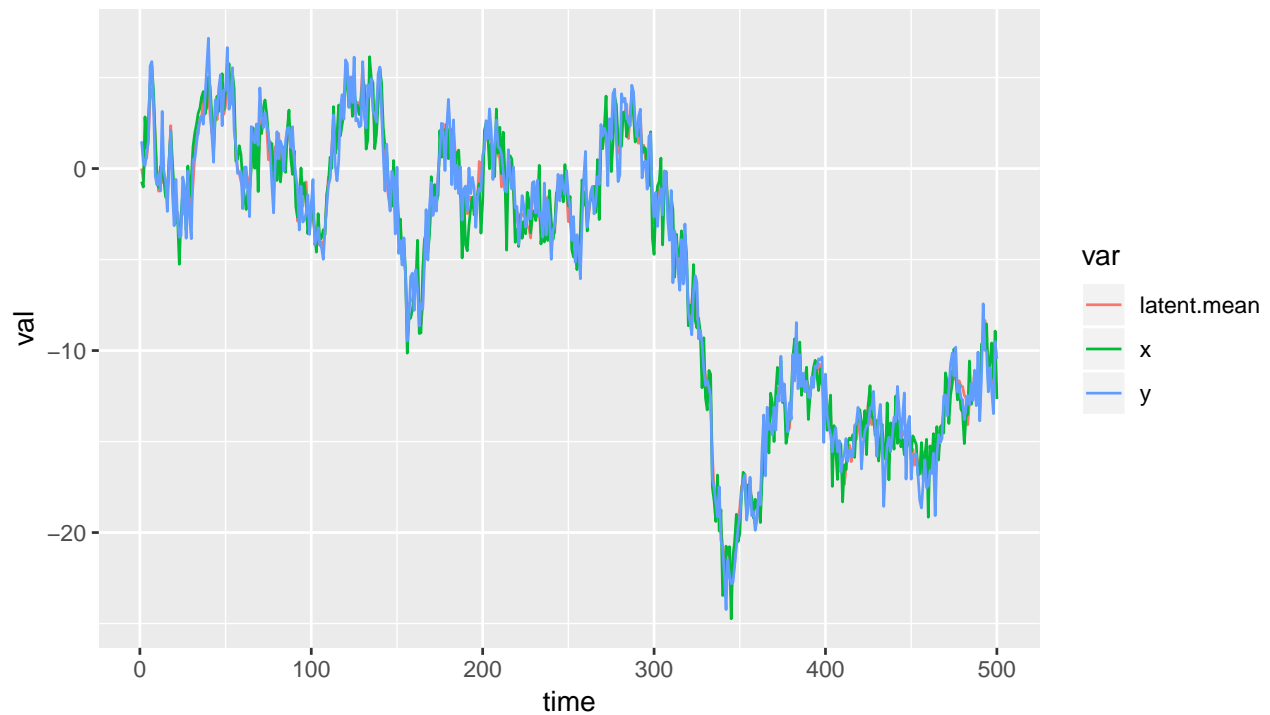
## Cointegration

- Two time series can have unit roots *and* be related. This phenomenon is called *cointegration*.
- Two non-stationary time series  $\{x_t\}$  and  $\{y_t\}$  are cointegrated if some linear combination  $ax_t + by_t$ , is stationary. Where  $a$  and  $b$  are constants.
- As an example of cointegrated time series, consider two time series that are generated from a latent mean process.

```
latent.mean <- y2 <- x2 <- rep(0, 100)
for (i in 2:500){
  latent.mean[i] <- latent.mean[i-1] + rnorm(1)
}
x2 <- latent.mean + rnorm(500)
y2 <- latent.mean + rnorm(500)

coint.df <- data.frame(time = rep(1:500,3), val = c(latent.mean, x2, y2), var = rep(c('latent.mean', 'x', 'y'), each = 500))

ggplot(data = coint.df, aes(x = time, y = val)) + geom_line(aes(color = var))
```



- The Phillips-Ouliaris test can be used to assess whether two time series are cointegrated.

```
po.test(cbind(x2,y2))
```

```
##
## Phillips-Ouliaris Cointegration Test
##
## data: cbind(x2, y2)
## Phillips-Ouliaris demeaned = -506.95, Truncation lag parameter =
## 4, p-value = 0.01
```

In this case, we'd reject the null that the two time series *are not* cointegrated.

## Vector Autoregressive Models

- One approach to model multivariate time series is to extend our ARIMA model framework.
- Two time series,  $\{x_t\}$  and  $\{y_t\}$  follow a vector autoregressive process of order 1 (VAR(1)) if

$$\begin{aligned}x_t &= \theta_{11}x_{t-1} + \theta_{12}y_{t-1} + w_{x,t} \\ y_t &= \theta_{21}x_{t-1} + \theta_{22}y_{t-1} + w_{y,t},\end{aligned}$$

where  $\{w_{x,t}\}$  and  $\{w_{y,t}\}$  are white noise and the  $\theta$  values are model parameters.

- Similar to the univariate case, characteristic equations can be defined to assess whether the models are stationary.
- VAR models can be fit using the `ar()` or `VAR()` functions.

```
VAR(cbind(x2,y2), p = 1)
```

```
##
## VAR Estimation Results:
## =====
##
## Estimated coefficients for equation x2:
## =====
## Call:
## x2 = x2.l1 + y2.l1 + const
##
##      x2.l1      y2.l1      const
## 0.4158914 0.5583220 -0.1705214
##
##
## Estimated coefficients for equation y2:
## =====
## Call:
## y2 = x2.l1 + y2.l1 + const
##
##      x2.l1      y2.l1      const
## 0.4634932 0.5201526 -0.1027198
```

- The `VARselect` model can also be used to choose the order of a model based on some criteria.

```
VARselect(cbind(x2,y2), lag.max = 5, type="const")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
##
## $criteria
##           1           2           3           4           5
## AIC(n) 1.559967 1.505137 1.506841 1.521433 1.533933
## HQ(n)  1.579974 1.538482 1.553524 1.581453 1.607291
## SC(n)  1.610931 1.590078 1.625758 1.674326 1.720802
## FPE(n) 4.758666 4.504777 4.512471 4.578818 4.636443
```

- The `predict(n.ahead =)` and `resid()` functions can be applied to VAR objects.

### State-Space approach

- More general multivariate time series models can be fit using a state-space approach.

- The `dlm` package in R has the capability to fit models of this type.