Modeling Multiple Trends

Mike Bader

June 24, 2016

Contents

Modeling Multiple Trends

Hopefully at this point, you can see that measuring growth doesn't require anything fancy; it's ultimately just a regression model. That being said, for the types of things that we care about, we often don't want to look at a trend for a single unit. We care about the progresion of a disease among many people. We care about the growth of BMI over time, the propensity to violence, or earnings over a population of people. Or we care about how prices change in metropolitan areas across the country.

Variation in Fake Home Value in Two Metro Areas

Let's start with a really simple example where we have trends across two units that we want to consider. And, to keep consistency, we will generate a process of change in housing prices across two metropolitan areas. Let's keep the model that we had from the last section for our fake New York metro. Let's write out the general model that we used:

$$\ln(price_t) = \beta_0 + \beta_1(month_t) + \epsilon_t$$

And then for our fake New York metro, we set the following parameters:

$$\ln(price_t) = 5.10 + 0.01 \times month_t + \epsilon_t$$

Just to remind us, let's look quickly at the data for our fake New York metro (the first line gets rid of the data from our mean-only model):

```
ny.metro.12mo <- metro.12mo[,c(-2:-5,-12:-13)]
ny.coef <- fake.trend.model$coefficients
round(ny.metro.12mo,4)
```

```
##     t lnprice_t  price_t beta_0 beta_1 pred.lnprice_t    ehat
## 1   0    5.0972 163.5668  5.095 0.0105         5.0950  0.0022
## 2   1    5.1073 165.2191  5.095 0.0105         5.1055  0.0018
## 3   2    5.1099 165.6586  5.095 0.0105         5.1160 -0.0061
```

```
## 4   3    5.1327 169.4775  5.095 0.0105         5.1266  0.0062
## 5   4    5.1318 169.3286  5.095 0.0105         5.1371 -0.0053
## 6   5    5.1494 172.3203  5.095 0.0105         5.1476  0.0017
## 7   6    5.1611 174.3590  5.095 0.0105         5.1581  0.0030
## 8   7    5.1661 175.2217  5.095 0.0105         5.1687 -0.0026
## 9   8    5.1762 177.0045  5.095 0.0105         5.1792 -0.0030
## 10  9    5.1914 179.7264  5.095 0.0105         5.1897  0.0017
## 11 10    5.1914 179.7115  5.095 0.0105         5.2002 -0.0089
## 12 11    5.2201 184.9517  5.095 0.0105         5.2108  0.0093
```

Write Model and Generate Fake Home Value Trend in Second Metro Area

Now let's generate data for another metropolitan area. Let's generate a fake Washington, D.C. metro. I would guess that homes cost about 85% as much in Washington as they do in New York. That means that our *intercept*, β_0 in our fake D.C. metro will be 95% lower: $5.10\times\ 0.95 = 4.845$. I also anticipate that housing prices have stagnated in the D.C. area given the long-term consequences of sequestration and Congressional inaction. I anticipate that prices only grew by 0.1% per month. I imagine that prices would be somewhat unstable, so I will set the standard deviation to be larger than the underlying trend, $\sigma = 0.01$. That makes our our fake D.C. look like this:

$$\ln\left(price_t\right) = 4.335 + 0.001 \times month_t + \epsilon_t$$

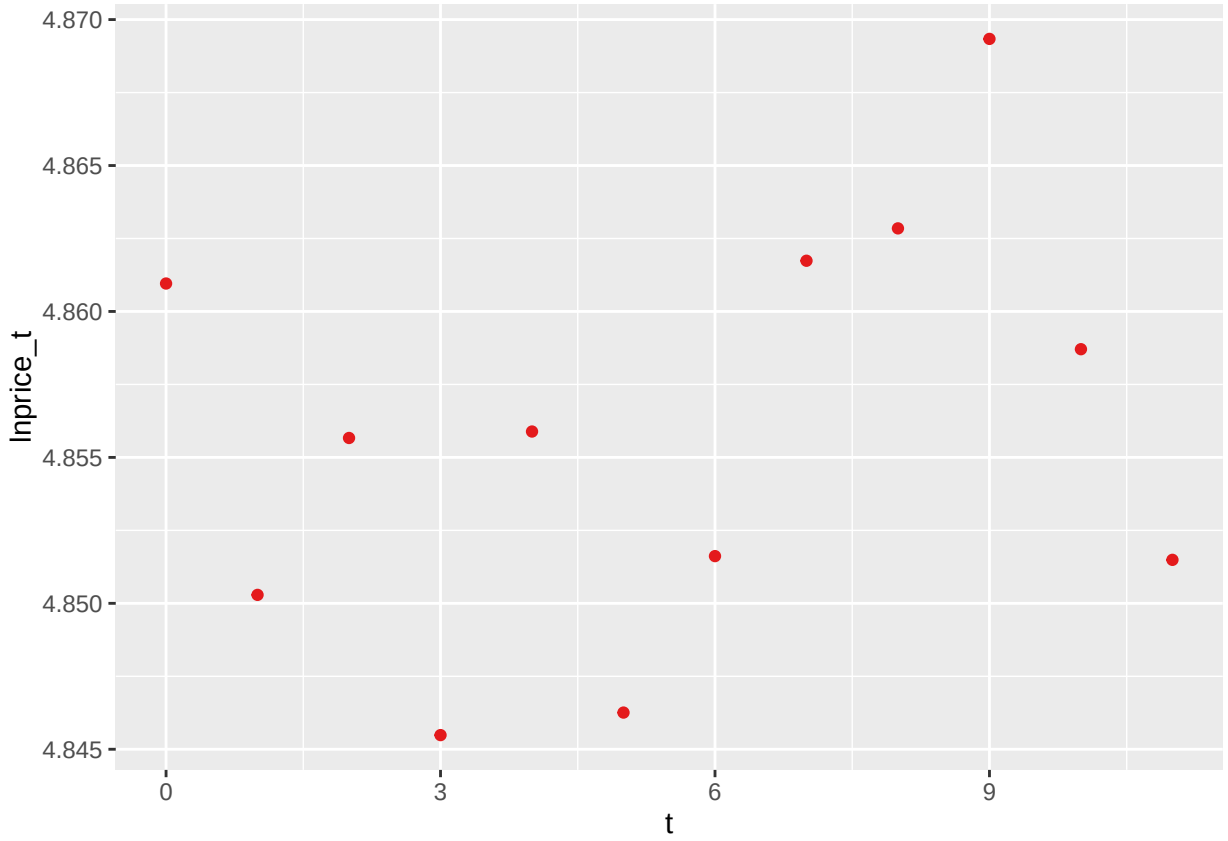
```
set.seed(5831209)
N       <- 12
beta_0 <- 4.845
beta_1 <- 0.001
sigma  <- 0.01

dc.metro.12mo <- data.frame(t=seq(0,N-1))
dc.metro.12mo <- mutate(dc.metro.12mo
                ,lnprice_t = beta_0 + beta_1*t + rnorm(12,mean=0,sd=sigma)
                ,price_t = exp(lnprice_t)
                )
```

And to look at the data in tabular and graphical formats:

```
dc.color <- "#e41a1c"
dc.metro.12mo
```

```
##     t lnprice_t  price_t
## 1   0  4.860957 129.1477
## 2   1  4.850290 127.7774
## 3   2  4.855665 128.4661
## 4   3  4.845486 127.1650
## 5   4  4.855886 128.4945
## 6   5  4.846257 127.2631
## 7   6  4.851618 127.9472
## 8   7  4.861738 129.2486
## 9   8  4.862847 129.3920
## 10  9  4.869339 130.2347
## 11 10  4.858707 128.8575
## 12 11  4.851490 127.9308
```

```
fake.plt <- qplot(t,lnprice_t,data=dc.metro.12mo,colour = I(dc.color))
fake.plt
```



Analyze Fake Trend Data

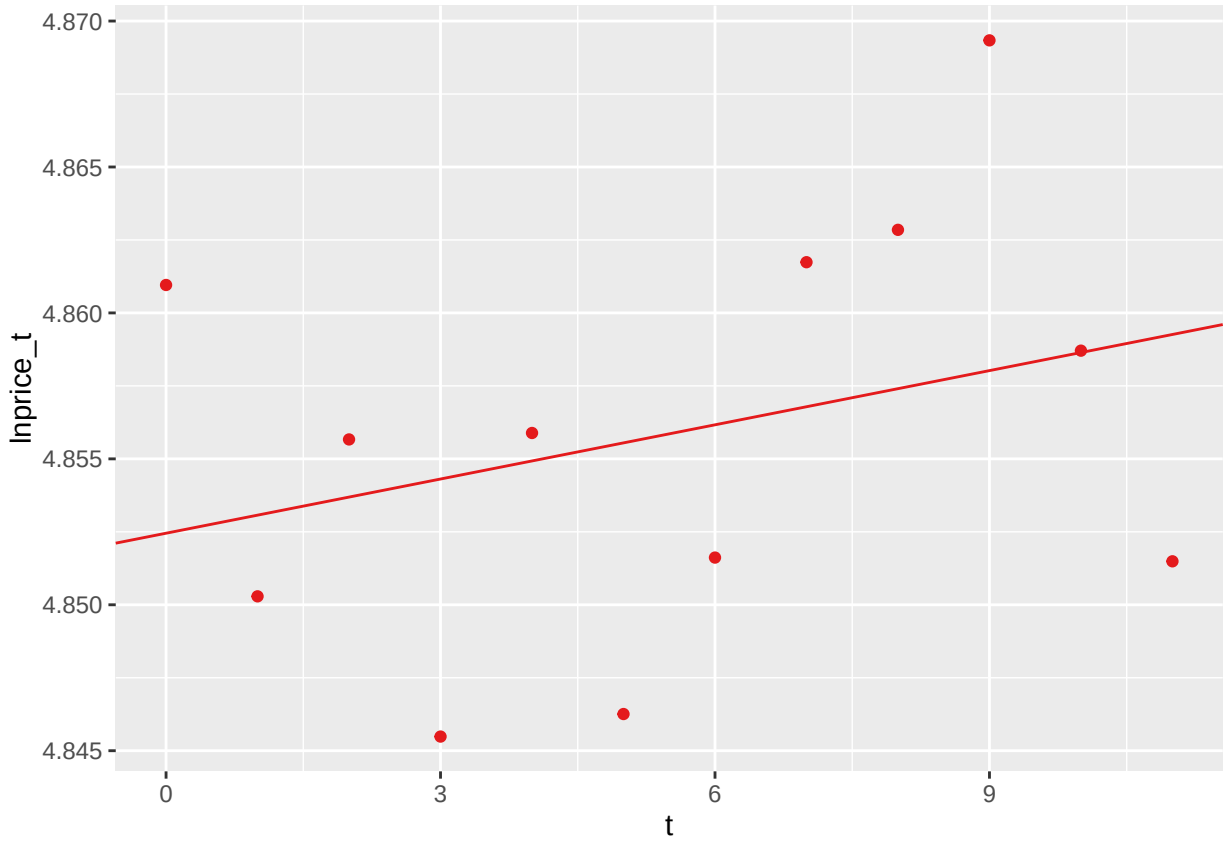
Now let's analyze the trend in our fake Washington, D.C. metropolitan area and add the predicted values and errors to our data frame.

```
dc.fake.trend.model <- lm(lnprice_t ~ t,data=dc.metro.12mo)
summary(dc.fake.trend.model)
```

```
##
## Call:
## lm(formula = lnprice_t ~ t, data = dc.metro.12mo)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0092902 -0.0053541  0.0005117  0.0050751  0.0113153
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 4.8524518  0.0038868 1248.447   <2e-16 ***
## t           0.0006191  0.0005986    1.034    0.325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.007158 on 10 degrees of freedom
## Multiple R-squared:  0.09663,    Adjusted R-squared:  0.006291
## F-statistic:  1.07 on 1 and 10 DF,  p-value: 0.3254
```

```
dc.coef <- dc.fake.trend.model$coefficients
fake.plt + geom_abline(intercept=dc.coef[1],slope=dc.coef[2],color=I(dc.color))
```




```
dc.metro.12mo <- mutate(dc.metro.12mo
                        , beta_0 = dc.coef[1]
                        , beta_1 = dc.coef[2]
                        , pred.lnprice_t = beta_0 + beta_1*t
                        , ehat = lnprice_t - pred.lnprice_t
                       )
dc.metro.12mo
```

```
##     t lnprice_t  price_t   beta_0       beta_1 pred.lnprice_t           ehat
## 1   0  4.860957 129.1477 4.852452 0.0006190503       4.852452  0.00850530636
## 2   1  4.850290 127.7774 4.852452 0.0006190503       4.853071 -0.00278115801
## 3   2  4.855665 128.4661 4.852452 0.0006190503       4.853690  0.00197555163
## 4   3  4.845486 127.1650 4.852452 0.0006190503       4.854309 -0.00882339414
## 5   4  4.855886 128.4945 4.852452 0.0006190503       4.854928  0.00095847547
## 6   5  4.846257 127.2631 4.852452 0.0006190503       4.855547 -0.00929019337
## 7   6  4.851618 127.9472 4.852452 0.0006190503       4.856166 -0.00454821375
## 8   7  4.861738 129.2486 4.852452 0.0006190503       4.856785  0.00495265470
## 9   8  4.862847 129.3920 4.852452 0.0006190503       4.857404  0.00544240420
```

```
## 10  9  4.869339 130.2347 4.852452 0.0006190503       4.858023  0.01131532123
## 11 10  4.858707 128.8575 4.852452 0.0006190503       4.858642  0.00006482538
## 12 11  4.851490 127.9308 4.852452 0.0006190503       4.859261 -0.00777157971
```

```
round(c(mean(dc.metro.12mo$ehat),sd(dc.metro.12mo$ehat)),4)
```

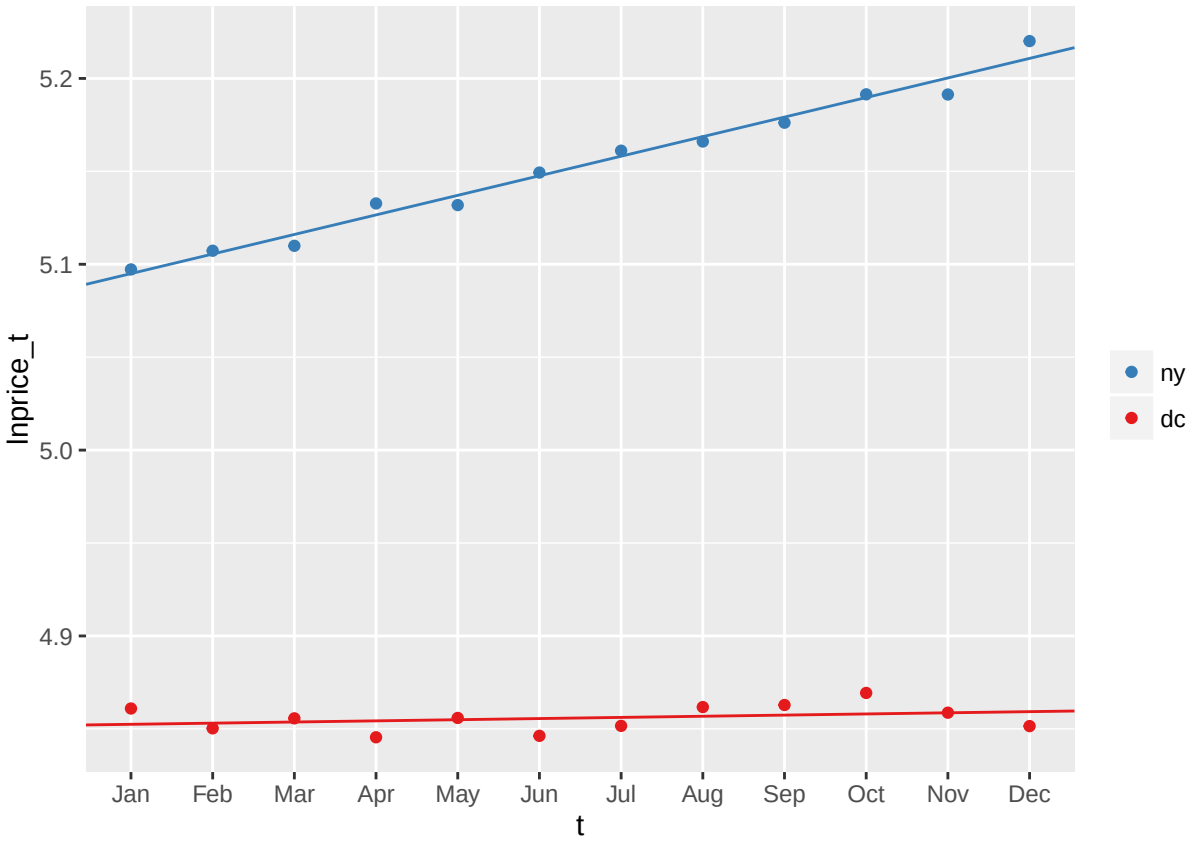
```
## [1] 0.0000 0.0068
```

Everything looks in order. Now, for fun, let's plot our fake New York and fake D.C. metro data together on the same plot. In order to do that, we need to combine our two data frames into a single data frame. First, we will create a new variable in each dataset called `i` that will take on the value of `1` for New York and `2` for D.C. Second, we will merge the data together on `i`; obviously none of the data will merge since the datasets don't share a common value of `i`. The result will be a dataframe containing 24 rows: 12 for our fake New York and 12 for our fake D.C. Then we plot both together on the same plot.

```
ny.metro.12mo$i <- 1
dc.metro.12mo$i <- 2
merged.metro.12mo <- rbind(ny.metro.12mo,dc.metro.12mo)
merged.metro.12mo
```

```
##     t lnprice_t  price_t   beta_0       beta_1 pred.lnprice_t           ehat i
## 1   0  5.097222 163.5668 5.094986 0.0105263590       5.094986  0.00223613323 1
## 2   1  5.107273 165.2191 5.094986 0.0105263590       5.105512  0.00176079613 1
## 3   2  5.109929 165.6586 5.094986 0.0105263590       5.116038 -0.00610927430 1
## 4   3  5.132720 169.4775 5.094986 0.0105263590       5.126565  0.00615582326 1
## 5   4  5.131841 169.3286 5.094986 0.0105263590       5.137091 -0.00525003588 1
## 6   5  5.149355 172.3203 5.094986 0.0105263590       5.147617  0.00173733400 1
## 7   6  5.161116 174.3590 5.094986 0.0105263590       5.158144  0.00297271744 1
## 8   7  5.166052 175.2217 5.094986 0.0105263590       5.168670 -0.00261786350 1
## 9   8  5.176175 177.0045 5.094986 0.0105263590       5.179196 -0.00302137062 1
## 10  9  5.191436 179.7264 5.094986 0.0105263590       5.189723  0.00171296949 1
## 11 10  5.191353 179.7115 5.094986 0.0105263590       5.200249 -0.00889625735 1
## 12 11  5.220095 184.9517 5.094986 0.0105263590       5.210775  0.00931902811 1
## 13  0  4.860957 129.1477 4.852452 0.0006190503       4.852452  0.00850530636 2
## 14  1  4.850290 127.7774 4.852452 0.0006190503       4.853071 -0.00278115801 2
## 15  2  4.855665 128.4661 4.852452 0.0006190503       4.853690  0.00197555163 2
## 16  3  4.845486 127.1650 4.852452 0.0006190503       4.854309 -0.00882339414 2
## 17  4  4.855886 128.4945 4.852452 0.0006190503       4.854928  0.00095847547 2
## 18  5  4.846257 127.2631 4.852452 0.0006190503       4.855547 -0.00929019337 2
## 19  6  4.851618 127.9472 4.852452 0.0006190503       4.856166 -0.00454821375 2
## 20  7  4.861738 129.2486 4.852452 0.0006190503       4.856785  0.00495265470 2
## 21  8  4.862847 129.3920 4.852452 0.0006190503       4.857404  0.00544240420 2
## 22  9  4.869339 130.2347 4.852452 0.0006190503       4.858023  0.01131532123 2
## 23 10  4.858707 128.8575 4.852452 0.0006190503       4.858642  0.00006482538 2
## 24 11  4.851490 127.9308 4.852452 0.0006190503       4.859261 -0.00777157971 2
```

```
ny.color <- "#377eb8"
fake.2metro <- qplot(t,lnprice_t,data=merged.metro.12mo,color=factor(i,labels=c("ny","dc"))) +
                    scale_x_continuous(breaks=seq(0,11),labels=month.abb,minor_breaks=NULL) +
                    scale_colour_manual(values=c(ny.color,dc.color)) +
                    geom_abline(intercept=ny.coef[1],slope=ny.coef[2],colour=ny.color) +
                    geom_abline(intercept=dc.coef[1],slope=dc.coef[2],colour=dc.color) +
                    theme(legend.title=element_blank())
fake.2metro
```

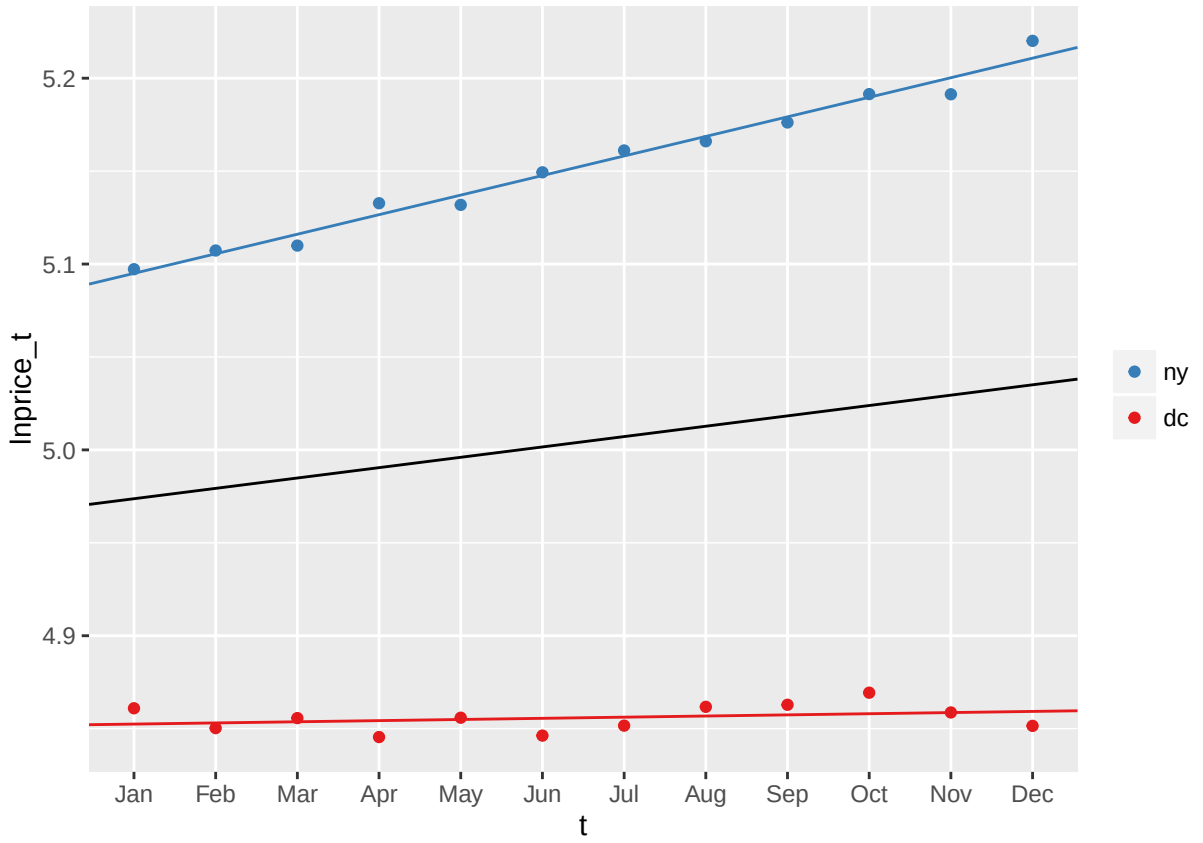


Now that we have both of these datasets together, we can model the trend *across* the two (fake) cities easily and we add the overall model to the plot:

```
fake.nydc.model <- lm(lnprice_t ~ t, data=merged.metro.12mo)
nydc.coef <- fake.nydc.model$coefficients
summary(fake.nydc.model)
```

```
##
## Call:
## lm(formula = lnprice_t ~ t, data = merged.metro.12mo)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.183529 -0.151876  0.005371  0.149161  0.185076
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.973719   0.060000  82.896   <2e-16 ***
## t           0.005573   0.009240   0.603    0.553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1563 on 22 degrees of freedom
## Multiple R-squared:  0.01627,    Adjusted R-squared:  -0.02845
## F-statistic: 0.3638 on 1 and 22 DF,  p-value: 0.5526
```

```
fake.2metro + geom_abline(intercept=nydc.coef[1],slope=nydc.coef[2])
```

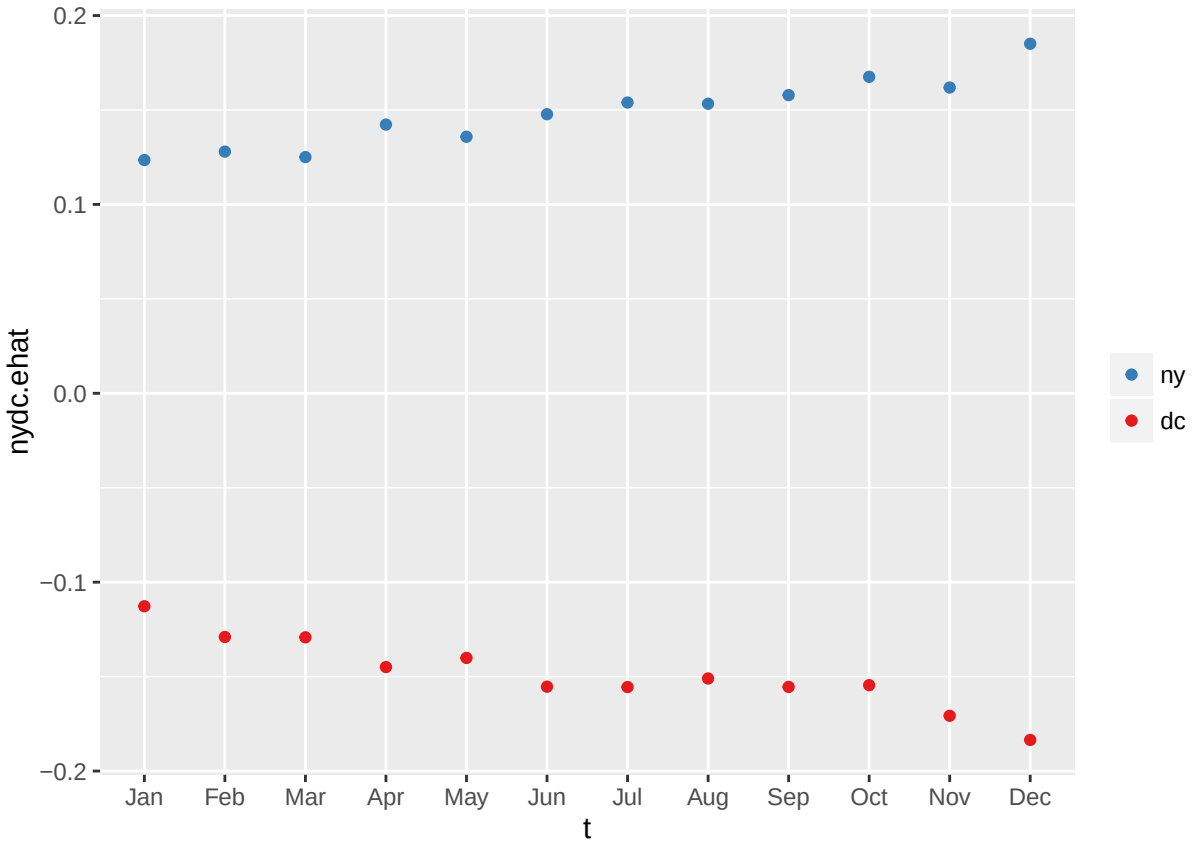


Cool, now we have started to summarize the data across two cities. Let's take a look at the residual plot of the overall model:

```
merged.metro.12mo$nydc.pred.lnprice_t <- fake.nydc.model$fitted.values
merged.metro.12mo$nydc.ehat <- fake.nydc.model$residuals
round(mean(merged.metro.12mo$nydc.ehat,3))
```

```
## [1] 0
```

```
fake.2metro.r <- qplot(t,nydc.ehat,data=merged.metro.12mo,color=factor(i,labels=c("ny","dc"))) +
                    scale_x_continuous(breaks=seq(0,11),labels=month.abb,minor_breaks=NULL) +
                    scale_colour_manual(values=c(ny.color,dc.color)) +
                    theme(legend.title=element_blank())
fake.2metro.r
```



Uh, oh... Now we see a pattern in our residuals, which is exactly what we are *not* supposed to see. Although the overall mean of the residuals equals zero, all of the residuals from our fake New York are positive and all of the residuals from our fake D.C. are negative! That's not good. **This is why we need to think of growth models differently.**

Write the Model of Growth Across a Population

This plot shows us that variation in models of growth across a population have two components: they have a distribution of unit-level (in our case metro-level) errors and a distribution of time-level errors. In fact, we now have a (very small) distribution of β_0s and β_1s. When we combine the data together, the intercept will equal the mean of the two β_0 coefficients from our two models and the slope will equal the average of the two β_1 coefficients.

What this means is that our β_0 and β_1 coefficients can now be represented by their own equations:

$$\beta_{0i} = \gamma_{00} + \upsilon_{0i}$$
$$\beta_{1i} = \gamma_{10} + \upsilon_{1i}$$

The top equation shows that each metro area, indexed by i, has it's own intercept, β_{0i} (the value homes in January 2015) that equals the overall intercept across metropolitan areas plus some metro-level unique variation, υ_{0i}, away from that overall mean. We generally assume that the distribution of these errors (which we don't have yet because we only have two intercepts) follow a normal distribution that we represent as τ_0.

The second equation is analogous to the first, exept it describes the variation in the slopes. Each metro area follows its own trend, β_{1i}, that equals the overall trend, γ_{10}, and some unique variation off of that trend, υ_{1i}. You can see this in the plot of the NY and DC residuals above: because we made the growth rate in D.C. be

slower than that of New York, the residuals from the combined model get larger in magnitude in the later months. Once again, we assume (and will later assert) that the slopes are drawn from a normal distribution.

Let's go back to our original equation for trend analysis and add this new information into the equation:

$$\ln\left(price_{ti}\right) = \beta_{0i} + \beta_1\left(month_t\right) + \epsilon_{ti}$$

What we did here was indicate that the logged price per square foot depends not on some single intercept and slope, but on the intercept and slope *for the particular metro* i *in which we observe prices at time* t. We can now substitute the equations from above to show how observations of logged price vary:

$$\ln\left(price_{ti}\right) = \gamma_{00} + \gamma_{10}\left(month_t\right) + \upsilon_{0i} + \upsilon_{1i} + \epsilon_{ti}$$

This equation says that the (logged) median home value in a metro in month t equals the average home price across metros, γ_{00} (notice there is no index, so all observations have the same value), plus the average trend across metros, γ_{10} (also no index), plus the unique deviation metro i's intercept away from the mean intercept, υ_{0i}, and it's unique devation away from the mean slope, υ_{1i}, plus the unique deviation of each month's value away from the metro-specific trend line, ϵ_{ti}.

Exercise: I want you to extend the model that you wrote a little bit ago to now include two trends. I want you to simulate the data.