Econometrics Review

PCC Capacity Building [September 2019] **INTERNAL USE ONLY**

Gabriel Domingo

Contents

Chapter 1

Introduction

This document is to document the econometrics used by the Economics Office in some of its major merger, enforcement and adjudication cases. Reading this document will help new employees get up to speed with the econometrics useful at PCC. For the experienced analyst, it might help in identifying what further techniques or research would be useful to *EconO* in the future. This document is not intended to replace textbooks; for further clarifications or elucidation for techniques that extend the discussion here, please see (Finkelstein and Levin, 2015) and (Kleiber and Zeileis, 2008) among other texts.

This document would also include examples from merger or enforcement cases from 2017 to early 2019. These are the *Grab* acquisition , the *Udenna* acquisition, and the Rice PIER. This is by no means an exhaustive list – there are other cases, but these ones are illustrative of what has been done. Also, if you are interested in the actual cases, please see the original documents for the complete analysis.

The `Rmarkdown` code will be available to any employee who wants to tinker with the code herein. Any data that is included in the code is considered **confidential** and we ask all employees not to share the `Rmarkdown` and associated files with anyone outside of the Economics Office.

Chapter 2

Basic Probability and Statistics

Chapter 2's heading is somewhat misleading as we are **not** going to go through basic probability and statistics. However, we will focus on a few select techniques used in our work.

2.1 Tests on Proportions

To understand this concept, we first comes from the binomial distribution. A survey or test of n respondents, and each has a P probability of a "success" (however defined – shooting a basketball,changing one's purchase given a SSNIP question, etc.) of $y = 1$. Each respondent answers independently of each other. We can write this information parsimoniously as $y \sim \text{Binomial}(n, P)$.

We can show that $E(y) = P$, which means that the average value of a draw from the binomial distribution is equal to P, the population value for success. We can define the sample estimate to be $p = \frac{\sum y}{n}$. Its variance is $P(1 - P)$.

2.1.1 Confidence Intervals on Proportions

A confidence interval for p is a range of values around the proportion observed in a sample with the property that no value in the interval would be considered unacceptable as a possible value for P in light of the sample data. Again, consider that experiment with n (large value) independent observations with Y successes and $n - Y$ failures. The sample estimate of P is $p = \frac{Y}{n}$ and its standard error is $\sqrt{p(1-p)/n}$. With a large sample size, we can invoke the normal distribution.

Let us use `R` to calculate an confidence interval where $n = 1000$, and there are 700 sucesses:

```
n<-1000
est<-700/n
se<-sqrt(est*(1-est)/n)
int.95<-est+qnorm(c(.025,.975))*se
int.95
## [1] 0.6715974 0.7284026
```

We plot the interval below in Figure 2.1. The red areas are the lower 2.5% and the upper 2.5%.

2.1.1.1 Marcos Victims' Damages

The following comes from (Finkelstein and Levin, 2015), page 185.

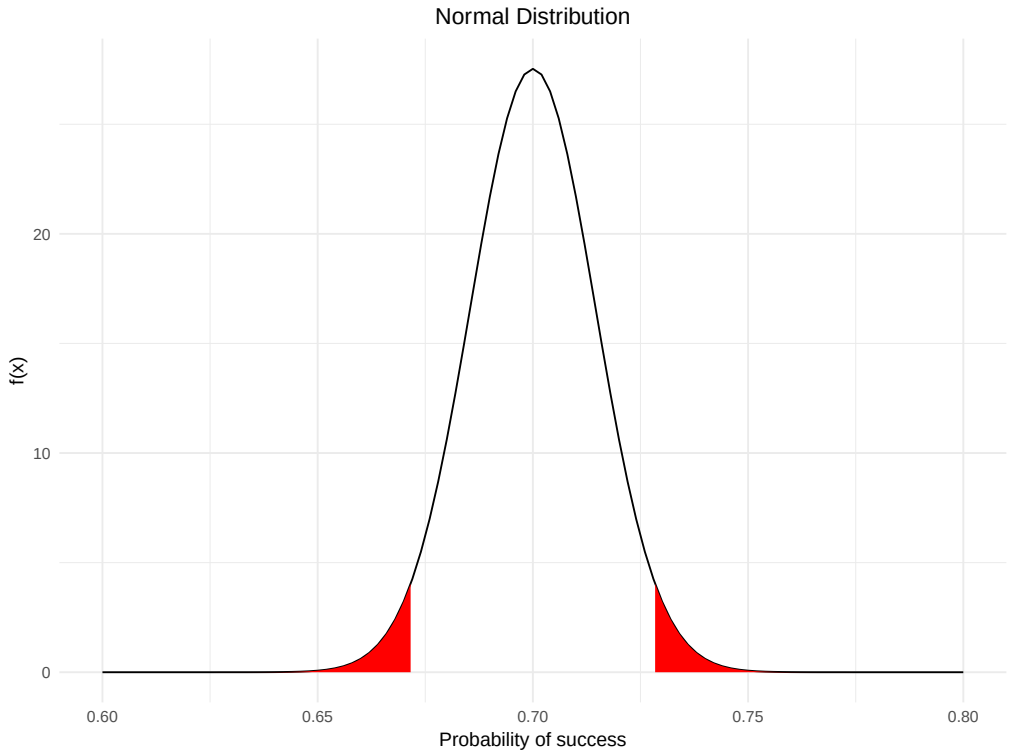


Figure 2.1: Normal Distribution to Approximate the Binomial Distribution

Table 2.1: Marcos Victims' Damage Claims Data

Statistic	Torture	Summary Execution	Disappearance
Number of valid claims in sample	64	50	17
Sample Average (Pesos)	51,719	128,515	107,853
Sample Standard Deviation (Pesos)	26,174	34,143	43,104
No. of Valid Claims	4,869	3,184	880
Subclass Amounts (Pesos)	251,819,811	409,191,760	94,910,640

Table 2.1 has the statistics for the three types of damages and claims of the Marcos Human rights Victims.[1] In the trial, the statistical expert hired by the court that instead of examining the 9,541 claims, it is enough to take random sample of 137. The expert testified that 137 claims would be enough to have a 95% confidence interval (for the % of valid claims) of plus or minus 5%. He further said, the share of valid claims [among the total claimns] is 90%. Is the expert correct to say 137 is enough?

2.1.2 SSNIP Results

When a SSNIP question is asked in a market where market definition is difficult. A SSNIP question has the same form as a proportion, where we can define proportion of success as the share of people who stayed despite a 10% price increase. A PCC procured survey of 800 TNVS users showed that given a 10% price increase on App-ATS car services, 88.7% of trips would still be retained under the same service, where the actual parameter lies within the range of 86.56% to 90.52% at 95% level of confidence.[2]

[1] *Hilao v. Estate of Marcos* 103 F.3d 768 (9th Circuit, 1996)
[2] M-2018-12, par 90.1

2.2 Tests on Means

We frequently calculate means, and we often wish to draw confidence intervals around them. The formula is the same, but this time we use the t-test if the sample is small [with degrees of freedom of $n-1$], and use the standard normal is if it large enough (typically >30). Further, we can use the `sd` command to calculate the sample standard eviation, and calculate the standard error as $se = sd/\sqrt{n}$.

2.2.1 Using the Rice Price Monitoring data: difference between two means

Using the *rice price monitoring* data, we extract a date (Feb 15, 2017), and compare the average of the average price of wmr and rmr over 5 stalls. We use the t test to see if there is a difference in the means.

```
ricedat<-data.frame(price=c(36.10000,37.12000,38.70000,37.400,
                  36.38460,30.720, 32.11429,33.65714,32.70000, 32.44),
      type=c(rep("wmr",5),rep("rmr",5)))
wmr<-mean(ricedat[1:5,1])
rmr<-mean(ricedat[6:10,1])
wmr.se<-sd(ricedat[1:5,1])/sqrt(5)
rmr.se<-sd(ricedat[6:10,1])/sqrt(5)
wmr-rmr
## [1] 4.814634
(wmr-rmr)+qt(c(0.025,.975),4)*(sqrt((wmr.se^2+rmr.se^2)))
## [1] 2.983160 6.646108
```

The confidence interval does not include zero, we say that at this date, the estimated difference of well milled rice and regular milled rice is 4.81 pesos with a standard error of 0.66 pesos.

2.2.2 Paired Data t-test

Occassionally, we will be following a panel of consumers or businesses over time, or over related decisions. If so, their responses will be positively correlated and then we cannot use the independent t-test.

2.2.2.1 Construct Dataset to simulate a paired t-test

We can construct a dataset to simulate paired data. Below, the code is designed so that $x2$ is positively correlated to $x1$, and its difference has a mean of 1.

```
set.seed(1000)
x1<-rnorm(10)
x2<-x1+rnorm(10,mean = 1,sd=.5)
```

2.2.2.2 Paired T-Test

We designed the data to have a difference of 1.

```
t.test(x1,x2)
##
##  Welch Two Sample t-test
##
## data:  x1 and x2
## t = -2.301, df = 17.561, p-value = 0.03388
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -1.49768572 -0.06673309
## sample estimates:
##  mean of x  mean of y
## -0.3290904  0.4531190
t.test(x1,x2,paired = T)
##
##  Paired t-test
##
## data:  x1 and x2
## t = -6.3415, df = 9, p-value = 0.0001342
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.0612393 -0.5031795
## sample estimates:
## mean of the differences
##              -0.7822094
```

The paired t-test finds very strong evidence that there is a difference in means. In this case, the independent test shows significance at 5% level, but we also see that the independent samples t-test is underpowered. There is a chance it will not detect a difference even if there is one.

2.3 Tests on Variance

We frequently have to check if the variances are different. Normally, we do this over time – during, say a suspected collusive period vs. a non-collusive period. Here we can also look at *actual* variances and compare them to each other. To test hypotheses about variances, we use χ^2 and the F distributions. We can build some data, and calcluate if the variances are different.

```
set.seed(100)
x <- rnorm(50, mean = 0, sd = 2)
y <- rnorm(30, mean = 1, sd = 1)
var.test(x, y,alternative = "greater")          # Do x and y have the same variance?
##
##  F test to compare two variances
##
## data:  x and y
## F = 1.7382, num df = 49, denom df = 29, p-value = 0.05681
## alternative hypothesis: true ratio of variances is greater than 1
## 95 percent confidence interval:
##  0.977951      Inf
## sample estimates:
## ratio of variances
##           1.738198
```

With the number of observations this small, there is some, but not overwhelimg, evidence that x has a higher variance.

Chapter 3

Linear Regression

Now, we go to the staple for linear regression. I won't bore anyone with the basics, or with the topics one which is typically the subject matter of the first course in Econometrics.

$$y_i = x_i^T \beta + \epsilon_i, \quad i = 1, ..., n$$

In R, it is straighforward to run a linear regression. We demonstrate that here, in the dataset from (Finkelstein and Levin, 2015) on explaining airline's costs. The regression output for R is similar. It uses a `forumla` notation, with the dependent variable on the left of `~` and the independent variables on the right side, separated by `+`. The `+` is not addition; it is to add variables to the model. We can use `I()` to create numerical transforms, such as quadratic terms.

```
require(tidyverse)
AirlineCost<-tibble("Year"=1998:2007,
            "Military Revenue ($ Millions)"=c(128.5,114,185.4,159.5,176.2,
                  276.4,302.7,406.1,342.5,276.9),
            "Military Cost ($ Millions)"=c(113.9,106.9,175,155.8,177.1,
                  257.5,268.4,315.1,306.7,274.8))

reg1<-lm(`Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)`,
         data=AirlineCost)
summary(reg1)
##
## Call:
## lm(formula = `Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)`,
##     data = AirlineCost)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.979  -9.368   1.439   9.226  28.436
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    30.51119   15.52697   1.965    0.085 .
## `Military Revenue ($ Millions)`  0.77953    0.06103  12.773 1.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.94 on 8 degrees of freedom
```

```
## Multiple R-squared:  0.9533, Adjusted R-squared:  0.9474
## F-statistic: 163.2 on 1 and 8 DF,  p-value: 1.33e-06
```

We observe the cofficients 0.78 has a t value of 12.773 [ratio of estimate and the standard error], a p-value of approximately zero, and is highly statistically significant. Its R^2 is 0.953. The F-statistic allows us to evaluate the significance of the entire regression. In this case, as it is just one regressor, the F stat is just the square of the t-stat.

```
anova(reg1)
## Analysis of Variance Table
##
## Response: Military Cost ($ Millions)
##                                Df Sum Sq Mean Sq F value   Pr(>F)
## `Military Revenue ($ Millions)`  1  52535   52535  163.15 1.33e-06 ***
## Residuals                        8   2576     322
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

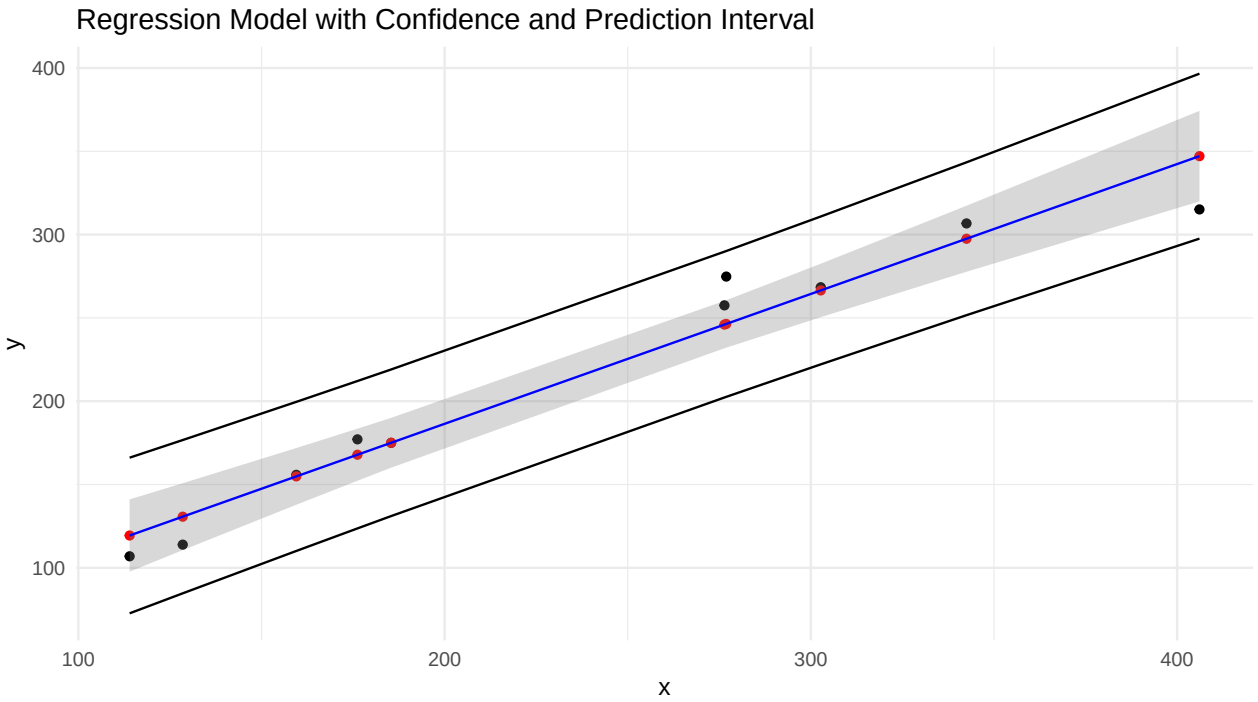
We can also ask for confidence intervals for the regressors:

```
confint(reg1)
##                                    2.5 %     97.5 %
## (Intercept)                    -5.2940688 66.3164512
## `Military Revenue ($ Millions)`  0.6387986  0.9202657
```

Let us plot the regression estimates vs the real data. In the plot below, we plot both the prediction interval and the confidence interval. A prediction interval is the prediction for a specific y, while the confidence interval is a statement about the location of the regression line itself, or $E[y|x]$.

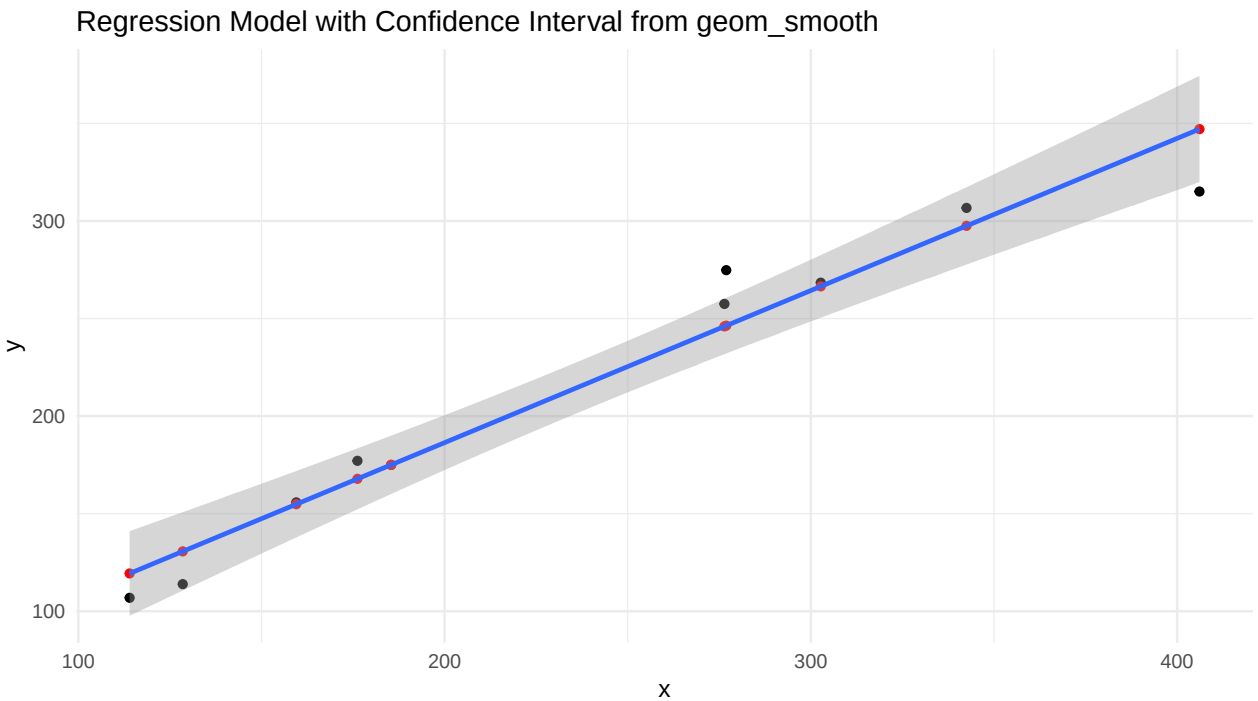
```
plotz<-cbind(as.tibble(predict(reg1,interval = "confidence")),
             x=AirlineCost$`Military Revenue ($ Millions)`,
             y=AirlineCost$`Military Cost ($ Millions)`,
             Year=AirlineCost$Year)
plotz.p<-cbind(as.tibble(predict(reg1,interval = "prediction")),
               x=AirlineCost$`Military Revenue ($ Millions)`,
               Year=AirlineCost$Year)

ggplot(plotz)+geom_point(aes(x=x,y=y))+geom_point(aes(x=x,y=fit),color="red")+
  geom_ribbon(aes(x=x,ymin=lwr,ymax=upr),fill="gray50",alpha=.3)+
  geom_line(data=plotz.p,aes(x=x,y=fit),color="blue")+
  geom_line(data=plotz.p,aes(x=x,y=upr))+geom_line(data=plotz.p,aes(x=x,y=lwr))+
  theme_minimal()+ggtitle("Regression Model with Confidence and Prediction Interval")
```



Using ggplot, we can do it "automatically", with `geom_smooth`.

```
ggplot(plotz)+geom_point(aes(x=x,y=y))+geom_point(aes(x=x,y=fit),color="red")+
  theme_minimal()+geom_smooth(aes(x=x,y=y),method="lm")+
  ggtitle("Regression Model with Confidence Interval from geom_smooth")
```



All of the above carries over in the *multiple regression* context, as we will see. We include a time trend variable below. Is this model better than the simpler model?

```
reg2<-lm(`Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)`+I(Year-1997),
         data=AirlineCost)
```

```
summary(reg2)
##
## Call:
## lm(formula = `Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)` +
##     I(Year - 1997), data = AirlineCost)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.980  -3.741  -1.326   3.775  16.506
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    37.41984    8.99980   4.158  0.00425 **
## `Military Revenue ($ Millions)`  0.53845    0.06712   8.022 8.96e-05 ***
## I(Year - 1997)                  9.12465    2.17282   4.199  0.00404 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.23 on 7 degrees of freedom
## Multiple R-squared:  0.9867, Adjusted R-squared:  0.9829
## F-statistic:   260 on 2 and 7 DF,  p-value: 2.7e-07
```

```
anova(reg1,reg2)
## Analysis of Variance Table
##
## Model 1: `Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)`
## Model 2: `Military Cost ($ Millions)` ~ `Military Revenue ($ Millions)` +
##     I(Year - 1997)
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1      8 2575.97
## 2      7  731.95  1      1844 17.635 0.004038 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can look at the R^2, and the adjusted R^2 is higher 0.9867185 than `reg1` at 0.953258. The F-test for the difference between the two models is also significant, which is the same result as the t-test. We can also look at the AIC, which is $-2 * LogLike + k * df$. An AIC closer to zero means the log likelihood is is better. The AIC of reg1 is 89.8927505 and the AIC of reg2 is 79.3100472.

3.1 Link between Regression and Classical T-Tests/ANOVA

We can evaluate difference in means using a regression framework as well. This is likely to be a preferred way to determine differences between mean values between groups, because the regression framework is more general than ANOVA. In the example below, there are two values for ethnicity: "cauc" and "afam". The coefficient for ethnicity gives the mean wage difference between these two values.

```
require(AER)
data("CPS1988")

cps_lm<-lm(wage~ethnicity,data=CPS1988)
summary(cps_lm)
##
## Call:
```

```
## lm(formula = wage ~ ethnicity, data = CPS1988)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -567.2  -284.9   -76.5   190.0 18160.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    617.234      2.802  220.25   <2e-16 ***
## ethnicityafam -170.381      9.953  -17.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 451.2 on 28153 degrees of freedom
## Multiple R-squared:  0.0103, Adjusted R-squared:  0.01027
## F-statistic:   293 on 1 and 28153 DF,  p-value: < 2.2e-16
```

This is also the same test as the ANOVA, where all the coefficients are equal to zero (and the means are the same).The F-stat is the ratio of the Mean Squared error explained by our regressor divided by the mean squared error unexplained by the model (the residuals).

```
anova(cps_lm)
## Analysis of Variance Table
##
## Response: wage
##              Df     Sum Sq  Mean Sq F value    Pr(>F)
## ethnicity     1   59657902 59657902  293.02 < 2.2e-16 ***
## Residuals 28153 5731766262   203593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The t-test in the regression is also the same as the independent samples t-test from earlier, with equal variances.

```
caus<-subset(CPS1988,ethnicity=="cauc")$wage
afam<-subset(CPS1988,ethnicity=="afam")$wage
t.test(caus,afam,alternative = "two.sided",var.equal = T) # it is the same as in regression/ANOVA
##
##  Two Sample t-test
##
## data:  caus and afam
## t = 17.118, df = 28153, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  150.8723 189.8904
## sample estimates:
## mean of x mean of y
##  617.2339  446.8526
```

3.1.1 Paired T-Tests are linked to Linear Mixed Effects/Random Effects Models

The random effects model will generate subject specific intercepts, but these are drawn from $N(\mu_{subj}, \sigma_{subj})$.

```
# Now create a dataframe for lme
myDat <- data.frame(c(x1,x2), c(rep("x1", 10), rep("x2", 10)), rep(paste("S", seq(1,10), sep=""), 2))
names(myDat) <- c("y", "x", "subj")
head(myDat)
##             y  x subj
## 1 -0.44577826 x1   S1
## 2 -1.20585657 x1   S2
## 3  0.04112631 x1   S3
## 4  0.63938841 x1   S4
## 5 -0.78655436 x1   S5
## 6 -0.38548930 x1   S6

require(nlme) # nlme has options for the covariance matrix
summary(fm1 <- lme(y ~ x, random=~1 | subj, data=myDat))
## Linear mixed-effects model fit by REML
##  Data: myDat
##        AIC      BIC    logLik
##   41.19113 44.75261 -16.59556
##
## Random effects:
##  Formula: ~1 | subj
##         (Intercept)  Residual
## StdDev:   0.7083312 0.2758121
##
## Fixed effects: y ~ x
##                  Value Std.Error DF   t-value p-value
## (Intercept) -0.3290904 0.2403759  9 -1.369066  0.2042
## xx2          0.7822094 0.1233469  9  6.341539  0.0001
##  Correlation:
##     (Intr)
## xx2 -0.257
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.48378444 -0.52730267 -0.06887495  0.55790488  1.43668919
##
## Number of Observations: 20
## Number of Groups: 10
```

I note that the the coefficients of xx2 is the difference between the treatments, and the p-value and t-stat is the same as the paired t-tests.

3.2 Interaction Terms

We return to multiple linear regression, and focus on how to estimate interaction terms. Using the ethnicity factor variable in a wage regression, we get the following results.

```
cps_lm<-lm(log(wage)~experience+I(experience^2)+education+ethnicity,data=CPS1988)
summary(cps_lm)
##
## Call:
## lm(formula = log(wage) ~ experience + I(experience^2) + education +
##     ethnicity, data = CPS1988)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9428 -0.3162  0.0580  0.3756  4.3830
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.321e+00  1.917e-02  225.38   <2e-16 ***
## experience       7.747e-02  8.800e-04   88.03   <2e-16 ***
## I(experience^2) -1.316e-03  1.899e-05  -69.31   <2e-16 ***
## education        8.567e-02  1.272e-03   67.34   <2e-16 ***
## ethnicityafam   -2.434e-01  1.292e-02  -18.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5839 on 28150 degrees of freedom
## Multiple R-squared:  0.3347, Adjusted R-squared:  0.3346
## F-statistic:  3541 on 4 and 28150 DF,  p-value: < 2.2e-16
```

We can add an interaction term for the factor variable ethnicity. The * operator in the forumla will include the direct effect and the interaction effect.

```
cps_lm<-lm(log(wage)~experience+I(experience^2)+education*ethnicity,
           data=CPS1988)
summary(cps_lm)
##
## Call:
## lm(formula = log(wage) ~ experience + I(experience^2) + education *
##     ethnicity, data = CPS1988)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9451 -0.3162  0.0578  0.3761  4.3929
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               4.313e+00  1.959e-02 220.170   <2e-16 ***
## experience                7.752e-02  8.803e-04  88.063   <2e-16 ***
## I(experience^2)          -1.318e-03  1.901e-05 -69.339   <2e-16 ***
## education                 8.631e-02  1.309e-03  65.944   <2e-16 ***
## ethnicityafam            -1.239e-01  5.903e-02  -2.099   0.0358 *
## education:ethnicityafam -9.648e-03  4.651e-03  -2.074   0.0380 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5839 on 28149 degrees of freedom
## Multiple R-squared:  0.3348, Adjusted R-squared:  0.3347
## F-statistic:  2834 on 5 and 28149 DF,  p-value: < 2.2e-16
```

We can use / to use nested coefficient coding. We can also use %in% to get differntiated slopes by category.

```
lm(log(wage)~experience+I(experience^2)+education / ethnicity,
   data=CPS1988)
##
## Call:
```

```
## lm(formula = log(wage) ~ experience + I(experience^2) + education/ethnicity,
##     data = CPS1988)
##
## Coefficients:
##             (Intercept)               experience          I(experience^2)
##                 4.30377                  0.07757                 -0.00132
##               education  education:ethnicityafam
##                 0.08699                 -0.01917

lm(log(wage)~experience+I(experience^2)+education %in% ethnicity,
   data=CPS1988)
##
## Call:
## lm(formula = log(wage) ~ experience + I(experience^2) + education %in%
##     ethnicity, data = CPS1988)
##
## Coefficients:
##             (Intercept)               experience          I(experience^2)
##                 4.30377                  0.07757                 -0.00132
## education:ethnicitycauc  education:ethnicityafam
##                 0.08699                  0.06781
```

3.3 Prediction

After estimating a relationship between prices (or whatever we are measuring pertinent to the case) to dependent variables (like costs), we might like to see how future values of dependent variables, given certain values for the control variables. This is particularly useful for merger cases to predict possible SLC, or for enforcement cases to estimate how much prices are elevated in the cartel period vis-a-vis the non-cartel period.

3.3.1 Airline Costs Predicted

We return to the example of Airline costs. In 2008, the revenue is 286.5. What is the prediction for y_{2008}, and the associated prediction window.

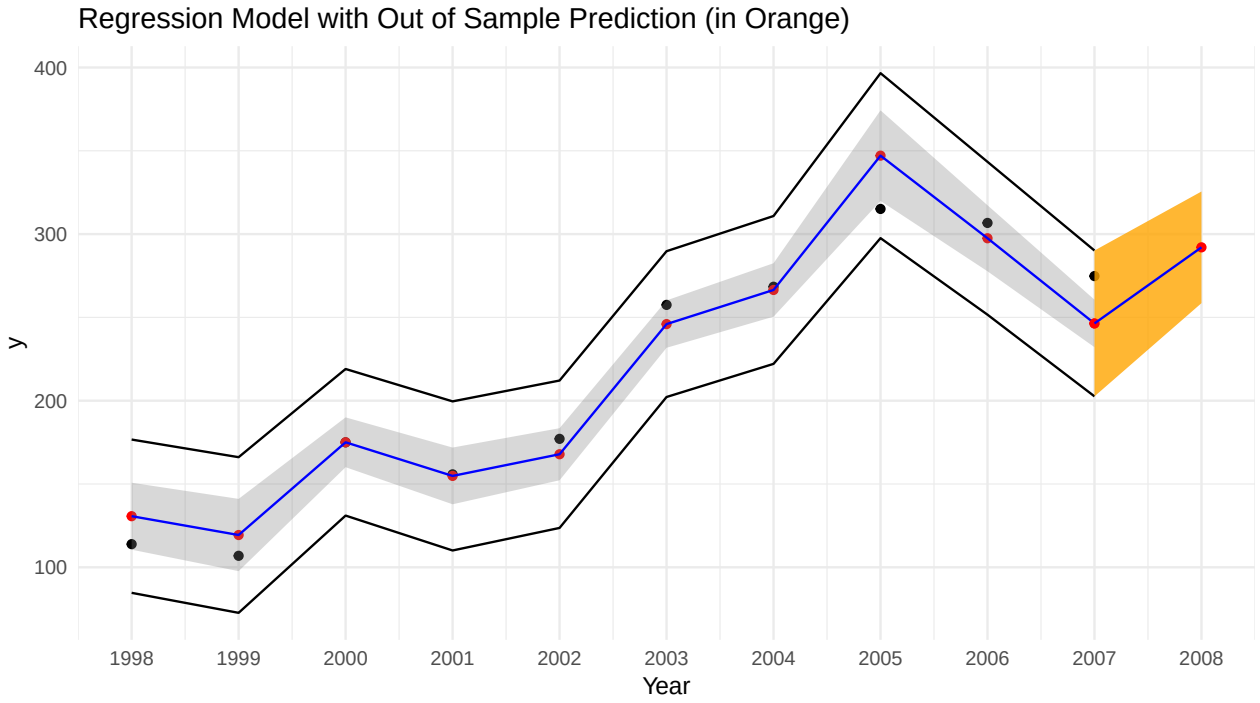
```
new.AirlineCost<-tibble(`Military Revenue ($ Millions)`=286.5,Year=2008)
predict(reg2,newdata = new.AirlineCost,interval="prediction")
##        fit      lwr      upr
## 1 292.0555 258.5489 325.5621
new.AirlineCost<-cbind(x=286.5,Year=2008,predict(reg2,newdata = new.AirlineCost,
                                                 interval="prediction"))
plotz.p2<-rbind(new.AirlineCost,plotz.p[10,])
```

We can plot it below:

```
plotz.p<-cbind(as.tibble(predict(reg1,interval = "prediction")),
               x=AirlineCost$`Military Revenue ($ Millions)`,
               Year=AirlineCost$Year)

ggplot(plotz)+geom_point(aes(x=Year,y=y))+geom_point(aes(x=Year,y=fit),color="red")+
  geom_ribbon(aes(x=Year,ymin=lwr,ymax=upr),fill="gray50",alpha=.3)+
  geom_line(data=plotz.p,aes(x=Year,y=fit),color="blue")+
```

```
geom_line(data=plotz.p,aes(x=Year,y=upr))+geom_line(data=plotz.p,aes(x=Year,y=lwr))+
geom_ribbon(data=plotz.p2,aes(x=Year,ymin=lwr,ymax=upr),fill="orange",alpha=.8)+
geom_point(data=plotz.p2,aes(x=Year,y=fit),color="red")+
geom_line(data=plotz.p2,aes(x=Year,y=fit),color="blue")+
theme_minimal()+ggtitle("Regression Model with Out of Sample Prediction (in Orange)")+
scale_x_continuous(labels=1998:2008,breaks=1998:2008)
```



3.3.2 Grab Fares Predicted

In the Grab case, we calculated daily average fares. We also calculated average daily values for the elements that go into a trip for a fare, such as distance and time, and the surge value for the trip. In Figure 3.1, we show the effect of the end of the interim measures on the predictions as against actual fares. Prior to the neutering of the interim mesures, the fare formula coefficients were able to predict daily averega fares accurately. After the end of the interim orders, the "old" coefficients became inaccurate as the algorithm changed. Actual fares (red line) became much higher than predicted fares (dots).

3.4 Regression Diagnostics

Once we have run the regression, we try to check if there is anything more we can do to assess the efficiency and consistency of our estimates. I will skip the usual discussion of endogeneity because it is rare for us to be overly concerned with dealing with endogenous estimates.[1]

Here, I focus on heteroskedasticity. Heteroskedasticity is simply the conditional variance of the errors is constant: $Var(\epsilon_i|x_i) \neq \sigma^2$. Here, we focus on testing for heteroskedasticity using the Journals dataset from the `AER` package.

[1] Except for demand estimation, which we conspicuously defer our discussion to a later date. We also have not incorporated demand estimation in our reports and analyses, for want of data.

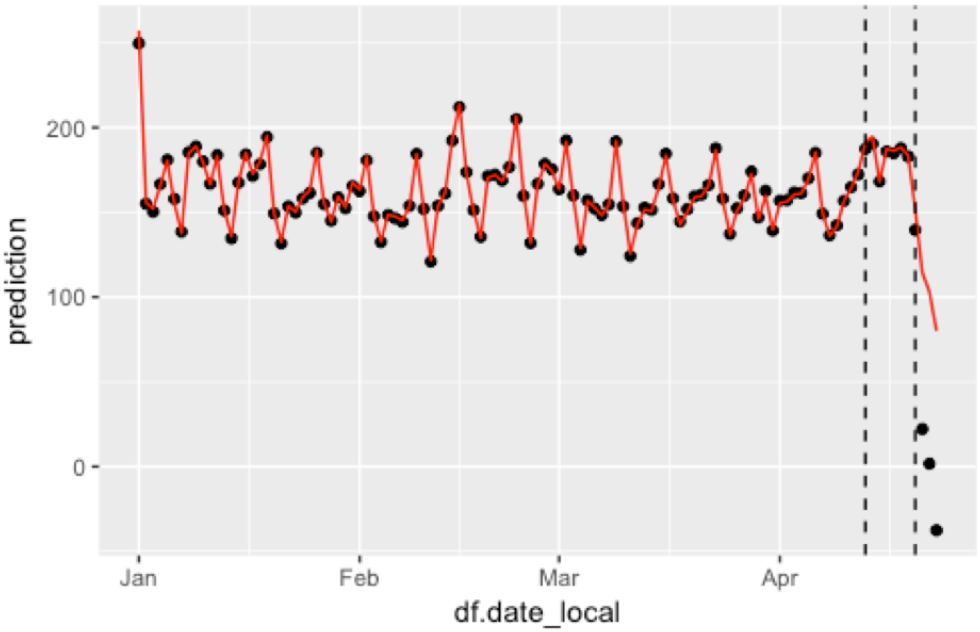


Figure 3.1: Actual Grab Average Daily Fare vs Predictions

```
require(lmtest);require(AER)
data("Journals")
journals<-Journals[c("subs","price")]
journals$citeprice<-Journals$price/Journals$citations
journals$age<-2000-Journals$foundingyear

jour_lm<-lm(log(subs)~log(citeprice)+log(age),data=journals)
bptest(jour_lm)
##
##  studentized Breusch-Pagan test
##
## data:  jour_lm
## BP = 13.605, df = 2, p-value = 0.001111
```

We see that the Breush-Pagan test rejects the null hypothesis of no heterskedasticity. The test is based on whether regressions of the residuals against the functions of the control variables.

3.4.1 Sandwich Estimators

If heteroskedasticity is a problem, a simpler and more practical approach is to to use sandwich estimators to correct inferences for regression coefficients. This is in the `robust` option in *Stata*. In `R`, we use *sandwich* or *clubSandwich* packages.

The sandwich estimator replaces the standard variance covariance matrix. While this is important in all models, it is particularly important in panel data for "clustered" standard errors, which we discuss later. The variance-coveriance matrix types are either HC "Heteroskedasticity Consistent" or HAC "Heteroskedasticity and Autocorrelation Consistent". There are different varieties of these HC/HAC matrices. To reproduce the `robust` option, we need to use the "HC1" variance matrix in Sandwich.

```
require(sandwich);require(lmtest)
coeftest(jour_lm,vcov=vcovHC,type="const") #same as usual regression
##
## t test of coefficients:
##
##                Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)     3.39641    0.30028  11.3109 < 2.2e-16 ***
## log(citeprice) -0.43425    0.03990 -10.8833 < 2.2e-16 ***
## log(age)        0.41872    0.09035   4.6344 6.923e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
coeftest(jour_lm,vcov=vcovHC,type="HC1") #Stata Robust
##
## t test of coefficients:
##
##                 Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)     3.396412   0.383850   8.8483 8.944e-16 ***
## log(citeprice) -0.434246   0.042543 -10.2071 < 2.2e-16 ***
## log(age)        0.418716   0.120928   3.4625  0.000671 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
coeftest(jour_lm,vcov=vcovHC) #Sandwich's usual HC3
##
## t test of coefficients:
##
##                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)     3.396412   0.398463  8.5238 6.618e-15 ***
## log(citeprice) -0.434246   0.043999 -9.8694 < 2.2e-16 ***
## log(age)        0.418716   0.125626  3.3330  0.001046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the journal dataset, we can see that the standard errors rise. Standard standard errors under linear regression are smaller than they ought to be if the errors are negatively correlated with extreme values of x. [see (Angrist and Pischke, 2010)]

Chapter 4

Time Series

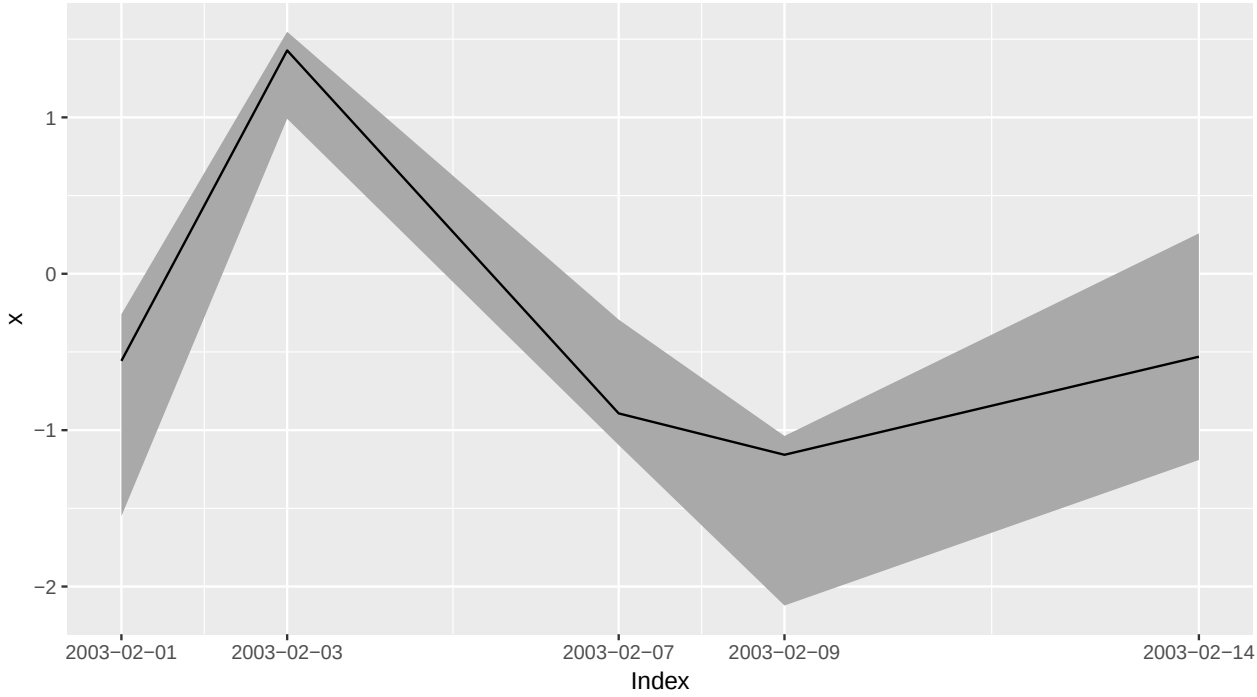
Frequently, we deal with partions in time, testing if a specific duration is "collusive"; or – in market definition – whether prices over time "move together". In R, we can work with time-indexed data with the help of a few packages. The original tool (in the base specification) is ts, and more recently, user have developed the zoo and tsibble packages. This is an active field of development, as R was not primarily used for time series. The ts uses a matrix, which is slightly different from the data.frame we normally use. Also, the time index is not easily used to subset the matrix. Let us create a simulated time series dataset below:

```
x<-rnorm(n=10,mean=1,sd=2)
ts(x,deltat=1/12) #monthly
##          Jan        Feb        Mar        Apr        May        Jun
## 1  2.7936445  0.9000085 -1.6906986 -2.8624231  2.4191632  0.6841899
##          Jul        Aug        Sep        Oct
## 1  1.4327357  2.6347242  4.4543515  0.7924594
ts(x,start=c(2007,3),frequency = 12) #monthly, starts at March 2007
##             Mar        Apr        May        Jun        Jul        Aug
## 2007  2.7936445  0.9000085 -1.6906986 -2.8624231  2.4191632  0.6841899
##             Sep        Oct        Nov        Dec
## 2007  1.4327357  2.6347242  4.4543515  0.7924594
```

On the other hand, we can use zoo, which still technically a matrix, but can be more easily used to subset via dates. It also allows irregularly spaced series. We illustrate that below:

```
x.Date <- as.Date(paste(2003, 02, c(1, 3, 7, 9, 14), sep = "-"))
x <- zoo(rnorm(5), x.Date)
xlow <- x - runif(5)
xhigh <- x + runif(5)
z <- cbind(x, xlow, xhigh)

ggplot(aes(x = Index, y = x, ymin = xlow, ymax = xhigh), data = fortify(x)) +
  geom_ribbon(fill = "darkgray") + geom_line()+
  scale_x_date(breaks=index(z)) #adding the upper and lower bands
```




```
#use index(.) to extract the line date

# Subsetting ala dplyr
fortify(x) %>% filter(Index<as.Date("2003-2-10"))
##        Index          x
## 1 2003-02-01 -0.5571223
## 2 2003-02-03  1.4283014
## 3 2003-02-07 -0.8929574
## 4 2003-02-09 -1.1575712
```

We are skipping many time series topics that would be of interest to a competition authority. Forecasting a time series would require, at least, ARIMA or SARIMA time series modelling. I defer such a discussion to a later document, and instead will focus on Stationarity and Cointegration here.[1]

4.1 Correlation

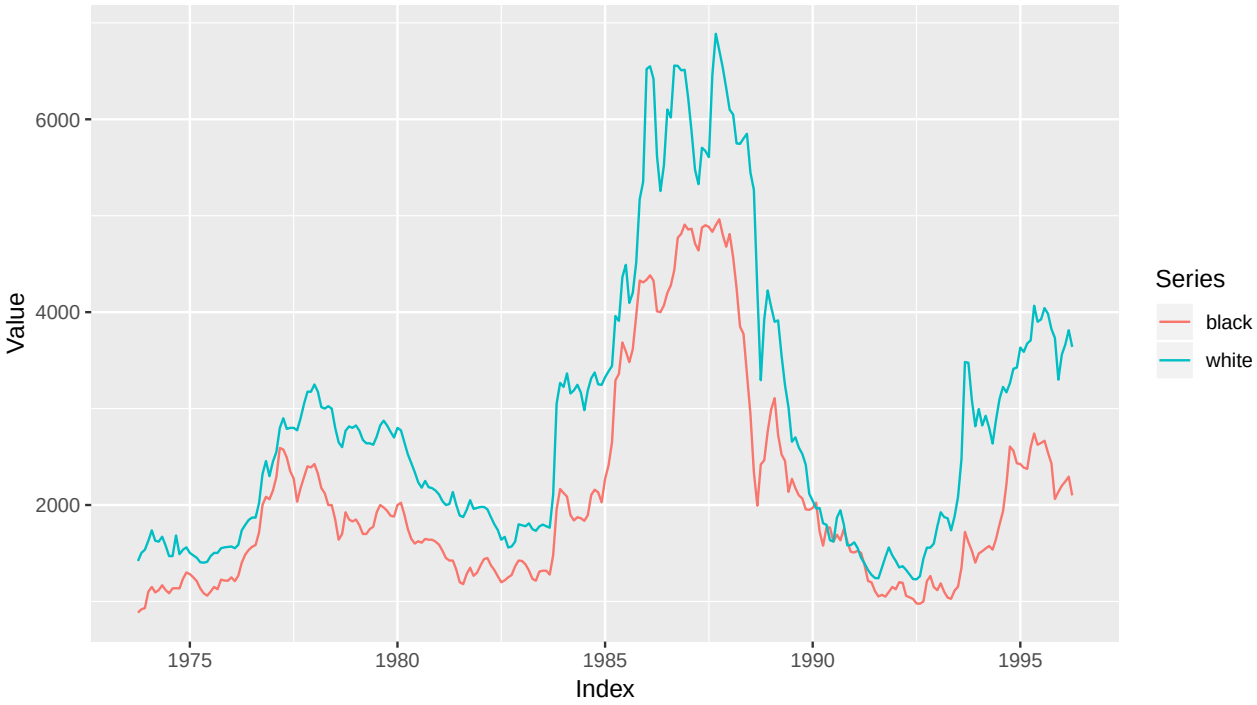
One first thing that used to be done is to calculate the correlations between two time series. A high correlation is evidence for market integration (or a single product market). We use the `PepperPrice` dataset from AER.

```
data("PepperPrice")
peprice<-as.zoo(PepperPrice)
cor.test(peprice$black,peprice$white)
##
##  Pearson's product-moment correlation
##
## data:  peprice$black and peprice$white
## t = 51.203, df = 269, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

[1] Even here, I discuss only the most basic issues on Stationarity and Cointegration.

```
##  0.9398282 0.9622938
## sample estimates:
##       cor
## 0.9523357
```

```
ggplot(fortify(peprice,melt=T))+geom_line(aes(x=Index,y=Value,color=Series))
```



4.2 Unit Root

Somewhat more recently, econometricians have realized that inference given autocorrelation in time series is problemtic. It is first recommended to determine if a series is stationary. In simple terms, a stationary time series is when the mean of the series is constant, allowing for linear trends. If a series is stationary, it can be used in a regression model. Here I demonstrate the dickey-fuller unit root test. The null hypothesis is non-stationary, and in the white pepper series, the null cannot be rejected. We use the `tseries` and `urca` package's augmented dickey fuller tests below.

```
require(tseries);require(urca)
adf.test(log(peprice$white))
##
##  Augmented Dickey-Fuller Test
##
## data:  log(peprice$white)
## Dickey-Fuller = -1.744, Lag order = 6, p-value = 0.6838
## alternative hypothesis: stationary
ur.df(log(peprice$white),type="trend",lags=6,selectlags = "Fixed") %>% summary()
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
```

```
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.202083 -0.034258 -0.006182  0.027274  0.304325
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.264e-01  7.110e-02   1.778   0.0766 .
## z.lag.1     -1.606e-02  9.210e-03  -1.744   0.0824 .
## tt           1.291e-05  5.383e-05   0.240   0.8107
## z.diff.lag1  3.423e-01  6.220e-02   5.502 9.12e-08 ***
## z.diff.lag2 -9.766e-02  6.565e-02  -1.488   0.1381
## z.diff.lag3  3.469e-03  6.581e-02   0.053   0.9580
## z.diff.lag4  2.660e-02  6.617e-02   0.402   0.6881
## z.diff.lag5 -2.040e-02  6.642e-02  -0.307   0.7590
## z.diff.lag6  9.123e-02  6.282e-02   1.452   0.1476
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06478 on 255 degrees of freedom
## Multiple R-squared:  0.1196, Adjusted R-squared:  0.092
## F-statistic: 4.331 on 8 and 255 DF,  p-value: 6.517e-05
##
##
## Value of test-statistic is: -1.744 1.1089 1.5356
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.98 -3.42 -3.13
## phi2  6.15  4.71  4.05
## phi3  8.34  6.30  5.36
```

Above, i used two adf tests, and both of then accepted the Null of Non-stationarity (and they have the same test stat of -1.74). The `urca` test allows the analyst to change the assumed process, which above is a process with a trend. We change the `type` settings below:

```
require(urca)
ur.df(log(peprice$white),type="none") # accept the NULL, see critical values add %>% summary()
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: 0.4478
ur.df(log(peprice$white),type="drift") #accept the NULL
##
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
```

```
##
## The value of the test statistic is: -1.7855 1.7474
```

Alternately, we can use the KPSS test. The null hypothesis of KPSS stationarity. At the 2.5% level below, we find that we reject the null hypothesis.

```
ur.kpss(log(peprice$white)) %>% summary()
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.6173
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

4.3 Cointegration

Given two non-stationary time series of the same level of integration, we can then test if these are cointegrated. That is, if these two have a "long-run" relationship.[2] We can use the `urca` package here, with the johansen vector cointegration test.

```
require(urca)
pepper_jo<-ca.jo(log(peprice),ecdet = "const",type="trace",spec="longrun")
summary(pepper_jo)
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1] 4.931953e-02 1.350807e-02 2.081668e-17
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  3.66  7.52  9.24 12.97
## r = 0  | 17.26 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##            black.l2 white.l2   constant
## black.l2  1.0000000  1.00000   1.000000
## white.l2 -0.8892307 -5.09942   2.280911
```

[2]The other thing we can do is to rewrite the conintegration as an *error correction model*, which we defer to a later document. As of September 2019, the PCC has not done a error correction model related to a enforcement or merger case.

```
## constant -0.5569943 33.02742 -20.032441
##
## Weights W:
## (This is the loading matrix)
##
##            black.l2    white.l2      constant
## black.d -0.07472300 0.002453210 -4.958157e-18
## white.d  0.02015611 0.003537005  8.850353e-18
```

The most important part of the output is the table of r, which is the count of the number of cointegrating relationships. We see above that the test stat for the null of "r=0" is less than the critical values. Therefore, the null cannot be rejected that there is *no cointegrating vector* between the variables. If it there is no cointegrating relationship, then we should difference the I(1) time series, and it would become stationary and lend itself to regression analysis.[3]

Another way to move forward is to get the ratio of the two non-stationary series, and test the stationarity of the ratio (for the simplest case only). We do this below:

```
ppricerat<-log(peprice$black)/log(peprice$white)

ur.df(ppricerat) %>% summary()
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.023078 -0.004062 -0.000102  0.003696  0.051984
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -3.339e-05  5.052e-04  -0.066    0.947
## z.diff.lag -1.370e-03  6.125e-02  -0.022    0.982
##
## Residual standard error: 0.007952 on 267 degrees of freedom
## Multiple R-squared:  1.83e-05,   Adjusted R-squared:  -0.007472
## F-statistic: 0.002443 on 2 and 267 DF,  p-value: 0.9976
##
##
## Value of test-statistic is: -0.0661
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

When we test the augmented dickey fuller, we find that the test statistic is less than the critical values, and we accept the null of non-stationarity.

[3] An assumption for the Johansen test is that the series in the test are all I(1) or I(0).

4.3.1 Rice Cointegration Test

In the PIER for the rice investigation, we looked at whether retail prices and wholesale prices for well-milled and regular milled rice move together. The analyst uses the Johansen vector test.

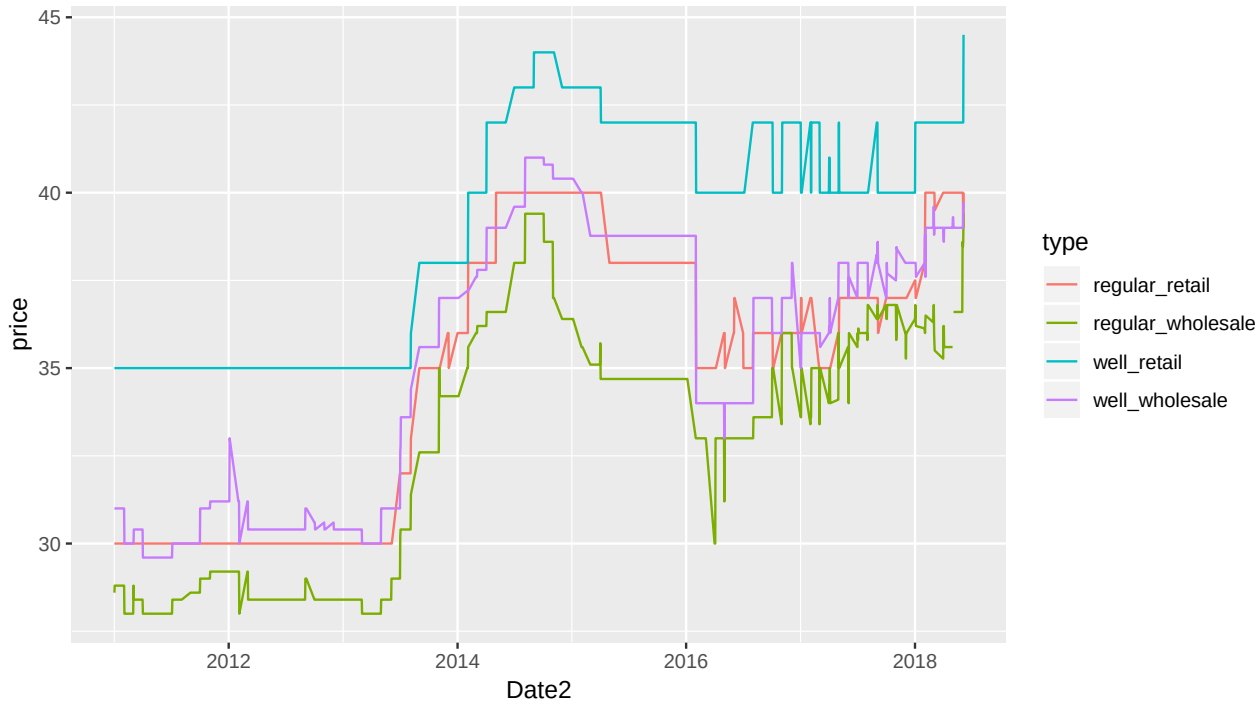
```
rice<-read_csv("data/rice_mla.csv")
require(lubridate)

#focus on regular
require(reshape2)
margins<-dcast(rice, Month + Year + Week + Date2 ~ Type + Chain, value.var = "Price")
margins$Date2<-as.Date(margins$Date2)

margins<-margins[order(margins$Date2),]

margins_long<-gather(margins,key="type",value="price",-Month,-Year,-Week,-Date2)

ggplot(data=margins_long,aes(x=Date2,y=price,color=type))+geom_line()
```




```
library(urca)
coRes=ca.jo(margins[,5:6],type="trace",K=2,ecdet="const", spec="longrun") #regular
summary(coRes)
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1] 7.073584e-02 7.101520e-03 8.673617e-18
##
```

```
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  2.77  7.52  9.24 12.97
## r = 0  | 31.23 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                      regular_retail.l2 regular_wholesale.l2    constant
## regular_retail.l2             1.000000             1.000000   1.0000000
## regular_wholesale.l2         -1.102829            -3.118234   0.7226263
## constant                      1.233920            86.993768 -56.3802899
##
## Weights W:
## (This is the loading matrix)
##
##                     regular_retail.l2 regular_wholesale.l2      constant
## regular_retail.d          -0.07597946         0.0009044288  2.868132e-16
## regular_wholesale.d        0.06607511         0.0017555885 -2.925149e-16
```

```
coRes=ca.jo(margins[,7:8],type="trace",K=2,ecdet="const", spec="longrun") #well milled
summary(coRes)
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , without linear trend and constant in cointegration
##
## Eigenvalues (lambda):
## [1]  4.612635e-02  5.790667e-03 -1.157674e-17
##
## Values of teststatistic and critical values of test:
##
##           test 10pct  5pct  1pct
## r <= 1 |  2.26  7.52  9.24 12.97
## r = 0  | 20.68 17.85 19.96 24.60
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                    well_retail.l2 well_wholesale.l2     constant
## well_retail.l2          1.0000000           1.00000   1.00000000
## well_wholesale.l2      -0.8577441          39.93588   0.07150267
## constant               -8.9667223       -1608.15583 -38.40384628
##
## Weights W:
## (This is the loading matrix)
##
##                   well_retail.l2 well_wholesale.l2      constant
## well_retail.d       -0.08135585     -5.925424e-05  5.212857e-16
## well_wholesale.d     0.01645154     -1.629211e-04 -1.049180e-16
```

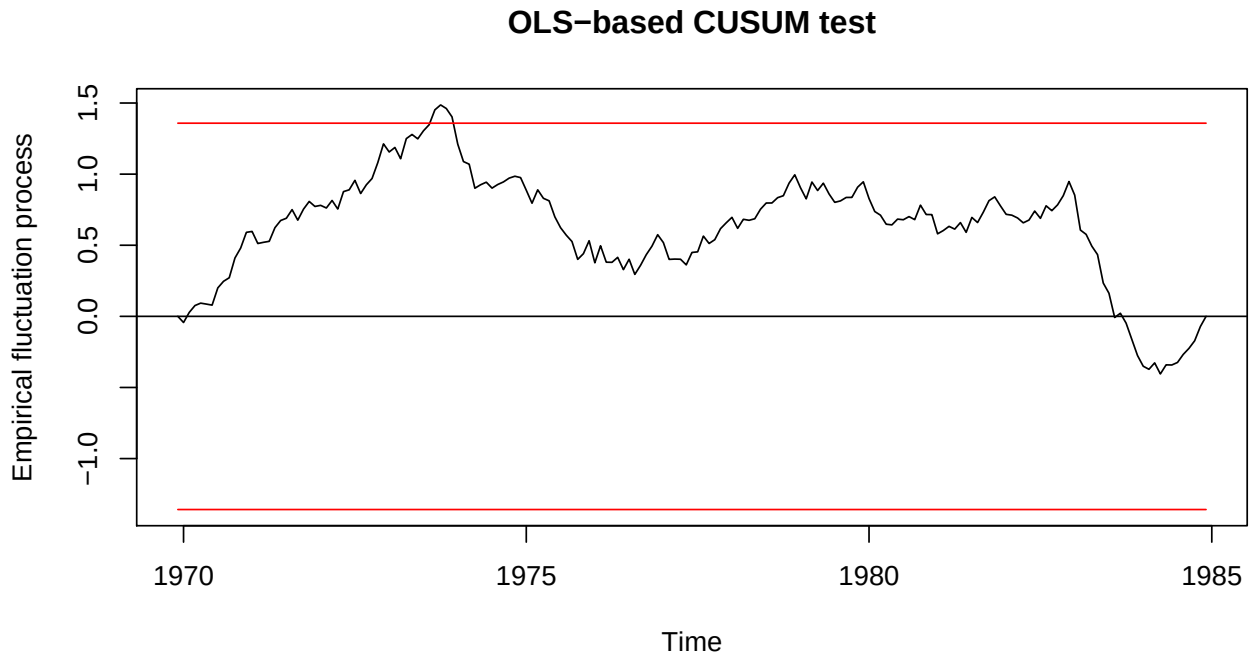
For both regular and well milled rice, there is a cointegrating relationship between retail and wholesale. This is because we can reject the null for *r=0* and cannot reject the null for *r<=1* cointegrated vector.

4.4 Structural Change

The package **strucchange** allows the researcher to test for the existince of structural change. We will use the german monetary economics example in (Kleiber and Zeileis, 2008), and use a fluctuation test using cumulative sums of the residuals.[4]

```
require(strucchange)
data("UKDriverDeaths")
dd<-log(UKDriverDeaths)
dd_dat<-ts.intersect(dd,dd1=stats::lag(dd,k=-1),dd12=stats::lag(dd,k=-12)) #buidling a lagged
  # dataset based on the ts class

dd_ocus<-efp(dd~dd1+dd12,data=dd_dat,type="OLS-CUSUM")
plot(dd_ocus)
```




```
sctest(dd_ocus)
##
##  OLS-based CUSUM test
##
## data:  dd_ocus
## S0 = 1.4866, p-value = 0.02407
```

The p-value is less than 0.05, and therefore There is a structural change. The package will also allow one to pinpoint the dates of the breakpoints.

```
dd_bp<-breakpoints(dd~dd1+dd12,data=dd_dat,h=.1)
summary(dd_bp)
##
```

[4]Other fluctuation tests are available. See the options of **efp**.

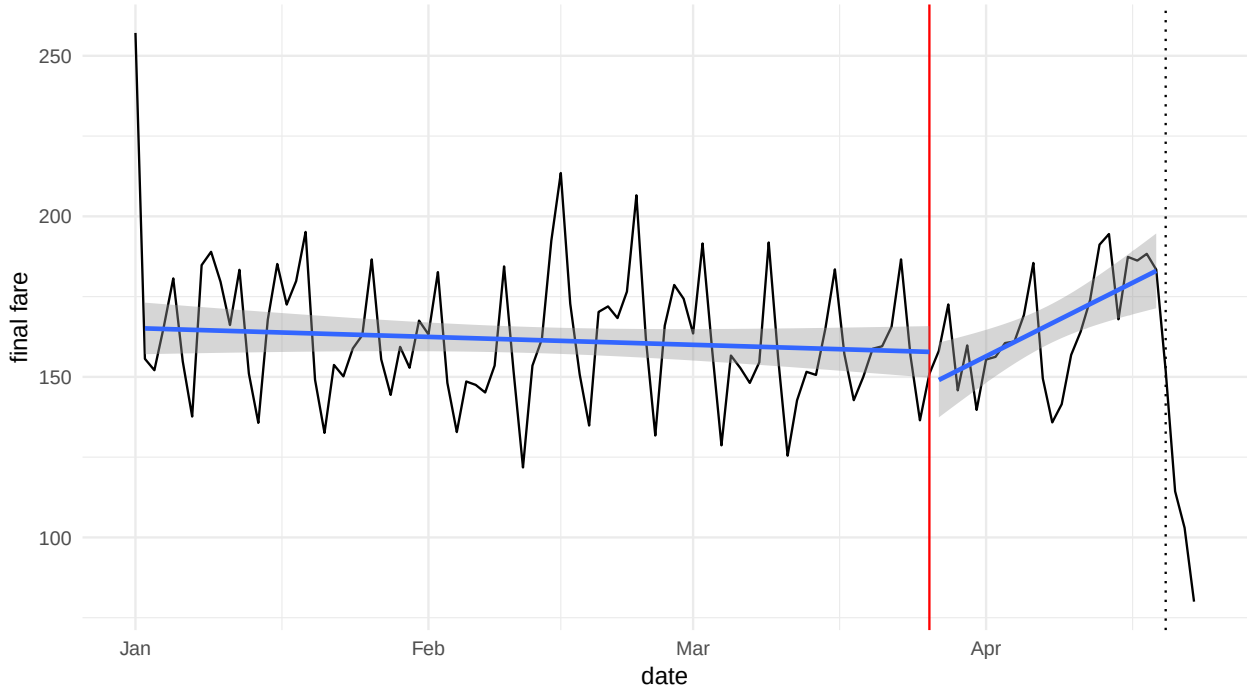
```
##   Optimal (m+1)-segment partition:
##
## Call:
## breakpoints.formula(formula = dd ~ dd1 + dd12, h = 0.1, data = dd_dat)
##
## Breakpoints at observation number:
##
## m = 1      46
## m = 2      46                  157
## m = 3      46 70               157
## m = 4      46 70    108        157
## m = 5      46 70        120 141 160
## m = 6      46 70 89 108    141 160
## m = 7      46 70 89 107 125 144 162
## m = 8   18 46 70 89 107 125 144 162
##
## Corresponding to breakdates:
##
## m = 1           1973(10)
## m = 2           1973(10)
## m = 3           1973(10) 1975(10)
## m = 4           1973(10) 1975(10)         1978(12)
## m = 5           1973(10) 1975(10)                  1979(12) 1981(9)
## m = 6           1973(10) 1975(10) 1977(5) 1978(12)          1981(9)
## m = 7           1973(10) 1975(10) 1977(5) 1978(11) 1980(5)  1981(12)
## m = 8   1971(6) 1973(10) 1975(10) 1977(5) 1978(11) 1980(5)  1981(12)
##
## m = 1
## m = 2   1983(1)
## m = 3   1983(1)
## m = 4   1983(1)
## m = 5   1983(4)
## m = 6   1983(4)
## m = 7   1983(6)
## m = 8   1983(6)
##
## Fit:
##
## m   0       1       2       3       4       5       6
## RSS    1.748   1.573   1.419   1.293   1.270   1.229   1.197
## BIC -302.609 -300.802 -298.652 -294.626 -277.039 -262.236 -246.111
##
## m   7       8
## RSS    1.190   1.183
## BIC -226.478 -206.737
```

4.4.1 Chow test – Grab Regression

In our investigation into the Grab Acquisition, we looked at the trend of prices before and after March 26, 2018, the date of the merger announcement. The purpose of the investigation at this juncture is to show if the trend of daily average final fare changes. For this task we included an interaction (not shown below), and conducted Chow test to look at instabiliy in the trend regression model.

```
require(haven);require(readxl)
grab1<-read_dta("data/Grab Trip Data 2018 by Day.dta")
driver<-read_xlsx("data/Annex B(9) - Driver Data.xlsx")
driver<-driver %>% gather(key=date,value=value,-X__1)
driver$date<-as.Date(as.numeric(driver$date),origin="1899-12-30")
driver<-driver %>% spread(X__1,value=value)
driver$bookingsperdriver=driver$`Unique Bookings`/driver$`Active drivers (for that day)`

ggplot(grab1)+geom_line(aes(x=date_local,y=finalfare,group=1))+
  geom_vline(xintercept = as.Date("03/26/18",format="%m/%d/%y"),color="red")+
  geom_vline(xintercept = as.Date("04/20/18",format="%m/%d/%y"),linetype=3)+
  geom_smooth(data=subset(grab1,date_local>=as.Date("01/02/18",format="%m/%d/%y") &
                            date_local<=as.Date("03/26/18",format="%m/%d/%y") ),
              aes(x=date_local,y=finalfare),method="lm")+
  geom_smooth(data=subset(grab1,date_local>=as.Date("03/27/18",format="%m/%d/%y") &
                            date_local<=as.Date("04/19/18",format="%m/%d/%y") ),
              aes(x=date_local,y=finalfare),method="lm")+
  xlab("date")+ylab("final fare")+theme_minimal()
```




```
data<-data.frame(ff=grab1$finalfare[1:109],bookpdrv=driver$bookingsperdriver[1:109]) #pretreatment is 1
data$date<-grab1$date_local[1:109]
data$treat<-ifelse(data$date>=as.Date("03/26/18",format="%m/%d/%y"),1,0)
data$trend<-1:nrow(data)

#manual F test
reg1.A<-lm(data=data[1:85,],formula = log(ff) ~ log(bookpdrv)+trend) #pre

reg1.B<-lm(data=data[86:109,],formula = log(ff) ~ log(bookpdrv)+ trend) #post

reg1.P<-lm(data=data,formula = log(ff) ~ log(bookpdrv)+trend)
```

```
reg1.P$df
## [1] 106

rssP <- sum(residuals(reg1.P)^2)
rssP
## [1] 1.108776

reg1.A$df
## [1] 82
rssA <- sum(residuals(reg1.A)^2)
rssA
## [1] 0.8907559

reg1.B$df
## [1] 21
rssB <- sum(residuals(reg1.B)^2)
rssB
## [1] 0.1108209

k=2

fcrit=qf(.95,df1=reg1.A$df,df2=reg1.B$df)
fcrit
## [1] 1.889626

Chow_Statistic=((rssP-(rssA+rssB))/k)/((rssA+rssB)/(reg1.A$df+reg1.B$df-(2*k)))
print("Chow_Statistic=((rssP-(rssA+rssB))/k)/((rssA+rssB)/(reg1.A$df+reg1.B$df-(2*k)))")
## [1] "Chow_Statistic=((rssP-(rssA+rssB))/k)/((rssA+rssB)/(reg1.A$df+reg1.B$df-(2*k)))"
paste("Chow Statistic:",Chow_Statistic)
## [1] "Chow Statistic: 5.29800316419645"
paste("Chow Critical Value:",fcrit)
## [1] "Chow Critical Value: 1.889625758871371"
```

The above Chow statistic is greater then the Chow critical value, which means there is a structural difference between the “pre” and the “post” regressions.

Chapter 5

Panel Data

For Panel data, we use the `plm` package if you wanted to run the fixed and random effects estimators as one might do in Stata.[1] In `plm` we specify the time series component and the cross-section unit component. Its rare to have a large (large N) panel of companies at PCC[2]; a merger investigation would normally have good data only from the merging parties. Below, i present how to estimate a fixed effects and random effects analysis in `R`.

```
require(plm);require(AER)
data("Grunfeld", package="plm")
pgr<-pdata.frame(Grunfeld,index=c("firm","year"))
```

The fixed and random effects are in the model option. For the fixed effects model, use "within", and "random" for the random effects model.

```
plm(inv~value+capital, data=pgr, model="within") %>% summary()
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = inv ~ value + capital, data = pgr, model = "within")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -184.00857  -17.64316    0.56337   19.19222  250.70974
##
## Coefficients:
##         Estimate Std. Error t-value  Pr(>|t|)
## value   0.110124   0.011857  9.2879 < 2.2e-16 ***
## capital 0.310065   0.017355 17.8666 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2244400
## Residual Sum of Squares: 523480
## R-Squared:      0.76676
```

[1] There are also other packages. Earlier, we use `nlme` for random effects.

[2] There are a few such panels being constructed for enforcement cases to investigate anti-competitive conduct, but results for such analyses are not available at this time.

```
## Adj. R-Squared: 0.75311
## F-statistic: 309.014 on 2 and 188 DF, p-value: < 2.22e-16
```

```
plm(inv~value+capital, data=pgr, model="random") %>% summary()
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = inv ~ value + capital, data = pgr, model = "random")
##
## Balanced Panel: n = 10, T = 20, N = 200
##
## Effects:
##                   var std.dev share
## idiosyncratic 2784.46   52.77 0.282
## individual    7089.80   84.20 0.718
## theta: 0.8612
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -177.6063  -19.7350    4.6851   19.5105  252.8743
##
## Coefficients:
##               Estimate Std. Error z-value Pr(>|z|)
## (Intercept) -57.834415  28.898935 -2.0013  0.04536 *
## value         0.109781   0.010493 10.4627  < 2e-16 ***
## capital       0.308113   0.017180 17.9339  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2381400
## Residual Sum of Squares: 548900
## R-Squared:      0.7695
## Adj. R-Squared: 0.76716
## Chisq: 657.674 on 2 DF, p-value: < 2.22e-16
```

5.1 Panel Regression Sandwich Estimator

Most panel (or quasi-panels) data have a clustering feature, the cross-sectional units can be divided into industries, or households can reside in neighborhoods. In the example below, there are state and year fixed effects, and we are saying that the errors are clustered at the state level. The effect of this correction is to raise the standard errors, but not enough to change the results of the regression.

```
library(clubSandwich)
data(MortalityRates)

# subset for deaths in motor vehicle accidents, 1970-1983
MV_deaths <- subset(MortalityRates, cause=="Motor Vehicle" &
                      year <= 1983 & !is.na(beertaxa))
```

```
# fit by OLS
lm_unweighted <- lm(mrate ~ 0 + legal + beertaxa +
                      factor(state) + factor(year), data = MV_deaths)
        #this is a "twoway" model, in PLM

coef_test(lm_unweighted, vcov = "CR1",
          cluster = MV_deaths$state, test = "naive-t")[1:2,]
##      Coef. Estimate   SE t-stat p-val (naive-t) Sig.
## 1    legal     7.59 2.44  3.108         0.00313   **
## 2 beertaxa     3.82 5.14  0.743         0.46128

coef_test(lm_unweighted, vcov = "CR2",
          cluster = MV_deaths$state, test = "Satterthwaite")[1:2,] #equivalent to
##      Coef. Estimate   SE t-stat  d.f. p-val (Satt) Sig.
## 1    legal     7.59 2.51  3.019 24.58      0.00583   **
## 2 beertaxa     3.82 5.27  0.725  5.77      0.49663
# HC2 correction
```

5.2 Udenna Merger Case – Panel data

Include a sample of the Udenna Regressions here

Bibliography

Angrist, J. and Pischke, J.-S. (2010). A note on bias in conventional standard errors under heteroskedasticity. unpublished.

Finkelstein, M. and Levin, B. (2015). *Statistics for Lawyers*. Springer, New York, NY, 3rd edition.

Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*. Springer, New York, NY.