UrbanModelData.R

plewis

Thu Feb 14 16:20:15 2019

############################  
# First, change directory to where  
# your (csv) files are   
############################  
  
# change working directory to  
# where our files are  
#  
# Be careful with the setwd command  
# and check where you are first  
print('I am in:')

## [1] "I am in:"

print(getwd())

## [1] "/Users/plewis/Downloads/GEOG0027\_Coursework/docs/files"

#  
# test for this file  
test = "Guangdong-Yearbook-2019.csv"  
  
# somewhere else it might be  
# if its not here   
# (put something appropriate!!)  
sub = 'files'  
  
if (test %in% list.files('.','\*.csv')){  
 print(paste('found',test))  
}else if (test %in% list.files(sub,'\*.csv')){  
 print(paste('found',test,'in',sub))  
   
 setwd(sub)  
   
 print('I have moved to:')  
 print(getwd())  
}

## [1] "found Guangdong-Yearbook-2019.csv"

#########################################################  
  
  
  
# The name of the file with year, urban\_land and possibly agr\_land  
result\_file <- 'results-2019.csv'  
# the name of the supplied data file with Guandong stats  
stats\_file <- "Guangdong-Yearbook-2019.csv"  
  
  
# load library  
library(readr)  
  
# read the datasets  
Guangdong\_Yearbook\_2019 <- read\_csv(stats\_file)

## Parsed with column specification:  
## cols(  
## index = col\_double(),  
## year = col\_double(),  
## investment = col\_double(),  
## population = col\_double(),  
## private\_wage = col\_double(),  
## agr\_output = col\_double(),  
## indust\_output = col\_double(),  
## agr\_pop = col\_double(),  
## avg\_wage = col\_double()  
## )

input <- read\_csv(result\_file)

## Parsed with column specification:  
## cols(  
## index = col\_double(),  
## year = col\_double(),  
## urban\_land = col\_double()  
## )

# fix the input dataset in case arg\_land doesnt exist  
if('agr\_land' %in% names(input)){  
 print(paste('using agricultural land data from',results))  
}else{  
 # If no ag land variable, insert one  
 print('inserting synthetic agricultural land data')  
 input$agr\_land = 2 \* max(input$urban\_land) - input$urban\_land  
}

## [1] "inserting synthetic agricultural land data"

# The years in Guangdong\_Yearbook\_2019$year and   
# input$year need to match  
# so we force this with the match function in R  
overlap <- match(Guangdong\_Yearbook\_2019$year,input$year,nomatch=0)  
input <- input[overlap,]  
Guangdong\_Yearbook\_2019 <- Guangdong\_Yearbook\_2019[overlap,]  
  
# print the datasets to visualise and check it looks ok  
print(input)

## # A tibble: 14 x 4  
## index year urban\_land agr\_land  
## <dbl> <dbl> <dbl> <dbl>  
## 1 0 1987 511000 3229000  
## 2 1 1988 520000 3220000  
## 3 2 1989 595000 3145000  
## 4 3 1990 656244 3083756  
## 5 4 1991 725000 3015000  
## 6 5 1992 780000 2960000  
## 7 6 1993 842833 2897167  
## 8 7 1994 938065 2801935  
## 9 8 1995 1003711 2736289  
## 10 9 1996 1122459 2617541  
## 11 10 1997 1250000 2490000  
## 12 11 1998 1450000 2290000  
## 13 12 1999 1650000 2090000  
## 14 13 2000 1870000 1870000

print(Guangdong\_Yearbook\_2019)

## # A tibble: 14 x 9  
## index year investment population private\_wage agr\_output indust\_output  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 1987 251. 5832. 4.12 349. 878.  
## 2 1 1988 354. 5928. 7.93 474. 1319.  
## 3 2 1989 347. 6025. 13.3 549. 1647.  
## 4 3 1990 381. 6246. 18.3 601. 1902.  
## 5 4 1991 478. 6349. 28.2 655. 2524.  
## 6 5 1992 922. 6463. 40.3 737. 3479.  
## 7 6 1993 1630. 6582. 70.6 899. 5237.  
## 8 7 1994 2141. 6691. 104. 1151. 7274.  
## 9 8 1995 2327. 6789. 151. 1445. 9721.  
## 10 9 1996 2328. 6897. 159. 1578. 10531.  
## 11 10 1997 2298. 7014. 198. 1656. 12373.  
## 12 11 1998 2668. 7116. 264. 1705. 13799.  
## 13 12 1999 3028. 7299. 301. 1745. 15303.  
## 14 13 2000 3234. 7499. 342. 1701. 16904.  
## # … with 2 more variables: agr\_pop <dbl>, avg\_wage <dbl>

################################################################  
  
  
X = data.frame(year=Guangdong\_Yearbook\_2019$year,  
 x1=Guangdong\_Yearbook\_2019$investment/Guangdong\_Yearbook\_2019$population,  
 x2=Guangdong\_Yearbook\_2019$agr\_output/Guangdong\_Yearbook\_2019$agr\_pop,  
 x3=log(Guangdong\_Yearbook\_2019$private\_wage),  
 x4=log(Guangdong\_Yearbook\_2019$avg\_wage),  
 x5=(Guangdong\_Yearbook\_2019$agr\_output/input$agr\_land)/  
 (Guangdong\_Yearbook\_2019$indust\_output/input$urban\_land))  
  
  
################################################################  
  
############################  
#  
# 4th, our model needs the change in urban area  
# with resect to year, which we might call du\_dy  
#  
# to get this, we need to calculate du and dy  
# i.e. the change ('delta') in urban area and year  
#   
# We will pack all of these datasets into a dataframe  
# called model\_data  
#  
############################  
  
# Calculate the delta in year for observation data  
# This will mostly be 1, but could be more  
# if you have missing years  
dy <- diff(input$year,differences = 1)   
  
# Calculate the delta in urban land for observation data  
du <- diff(input$urban\_land,differences = 1)  
  
# The rate of change of urban land per year  
du\_dy = du / dy  
  
# reconstruct the observation year  
# from the cumulative sum of dyear  
# This should give a 'year' dataset  
# with one fewer entry  
obs\_year <- data.frame(year=input$year[c(1)] + cumsum(dy))  
  
# Now select all columns of X that match obs\_year  
# and call it model\_data  
overlap <- match(input$year,obs\_year$year,nomatch=0)  
model\_data <- X[overlap,]  
  
# and add in du\_dy and dy to dataset  
model\_data$du\_dy <- du\_dy  
model\_data$dyear <- dy  
model\_data$urban\_land <- input$urban\_land[overlap]  
  
# save this file  
write.csv(model\_data, file = "model-2019.csv")  
  
###########################################################  
  
############################  
#  
# 5th, we do the linear modelling:  
# multi linear fit of du\_dy as function of X  
# using the function lm and the x and y data  
# stored in model\_data  
#  
############################  
  
  
# next time, you could skip all of the   
# above and just load the model\_data ...  
  
library(readr)  
model\_data <- read\_csv("model-2019.csv")

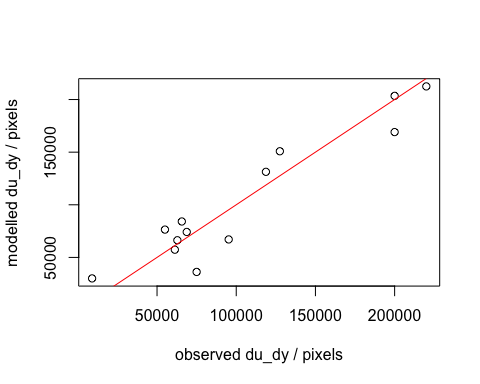
## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## X1 = col\_double(),  
## year = col\_double(),  
## x1 = col\_double(),  
## x2 = col\_double(),  
## x3 = col\_double(),  
## x4 = col\_double(),  
## x5 = col\_double(),  
## du\_dy = col\_double(),  
## dyear = col\_double(),  
## urban\_land = col\_double()  
## )

fit <- lm(du\_dy ~ x1 + x2 + x3 + x4 + x5, data=model\_data)  
  
print(summary(fit))

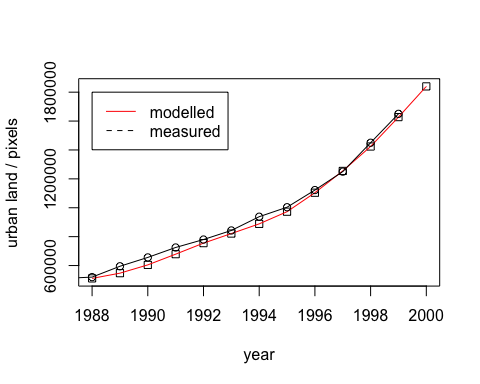
##   
## Call:  
## lm(formula = du\_dy ~ x1 + x2 + x3 + x4 + x5, data = model\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23328 -18491 -3506 7522 38790   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1346051 2575068 0.523 0.617  
## x1 16800 337936 0.050 0.962  
## x2 802897 710485 1.130 0.296  
## x3 72200 103742 0.696 0.509  
## x4 -213235 367635 -0.580 0.580  
## x5 1761121 993690 1.772 0.120  
##   
## Residual standard error: 27580 on 7 degrees of freedom  
## Multiple R-squared: 0.8961, Adjusted R-squared: 0.8218   
## F-statistic: 12.07 on 5 and 7 DF, p-value: 0.002475

###########################################################  
  
############################  
#  
# 6th, lets predict the du\_dy from  
# the model, and plot the measured and modelled  
# values  
#  
############################  
  
# predict from model  
model\_du\_dy <- predict(fit, data=model\_data)  
  
plot(model\_data$du\_dy,model\_du\_dy  
 ,xlab='observed du\_dy / pixels',  
 ,ylab='modelled du\_dy / pixels') + abline(0,1,col="red")



## integer(0)

###########################################################  
  
############################  
#  
# 6th, lets predict the du\_dy from  
# the model, and plot the measured and modelled  
# values  
#  
############################  
  
# Now reconstruct urban area from du\_dy  
modelled <- data.frame(year=model\_data$year+dy[c(1)],  
 y=model\_data$urban\_land[c(1)] - model\_du\_dy[c(1)] + cumsum(model\_du\_dy))  
measured <- data.frame(year=model\_data$year,  
 y=model\_data$urban\_land)  
  
# and plot the urban area measured and modelled  
plot(modelled$year,modelled$y,pch=0  
 ,xlab='year',  
 ,ylab='urban land / pixels')   
lines(modelled$year,modelled$y,col="red")  
points(measured$year,measured$y,pch=1)   
lines(measured$year,measured$y,,col="black")  
legend(1988, 1800000, legend=c("modelled","measured"),  
 col=c("red", "black"), lty=1:2)



###########################################################  
  
#########################  
#  
# We can take a subset of data   
# from model\_data  
#  
#########################  
  
# a utility function: last + first elements  
last <- function(x) { tail(x, n = 1) }  
first <- function(x) { head(x, n = 1) }  
  
  
count = 10  
ystart = first(model\_data$year)  
yend = last(model\_data$year)  
print(yend)

## [1] 1999

# sub-dataset for year from ystart for count years  
sub = model\_data[model\_data$year %in% seq(ystart,ystart+count-1),]  
  
print(sub)

## # A tibble: 10 x 10  
## X1 year x1 x2 x3 x4 x5 du\_dy dyear urban\_land  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1987 0.0430 0.0771 1.42 7.46 0.0628 9000 1 511000  
## 2 2 1988 0.0596 0.104 2.07 7.72 0.0580 75000 1 520000  
## 3 3 1989 0.0576 0.119 2.59 7.89 0.0630 61244 1 595000  
## 4 4 1990 0.0611 0.126 2.91 7.98 0.0672 68756 1 656244  
## 5 5 1991 0.0753 0.136 3.34 8.12 0.0624 55000 1 725000  
## 6 6 1992 0.143 0.153 3.70 8.30 0.0558 62833 1 780000  
## 7 7 1993 0.248 0.188 4.26 8.58 0.0499 95232 1 842833  
## 8 8 1994 0.320 0.244 4.64 8.87 0.0530 65646 1 938065  
## 9 9 1995 0.343 0.304 5.02 9.02 0.0545 118748 1 1003711  
## 10 10 1996 0.337 0.329 5.07 9.12 0.0643 127541 1 1122459

#########################  
#  
# What if we used fewer data points?  
#  
#########################  
fits <- lm(du\_dy ~ x1 + x2 + x3 + x4 + x5, data=sub)  
  
print(fits$coefficients)

## (Intercept) x1 x2 x3 x4 x5   
## -258091.28 -683858.06 1025197.52 -17180.24 67169.70 -3942492.35

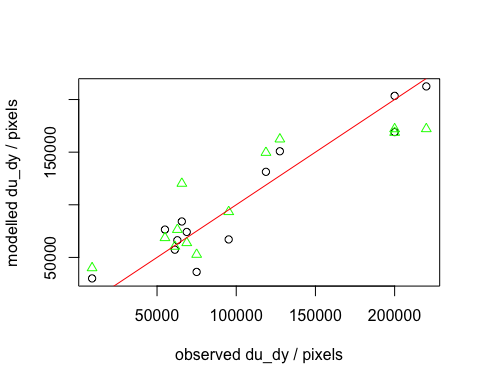
model\_data$year[5:7]

## [1] 1991 1992 1993

###########################################################  
#########################  
#  
# What if we used fewer data points?  
#  
#########################  
count <- 10  
  
# loop over count years   
for (y0 in seq(first(model\_data$year),last(model\_data$year)-count,1)){  
 years <- seq(y0,y0+count-1)  
 sub = model\_data[model\_data$year %in% years,]  
 fits <- lm(du\_dy ~ x2 + x5 , data=sub)  
  
 print(sub)  
 print(fits$coefficients)  
 print(summary(fits))  
}

## # A tibble: 10 x 10  
## X1 year x1 x2 x3 x4 x5 du\_dy dyear urban\_land  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1987 0.0430 0.0771 1.42 7.46 0.0628 9000 1 511000  
## 2 2 1988 0.0596 0.104 2.07 7.72 0.0580 75000 1 520000  
## 3 3 1989 0.0576 0.119 2.59 7.89 0.0630 61244 1 595000  
## 4 4 1990 0.0611 0.126 2.91 7.98 0.0672 68756 1 656244  
## 5 5 1991 0.0753 0.136 3.34 8.12 0.0624 55000 1 725000  
## 6 6 1992 0.143 0.153 3.70 8.30 0.0558 62833 1 780000  
## 7 7 1993 0.248 0.188 4.26 8.58 0.0499 95232 1 842833  
## 8 8 1994 0.320 0.244 4.64 8.87 0.0530 65646 1 938065  
## 9 9 1995 0.343 0.304 5.02 9.02 0.0545 118748 1 1003711  
## 10 10 1996 0.337 0.329 5.07 9.12 0.0643 127541 1 1122459  
## (Intercept) x2 x5   
## 18598.03 325284.88 -44338.31   
##   
## Call:  
## lm(formula = du\_dy ~ x2 + x5, data = sub)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -31889 -4603 4133 10789 25190   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 18598 83085 0.224 0.82927   
## x2 325285 84926 3.830 0.00646 \*\*  
## x5 -44338 1304739 -0.034 0.97384   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 21090 on 7 degrees of freedom  
## Multiple R-squared: 0.6975, Adjusted R-squared: 0.611   
## F-statistic: 8.069 on 2 and 7 DF, p-value: 0.01523  
##   
## # A tibble: 10 x 10  
## X1 year x1 x2 x3 x4 x5 du\_dy dyear urban\_land  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 1988 0.0596 0.104 2.07 7.72 0.0580 75000 1 520000  
## 2 3 1989 0.0576 0.119 2.59 7.89 0.0630 61244 1 595000  
## 3 4 1990 0.0611 0.126 2.91 7.98 0.0672 68756 1 656244  
## 4 5 1991 0.0753 0.136 3.34 8.12 0.0624 55000 1 725000  
## 5 6 1992 0.143 0.153 3.70 8.30 0.0558 62833 1 780000  
## 6 7 1993 0.248 0.188 4.26 8.58 0.0499 95232 1 842833  
## 7 8 1994 0.320 0.244 4.64 8.87 0.0530 65646 1 938065  
## 8 9 1995 0.343 0.304 5.02 9.02 0.0545 118748 1 1003711  
## 9 10 1996 0.337 0.329 5.07 9.12 0.0643 127541 1 1122459  
## 10 11 1997 0.328 0.342 5.29 9.18 0.0672 200000 1 1250000  
## (Intercept) x2 x5   
## -99364.34 395408.99 1872311.74   
##   
## Call:  
## lm(formula = du\_dy ~ x2 + x5, data = sub)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30527 -14093 -4377 17821 38208   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -99364 85171 -1.167 0.28156   
## x2 395409 92804 4.261 0.00374 \*\*  
## x5 1872312 1406836 1.331 0.22494   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 25740 on 7 degrees of freedom  
## Multiple R-squared: 0.747, Adjusted R-squared: 0.6747   
## F-statistic: 10.33 on 2 and 7 DF, p-value: 0.008146  
##   
## # A tibble: 10 x 10  
## X1 year x1 x2 x3 x4 x5 du\_dy dyear urban\_land  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 3 1989 0.0576 0.119 2.59 7.89 0.0630 61244 1 595000  
## 2 4 1990 0.0611 0.126 2.91 7.98 0.0672 68756 1 656244  
## 3 5 1991 0.0753 0.136 3.34 8.12 0.0624 55000 1 725000  
## 4 6 1992 0.143 0.153 3.70 8.30 0.0558 62833 1 780000  
## 5 7 1993 0.248 0.188 4.26 8.58 0.0499 95232 1 842833  
## 6 8 1994 0.320 0.244 4.64 8.87 0.0530 65646 1 938065  
## 7 9 1995 0.343 0.304 5.02 9.02 0.0545 118748 1 1003711  
## 8 10 1996 0.337 0.329 5.07 9.12 0.0643 127541 1 1122459  
## 9 11 1997 0.328 0.342 5.29 9.18 0.0672 200000 1 1250000  
## 10 12 1998 0.375 0.348 5.58 9.23 0.0783 200000 1 1450000  
## (Intercept) x2 x5   
## -133122.0 445329.1 2219433.6   
##   
## Call:  
## lm(formula = du\_dy ~ x2 + x5, data = sub)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28669 -9391 -964 4247 33648   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -133122 57686 -2.308 0.05437 .   
## x2 445329 87088 5.114 0.00138 \*\*  
## x5 2219434 981854 2.260 0.05829 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23560 on 7 degrees of freedom  
## Multiple R-squared: 0.8607, Adjusted R-squared: 0.8209   
## F-statistic: 21.62 on 2 and 7 DF, p-value: 0.001009

#fit <- lm(du\_dy ~ x1 + x2 + x3 + x4 + x5, data=model\_data)  
  
  
#c(array(fit$coefficients))  
  
###########################################################  
  
  
fit <- lm(du\_dy ~ x1 + x2 + x3 + x4 + x5, data=model\_data)  
  
#print(summary(fit))  
# lets try another model  
fit1 <- lm(du\_dy ~ x2 + x5 + 0 , data=model\_data)  
  
model\_du\_dy <- predict(fit, data=model\_data)  
model\_du\_dy1 <- predict(fit1, data=model\_data)  
  
  
plot(model\_data$du\_dy,model\_du\_dy  
 ,xlab='observed du\_dy / pixels',  
 ,ylab='modelled du\_dy / pixels')   
points(model\_data$du\_dy,model\_du\_dy1,pch=2,col='green')   
abline(0,1,col="red")



print(summary(fit1)) # $adj.r.squared

##   
## Call:  
## lm(formula = du\_dy ~ x2 + x5 + 0, data = model\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -54605 -30883 830 22231 47869   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## x2 484468 97327 4.978 0.000417 \*\*\*  
## x5 42272 361337 0.117 0.908979   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 31700 on 11 degrees of freedom  
## Multiple R-squared: 0.9428, Adjusted R-squared: 0.9324   
## F-statistic: 90.66 on 2 and 11 DF, p-value: 1.464e-07

print(summary(fit)) # $adj.r.squared

##   
## Call:  
## lm(formula = du\_dy ~ x1 + x2 + x3 + x4 + x5, data = model\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23328 -18491 -3506 7522 38790   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1346051 2575068 0.523 0.617  
## x1 16800 337936 0.050 0.962  
## x2 802897 710485 1.130 0.296  
## x3 72200 103742 0.696 0.509  
## x4 -213235 367635 -0.580 0.580  
## x5 1761121 993690 1.772 0.120  
##   
## Residual standard error: 27580 on 7 degrees of freedom  
## Multiple R-squared: 0.8961, Adjusted R-squared: 0.8218   
## F-statistic: 12.07 on 5 and 7 DF, p-value: 0.002475

###########################################################  
###########################################################