Unit 2 Paper, Reproducibility Appendix

Jack Schaffer - jmschaff - UM ID: 93199360

2/21/2018

Prerequisites

The following code is required in order to carry out the computations of this paper.

```r
# Load Dependencies
library(ggplot2)
library(knitr)
```

Overview

The computations in this document assess the power-law scaling proposal, develop alternate models to predict GMP as linear but no supra-linear function of population size, as well as additional variables describing the nature of its local economy, and compare model fits utilizing loss functions. In order to carry out this analysis, various data structures and functions were used to organize our data and compute effectively in addition to the generation of visuals and figures to represent data.

Conceptual Preliminaries

If $Y \approx cN^b$ with c>0 and b>1, then $log(Y/N) \approx \beta_0 + \beta_1 log(N)$, for some β_0 and some $\beta_1 > 0$, and also that $log(Y) \approx \beta_0 + (1+\beta_1)log(N)$.

We can show this with the following algebra.

Here we will find $log(Y/N) \approx \beta_0 + \beta_1 log(N)$ from $Y \approx cN^b$:

$$Y \approx cN^b$$

$$\frac{Y}{N} \approx cN^{b-1}$$

$$\log(Y/N) \approx \log(cN^{b-1})$$

$$\log(Y/N) \approx \log(c) + (b-1)\log(N)$$

and because c $>$ 0 we have some β_0 and because b $>$ 1, we have some $\beta_1 > 0$.

Now to show $log(Y) \approx \beta_0 + (1+\beta_1)log(N)$ from $\log(Y/N) \approx \beta_0 + \beta_1 \log(N)$:

$$\log(Y/N) \approx \beta_0 + \beta_1 \log(N)$$

$$\log(Y) - \log(N) \approx \beta_0 + \beta_1 \log(N)$$

$$\log(Y) - \log(N) - \beta_1 \log(N) \approx \beta_0$$

$$\log(Y) - \log(N)(1+\beta_1) \approx \beta_0$$

$$\log(Y) \approx \beta_0 + (1+\beta_1)\log(N)$$

As for our hypothesese, the appearance of supra-linear scaling is present when taking into account shares of the economy deriving from information, communication, and technology, however when the share of the economy from finance firms are taken into account, the appearance of supra-linear scaling disappears.

Additionally, while the appearance of supra-linear scaling is present when taking into account population, when taking into account shares of the economy deriving from professional and tehcnical services, the appearance of supra-linear scaling disappears.

Lastly. while the appearance of supra-linear scaling is present when taking into account population, when including the primary source of income for the area the appearance of supra-linear scaling disappears.

Exploratory Analysis

The following code reads in the data

```r
# Read gmp-2006.csv and store it as a dataframe variable named msadata```{r}
#msadata = read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/gmp-2006.csv")
msadata = read.csv("msadata.csv")

# Set rownames
row.names(msadata) <- msadata$MSA

msadata[,"pop"] = as.double(msadata[,"pop"])
msadata[,"pcgmp"] = as.double(msadata[,"pcgmp"])

# Create sub samples
msadata_omit_finance = msadata[complete.cases(msadata[,"finance"]),]
msadata_omit_prof.tech = msadata[complete.cases(msadata[,"prof.tech"]),]
msadata_omit_ict = msadata[complete.cases(msadata[,"ict"]),]
finance_ict = intersect(msadata_omit_finance, msadata_omit_ict)
finance_ict_prof.tech = intersect(finance_ict, msadata_omit_prof.tech)
```

In this chunk of code we create the variables gmp and primary:

```r
gmp = msadata$pcgmp * msadata$pop
msadata = data.frame(msadata, gmp)

# Create sub samples
msadata_omit_finance = msadata[complete.cases(msadata[,"finance"]),]
msadata_omit_prof.tech = msadata[complete.cases(msadata[,"prof.tech"]),]
msadata_omit_ict = msadata[complete.cases(msadata[,"ict"]),]
finance_ict = intersect(msadata_omit_finance, msadata_omit_ict)
finance_ict_prof.tech = intersect(finance_ict, msadata_omit_prof.tech)

primary = rep("", nrow(finance_ict_prof.tech))

for (i in 1:length(primary)){
  if (finance_ict_prof.tech[i,"finance"] > finance_ict_prof.tech[i,"ict"] & finance_ict_prof.tech[i,"fi
    primary[i] = "finance"
  }
```

```
  else if (finance_ict_prof.tech[i,"ict"] > finance_ict_prof.tech[i,"finance"] & finance_ict_prof.tech[i
    primary[i] = "ict"
  }
  else if (finance_ict_prof.tech[i,"prof.tech"] > finance_ict_prof.tech[i,"finance"] & finance_ict_prof
    primary[i] = "prof.tech"
  }
}

finance_ict_prof.tech = data.frame(finance_ict_prof.tech, primary)
```

As discussed in the paper, missing data was ommitted from our sample during analysis. The following code does this for us:

```
msadata_omit_nan = msadata[complete.cases(msadata),]
msadata_omit_management = msadata[complete.cases(msadata[,"management"]),]
finance_manage = intersect(msadata_omit_finance,msadata_omit_management)
finance_prof.tech = intersect(msadata_omit_finance, msadata_omit_prof.tech)
prof.tech_ict = intersect(msadata_omit_prof.tech, msadata_omit_ict)
prof.tech_manage = intersect(msadata_omit_prof.tech, msadata_omit_management)
ict_manage = intersect(msadata_omit_ict, msadata_omit_management)
finance_ict_manage = intersect(finance_ict, msadata_omit_management)
ict_prof.tech_manage = intersect(ict_manage, msadata_omit_prof.tech)
```

```
DF_Subsets = list(msadata,msadata_omit_finance, msadata_omit_ict, msadata_omit_prof.tech, msadata_omit_m

Subsets = c("All", "Finance", "ICT", "Prof.Tech", "Management","Finance + ICT", "Finance + Prof.Tech",
Rows = c(nrow(msadata),nrow(msadata_omit_finance), nrow(msadata_omit_ict), nrow(msadata_omit_prof.tech)

Mean_Finance = c()
Mean_ICT = c()
Mean_Prof.Tech = c()
Mean_Management = c()

for (df in DF_Subsets){
  Mean_Finance = c(Mean_Finance,mean(df[,"finance"]))
  Mean_ICT = c(Mean_ICT, mean(df[,"ict"]))
  Mean_Prof.Tech = c(Mean_Prof.Tech, mean(df[,"prof.tech"]))
  Mean_Management = c(Mean_Management,mean(df[,"management"]))
}

table1 = data.frame(Subsets, Rows, Mean_Finance, Mean_ICT, Mean_Prof.Tech, Mean_Management)
row.names(table1) = table1$Subsets

kable(table1[order(-Rows),])
```

The subset we will be using as our analysis sample is Finance + ICT + Prof.Tech. This table shows the differences of the Rows and Means for each of the three variables where their data is available:

```
difference = matrix(ncol = 3, nrow = 3)
difference[1,1] = "Finance"
difference[1,2] = "114"
difference[1,3] = "0.0015544"
difference[2,1] = "ICT"
difference[2,2] = "82"
difference[2,3] = "-0.0045088"
```

```
difference[3,1] = "Prof.Tech"
difference[3,2] = "43"
difference[3,3] = "0.0012284"
table = data.frame(difference)

kable(difference, col = c("Economy Source", "Row Difference", "Mean Difference"))
```

The following code generates various scatter plots: Create scatter plots of GMP (Y) vs population size (x), of log GMP vs population size, of GMP vs logged population size and of log GMP vs log population size. (This is overall GMP, not per-capita GMP.) Add smoothed curves to each plot, without accompanying standard error envelopes. Which is the better scale for capturing patterns in the data using a regression model of relatively simple structure?

```
gmp_pop_gp = ggplot(msadata,aes(x = pop, y = gmp)) + labs(title = "GMP vs. Population", x = "Population
gmp_pop_gp + geom_point() + geom_smooth(method = "loess", se = FALSE, aes(color = "blue")) + geom_smooth
```

```
loggmp_pop_gp = ggplot(msadata,aes(x = pop, y = log(gmp))) + labs(title = "log(GMP) vs. Population", x =
loggmp_pop_gp + geom_point() + geom_smooth(method = "loess", se = FALSE, aes(color = "blue")) + geom_smo
```

```
gmp_logpop_gp = ggplot(msadata,aes(x = log(pop), y = gmp)) + labs(title = "GMP vs. log(Population)", x =
gmp_logpop_gp + geom_point() + geom_smooth(method = "loess", se = FALSE, aes(color = "blue")) + geom_smo
```

```
loggmp_logpop_gp = ggplot(msadata,aes(x = log(pop), y = log(gmp))) + labs(title = "log(GMP) vs. log(Popu
loggmp_logpop_gp + geom_point() + geom_smooth(method = "loess", se = FALSE, aes(color = "blue")) + geom_
```

The preffered plot from above is log(GMP) vs. log(Population) plot as the plots clearly depict an upward linear trend, with less variance than the other plots.

The follow code generates plots that display other characteristics of the data such as shares of the economy from finance, ict, and prof.tech:

```
loggmp_logpop_gp + geom_point(alpha = 2/3, aes(colour = c(finance))) + scale_colour_continuous(low = "re

loggmp_logpop_gp + geom_point(alpha = 2/3, aes(colour = c(ict))) + scale_colour_continuous(low = "red",

loggmp_logpop_gp + geom_point(alpha = 2/3, aes(colour = c(prof.tech))) + scale_colour_continuous(low =

loggmp_logpop_gp = ggplot(finance_ict_prof.tech,aes(x = log(pop), y = log(gmp))) + labs(title = "log(GMP
loggmp_logpop_gp + geom_point(alpha = 2/3, aes(colour = primary)) + scale_color_manual(name = "Highest S
```

Interestingly from the primary variable we created to showcase which source of the economy is the highest contributor to an areas GMP, finance is orwhelmingly the most frequent, followed by ICT, and, lastly, Prof.Tech only having a higher share of the economy in a single datum. This may not be as useful as previously thought before this exploration.

```
loggmp_logpop_gp = ggplot(msadata,aes(x = log(pop), y = log(gmp))) + labs(title = "log(GMP) vs. log(Popu
loggmp_logpop_gp + geom_point() + geom_smooth(method = "loess", se = FALSE, aes(color = "blue")) + geom_
```

Now, we will linearly regress log GMPon the log of sample size:

```
mod1 = lm(log(pcgmp) ~ log(pop), data = msadata)
summary(mod1)
```

From the proofs we derived above, we known that $\log(pcgmp) = \log(c) + (b-1)\log(pop)$, thus from the model summary we can determine what c and b are in the power law scaling formula.

The coefficient for the intercept is 8.7623, thus:

$$8.7623 = \log(c)$$
$$c = 6388.788$$

The coefficient for log(pop) is 0.12326, thus:

$$0.12326 = (b - 1)$$
$$b = 1.12326$$

These findings are compatible with the supra-linear power-lawe scaling hypothesis becuase it follows the requirements that c > 0 and b > 1.

Now we will observe the residual of the linear model to determine the credibility of the model:

```r
plot(residuals(mod1), xlab = "Fitted Values", ylab = "Residual")
abline(0,0)
```

According to the residual plot, we can see that there is no indication of a pattern. This suggests that the fit is proper. If there was an indication of a pattern we would consider other variables, transformations of current variables, or increased degress of variables. Additionally, the summary output indicated a standard error of 0.238 on 242 degrees of error and the plot shows that we should believe the standard errors that the summary provides.

The in-sample loss can be calculated using the squared-error loss on the log scale, $L(Z, \theta) = (\log_{10}(Y) - \mu_\theta(N))^2$.

The following calculation gives us the in-sample loss:

```r
in_sample_loss = 0
predictions = as.double(predict(mod1))

for (i in 1:nrow(msadata)){
  in_sample_loss = in_sample_loss + (log(msadata[i,"pcgmp"]) - predictions[i])^2
}
in_sample_loss
```

The in-sample loss is: 13.71174

We will now use 5-Fold Cross Validation to determine out-of sample error:

```r
cv.lm <- function(data, formula, nfolds = 5) {
  formula <- sapply(formula, as.formula)
  n <- nrow(data)
  fold.labels <- sample(rep(1:nfolds, length.out = n))
  mses <- matrix(NA, nrow = nfolds, ncol = length(formula))
  colnames <- as.character(formula)
  for (fold in 1:nfolds){
    test.rows <- which(fold.labels == fold)
    train <- data[-test.rows, ]
    test <- data[test.rows, ]
    for (form in 1:length(formula)){
      current.model <- lm(formula = formula[[form]], data = train)
      predictions <- predict(current.model, newdata = test)
      test.responses <- eval(formula[[form]][[2]], envir = test)
      test.errors <- test.responses - predictions
      mses[fold,form] <- mean(test.errors^2)
    }
  }
  return(colMeans(mses))
}
```

```
formula = list(log(pcgmp)~log(pop))
cv.lm(msadata,formula)
```

The 5-fold cross validation mean squared error is 0.057.

Now that we have done the calculations for the full sample for population size, we must reduce our sample to an analysis sample in order to account for missing values of our data for the variables that are taken into account with our alternative models. Our analysis sample will omit nan values for the following column vectors: finance, ict, and prof.tech.

The alternative models are:

log(pcgmp) ~ finance log(pcgmp) ~ ict log(pcgmp) ~ finance + ict log(pcgmp) ~ finance + ict + prof.tech

For consistency with our analysis, we will repeat the above calculations for the model with formala $\log(pcgmp)\ \log(pop)$:

```
analysisSample = finance_ict_prof.tech

mod1 <- lm(log(pcgmp)~log(pop), data = analysisSample)
summary(mod1)
```

According to the summary output above, the coefficents only differ to the thousandth decimal and the difference in residual standard error is very minimal.

```
mod2 <- lm(log(pcgmp)~finance, data = analysisSample)
summary(mod2)
mod3 <- lm(log(pcgmp)~ict, data = analysisSample)
summary(mod3)
mod4 <- lm(log(pcgmp)~finance + ict, data = analysisSample)
summary(mod4)
mod5 <- lm(log(pcgmp)~finance + ict + prof.tech, data = analysisSample)
summary(mod5)
```

```
mod1_in_sample_loss = 0
mod1_predictions = as.double(predict(mod1))
mod2_in_sample_loss = 0
mod2_predictions = as.double(predict(mod2))
mod3_in_sample_loss = 0
mod3_predictions = as.double(predict(mod3))
mod4_in_sample_loss = 0
mod4_predictions = as.double(predict(mod4))
mod5_in_sample_loss = 0
mod5_predictions = as.double(predict(mod5))

for (i in 1:nrow(analysisSample)){
  mod1_in_sample_loss = mod1_in_sample_loss + (log(analysisSample[i,"pcgmp"]) - mod1_predictions[i])^2
  mod2_in_sample_loss = mod2_in_sample_loss + (log(analysisSample[i,"pcgmp"]) - mod2_predictions[i])^2
  mod3_in_sample_loss = mod3_in_sample_loss + (log(analysisSample[i,"pcgmp"]) - mod3_predictions[i])^2
  mod4_in_sample_loss = mod4_in_sample_loss + (log(analysisSample[i,"pcgmp"]) - mod4_predictions[i])^2
  mod5_in_sample_loss = mod5_in_sample_loss + (log(analysisSample[i,"pcgmp"]) - mod5_predictions[i])^2
}

in_sample_loss = matrix(c(mod1_in_sample_loss, mod2_in_sample_loss, mod3_in_sample_loss, mod4_in_sample_
```

```r
models = matrix(c("log(pcgmp)~log(pop)", "log(pcgmp)~finance", "log(pcgmp)~ict", "log(pcgmp)~finance+ic
df = data.frame(models, in_sample_loss)

formula = list(log(pcgmp)~log(pop), log(pcgmp)~finance, log(pcgmp)~ict, log(pcgmp)~finance+ict, log(pcg
cv = cv.lm(analysisSample,formula)
df = data.frame(df, cv)

in_sample_lossTbl <- kable(df, col.names = c("Model", "In-Sample Loss", "5-Fold Cross Validation"))
print(in_sample_lossTbl)
```