Power Law Scaling Proposal vs Alternate Linear Models in MSAs

author: Alex Haase

Required Libraries

```
require(boot)

## Loading required package: boot

## Warning: package 'boot' was built under R version 3.4.2

library(MASS)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.3

rm(list=ls()) # clear global environment

msadata = read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/gmp-
2006.csv", TRUE, ",")
class(msadata) #create msa dataframe

## [1] "data.frame"

#msadata objects
msaName = msadata$MSA #MSA name (metropolitan statistical areas)
pcgmp = msadata$pcgmp #per-capita GMP
pop = msadata$pop #population
finance = msadata$finance
prof.tech = msadata$prof.tech
ict = msadata$ict
management = msadata$management
```

Conceptual Preliminaries and Initial Hypothesis

primary fields: MSA name, pcgmp, pop comparison variables: prof.tech, information, ict, management

I agree with the competing theory to the supra-linear power law scaling proposition. It seems more reasonable that moving alone is not the only factor to attribute to economic productivity; it can also be attributed to current financial establishments and concentrations, along with information, communication, and technology.

Explanatory Analysis and GMP equation

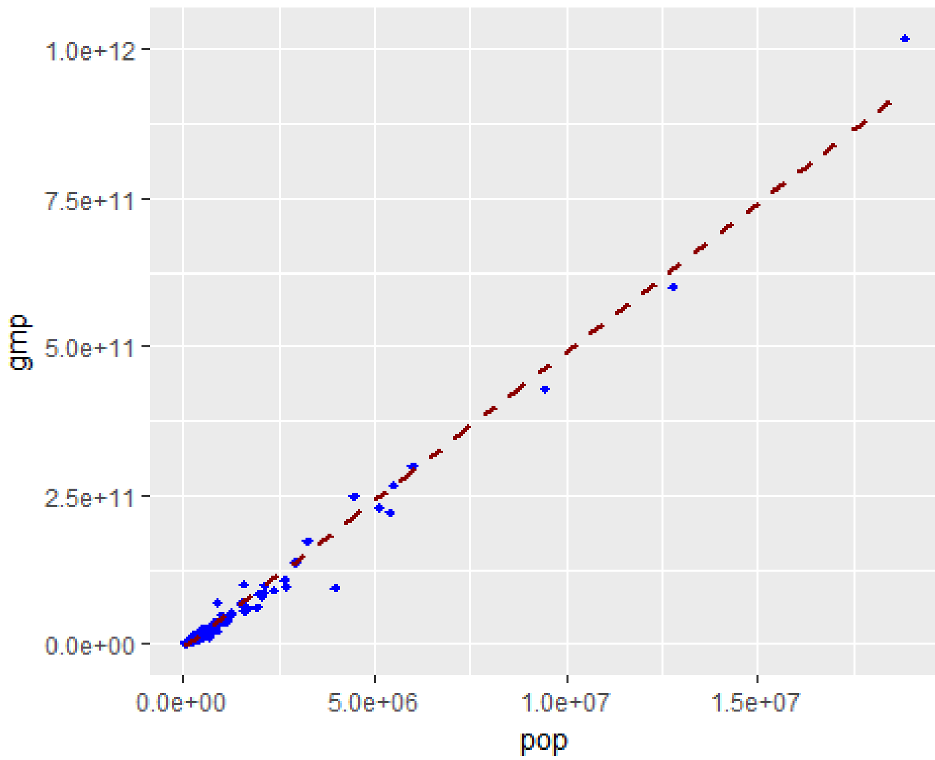
```
#per-capita_GMP = GMP/N, therefore: GMP = pop * per-capita_GMP
gmp = pop * as.double(pcgmp)
#gmp[1] #testing
#gmp[2] #testing
```

Summarizing the proportions of data by variable and handling Missing data

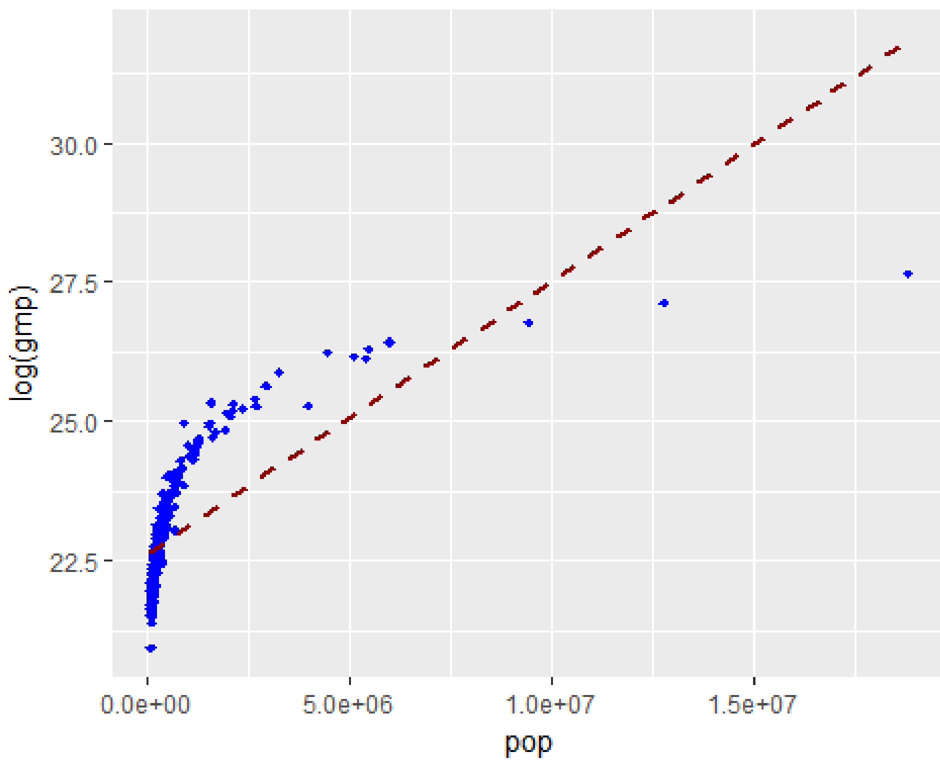
```
#first ommitting each case with an "NA" present per each variable

omit_finance = na.omit(finance)
omit_prof.tech = na.omit(prof.tech)
omit_itc = na.omit(ict)
omit_management = na.omit(management)

financeMean = mean(omit_finance)
#financeMean

prof.TechMean = mean(omit_prof.tech)
#prof.TechMean

itcMean = mean(omit_itc)
#itcMean

managementMean = mean(omit_management)
#managementMean

#removing entire rows with "NA" present in the dataset
#str(msadata)
#complete.cases(msadata) #return boolean for rows with NA

clean_msadata = msadata[complete.cases(msadata), ] #no NA's
#str(clean_msadata)
#clean_msadata
```

Scatterplots via ggplot with smoothing

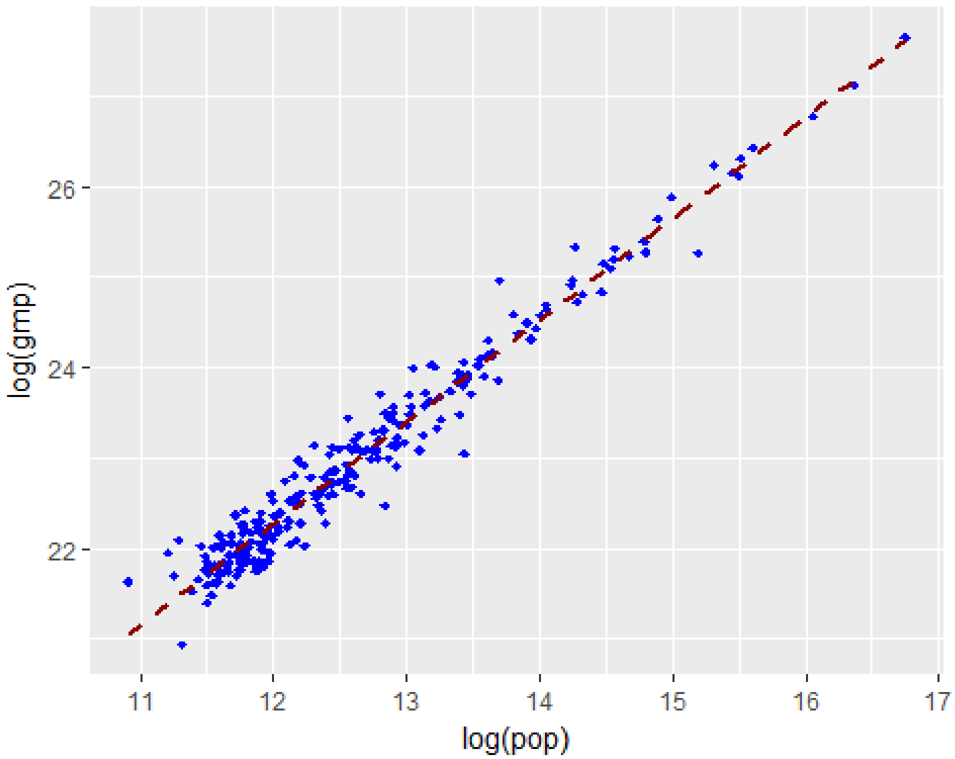
```r
#GMP vs Pop
gmp_vs_pop_Plot = ggplot(msadata, aes(x=pop, y=gmp)) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
gmp_vs_pop_Plot
```



```r
#log GMP vs Pop
log_GMP_vs_pop_Plot = ggplot(msadata, aes(x=pop, y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_pop_Plot
```



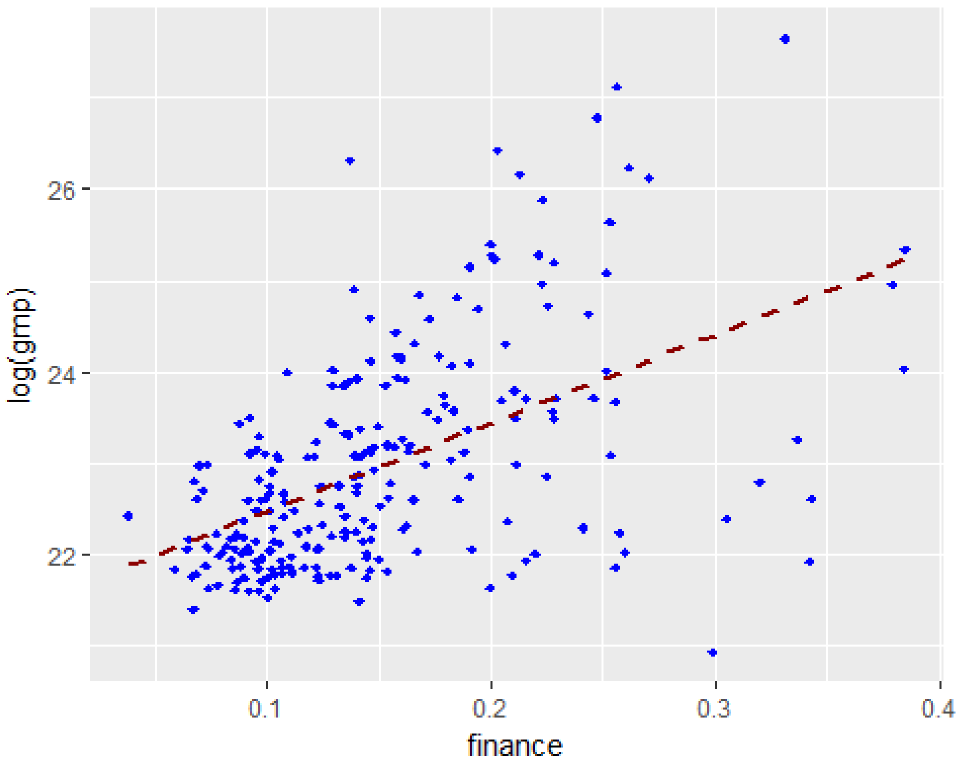
```
#log GMP vs log Pop
log_GMP_vs_log_pop_Plot = ggplot(msadata, aes(x=log(pop), y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_log_pop_Plot
```



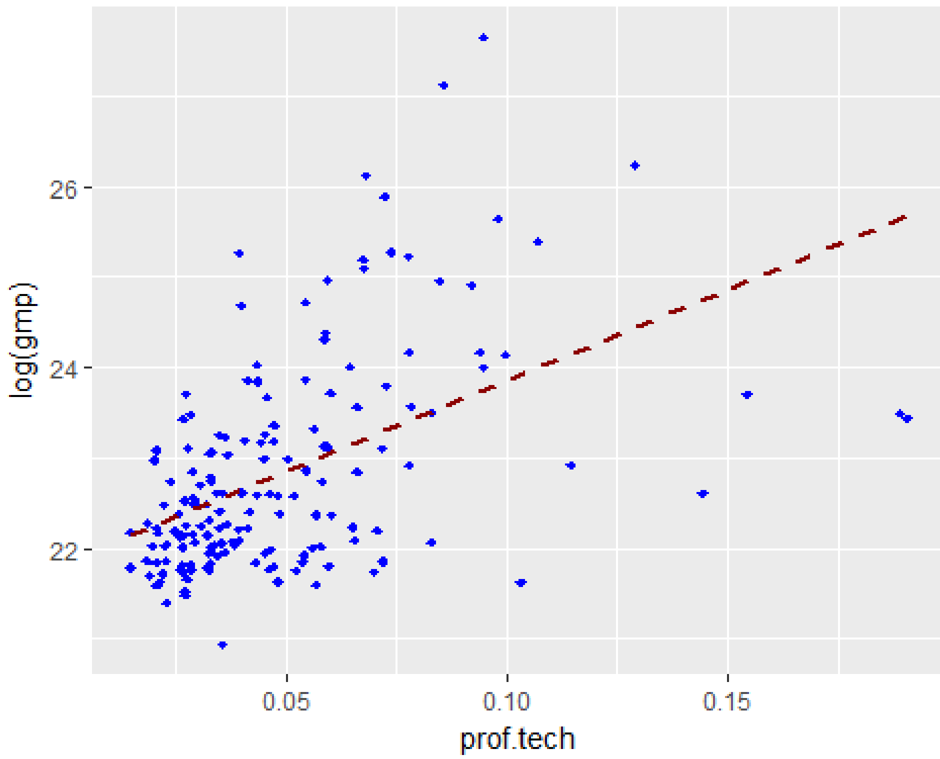
Notes: the log of both variables (GMP and pop) seems to be best choice for visual representation of the data. The other two plots heavily clusters the data on the lefthand side of the plot. log provides a much more expansive view of the data, resulting in a cleaner visual image.

log(GMP) vs. secondary variables

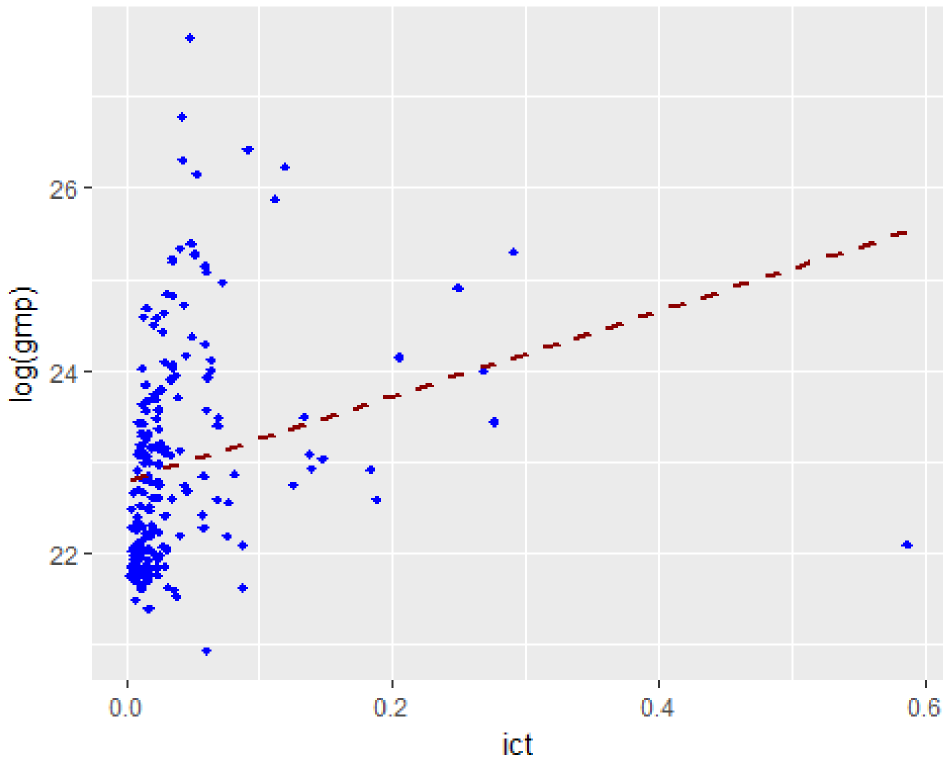
```
#gmp vs finance
log_GMP_vs_finance = ggplot(msadata, aes(x=finance, y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_finance
```



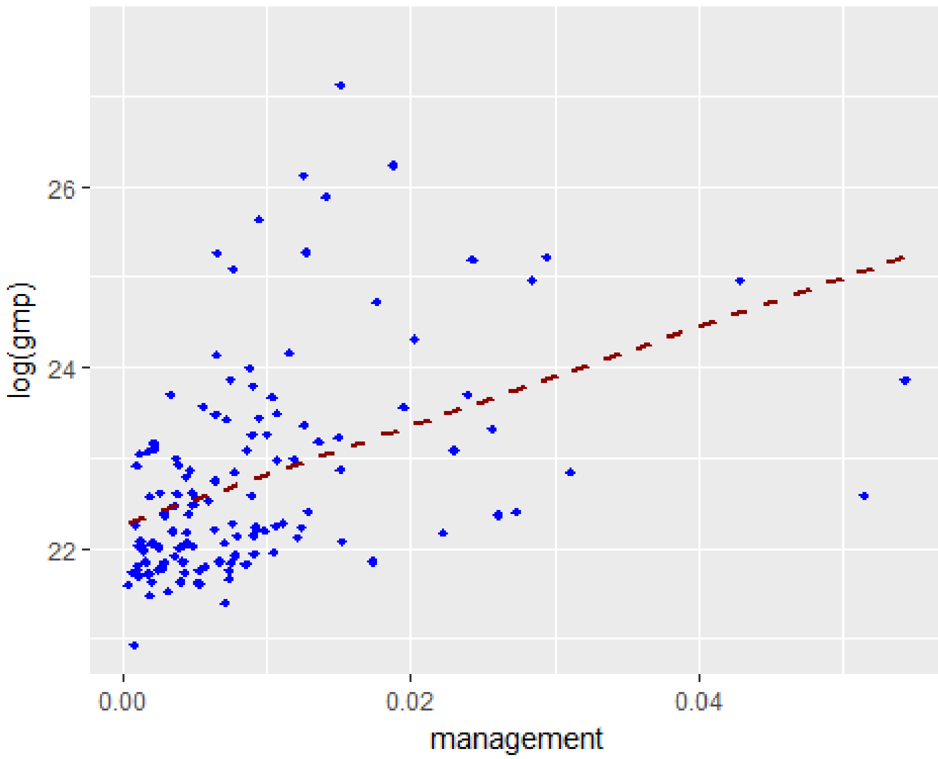
```
#gmp vs prof.tech
log_GMP_vs_prof.tech = ggplot(msadata, aes(x=prof.tech, y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_prof.tech
```



```r
#gmp vs ict
log_GMP_vs_ict = ggplot(msadata, aes(x=ict, y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_ict
```



```r
#gmp vs management
log_GMP_vs_management = ggplot(msadata, aes(x=management, y=log(gmp))) +
  geom_point(shape=18, color="blue") +
  geom_smooth(method=lm, se=FALSE, linetype="dashed", color="darkred")
log_GMP_vs_management
```



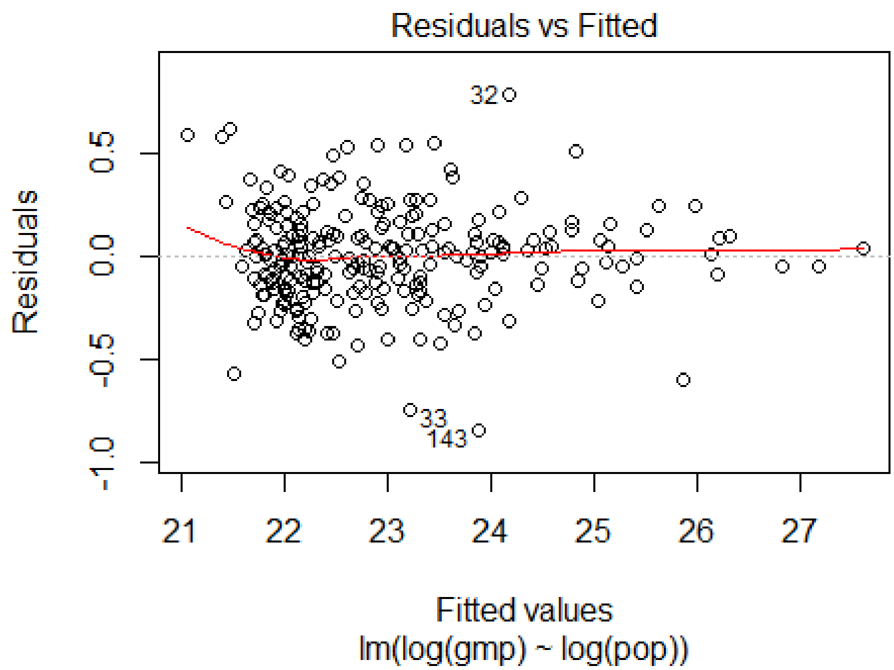
Fitting The Power Law Model

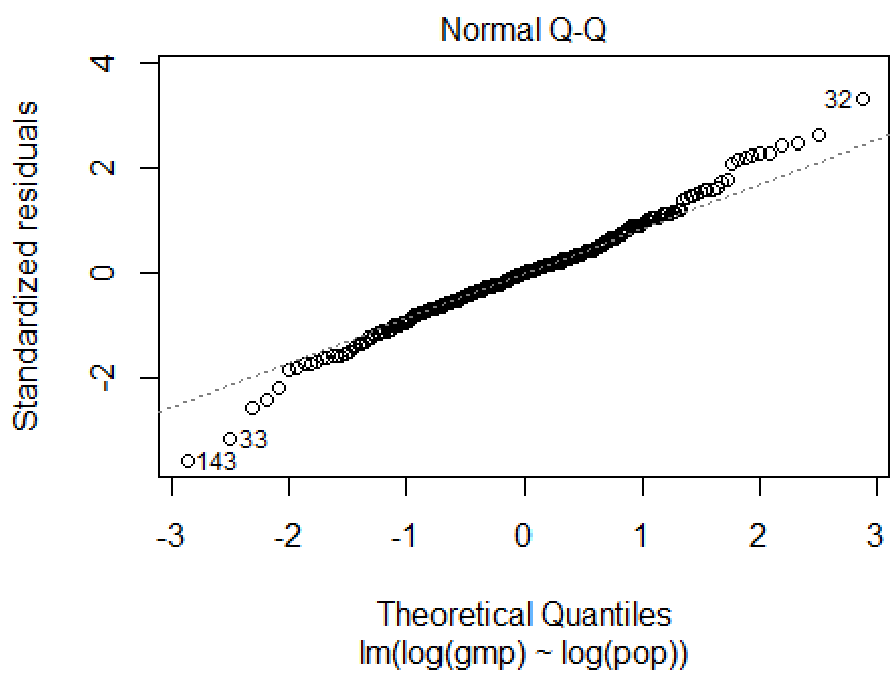
Using lm to linearly regress log(GMP) and log(pcgmp) on the log of the population size

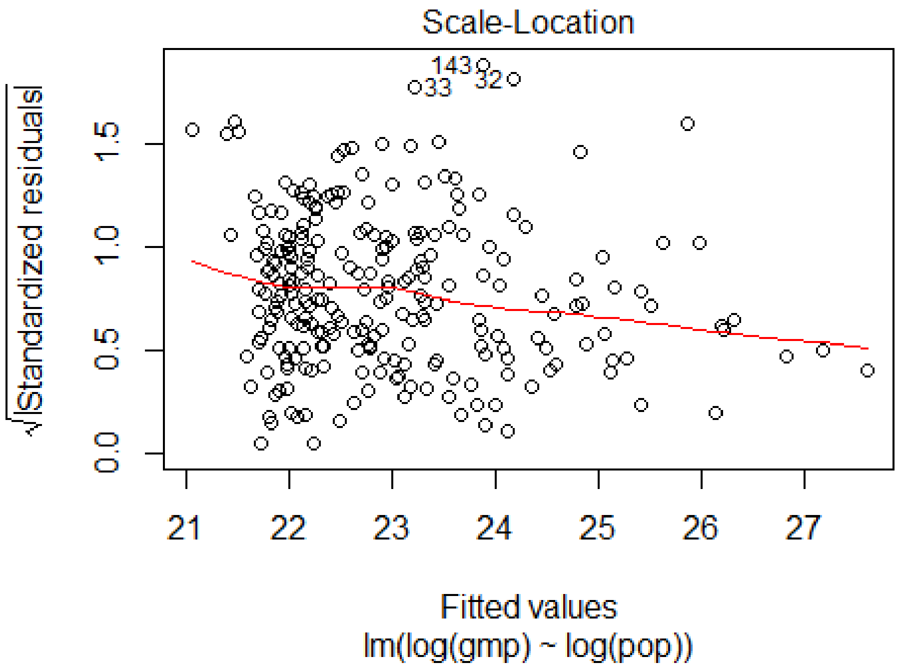
```
lm_GMP_pop = lm(log(gmp)~log(pop), data=msadata)
summary(lm_GMP_pop)

##
## Call:
## lm(formula = log(gmp) ~ log(pop), data = msadata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84226 -0.13993  0.00157  0.12942  0.77779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.79623    0.18350   47.94   <2e-16 ***
## log(pop)     1.12326    0.01449   77.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.238 on 242 degrees of freedom
## Multiple R-squared:  0.9613, Adjusted R-squared:  0.9611
## F-statistic:  6012 on 1 and 242 DF,  p-value: < 2.2e-16

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

##
## Call:
## lm(formula = log(pcgmp) ~ log(pop), data = msadata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84226 -0.13993  0.00157  0.12942  0.77779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.79623    0.18350  47.936  < 2e-16 ***
## log(pop)     0.12326    0.01449   8.509 1.86e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.238 on 242 degrees of freedom
## Multiple R-squared:  0.2303, Adjusted R-squared:  0.2271
## F-statistic:  72.4 on 1 and 242 DF,  p-value: 1.86e-15
```

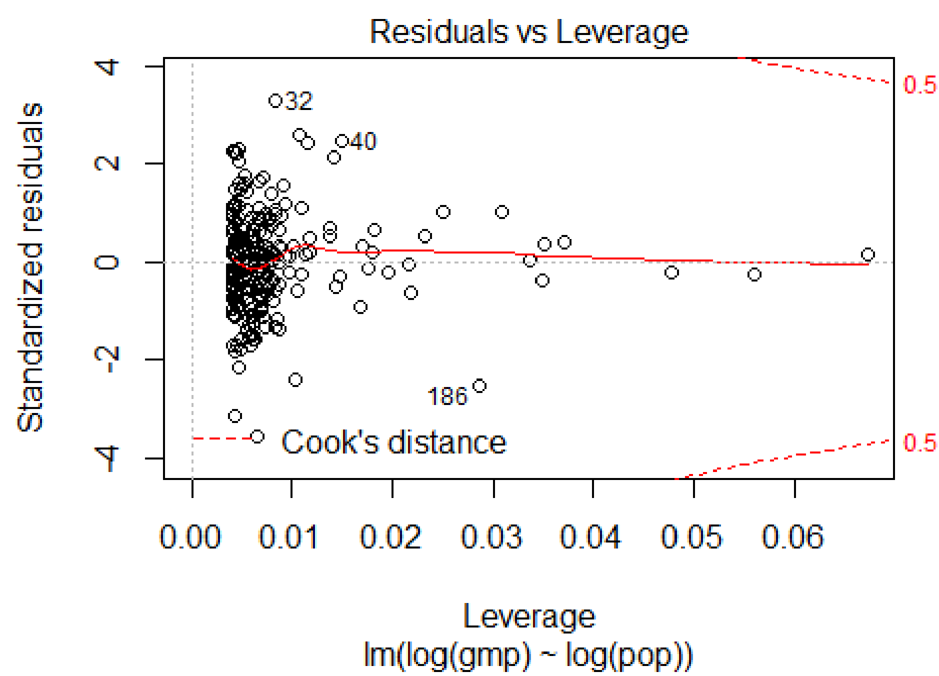
Generate residual plots to scrutinize the credibility of the normal, homoskedastic errors version of the regression model

```r
plot(lm_GMP_pop)
```

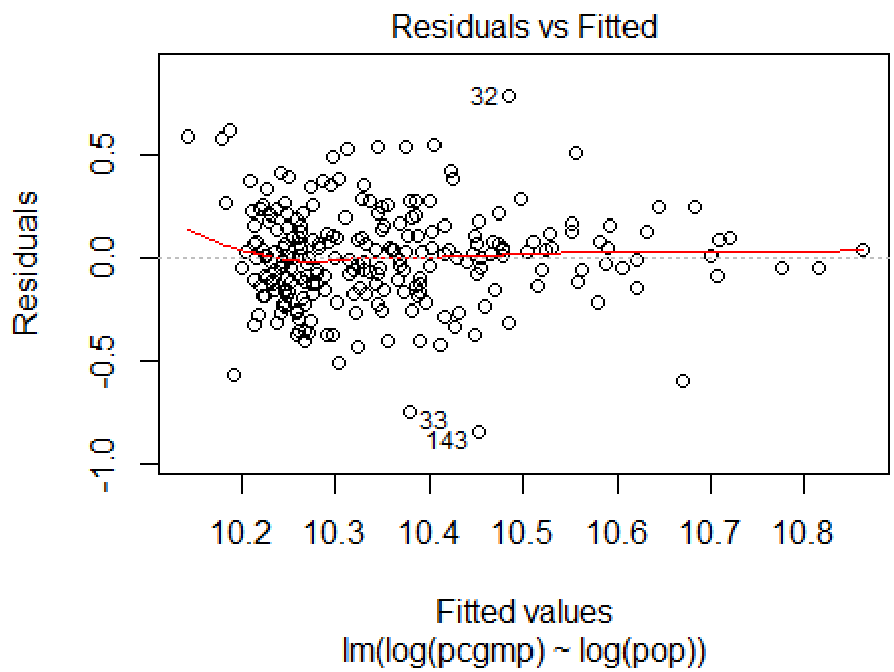


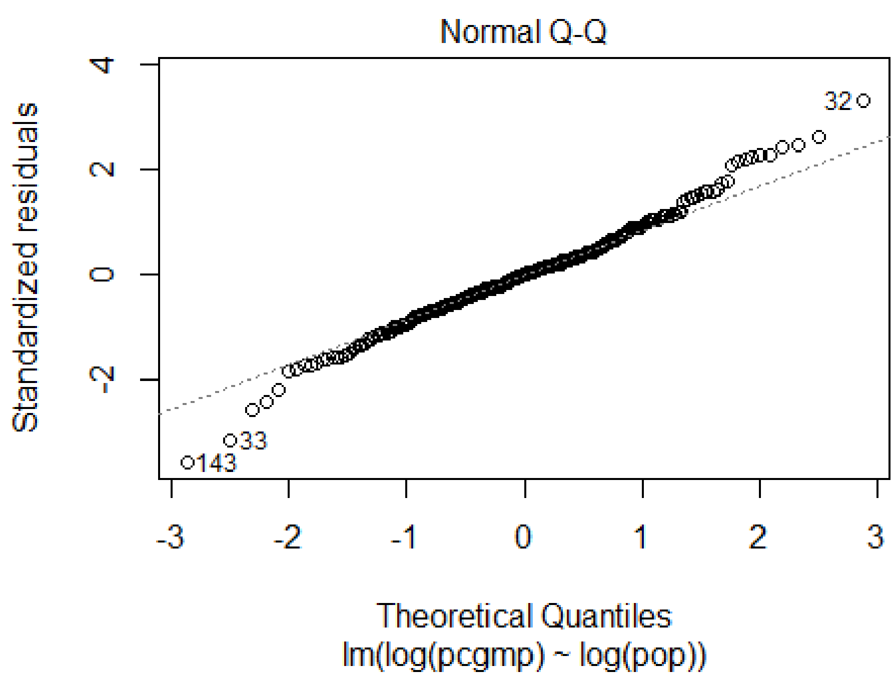


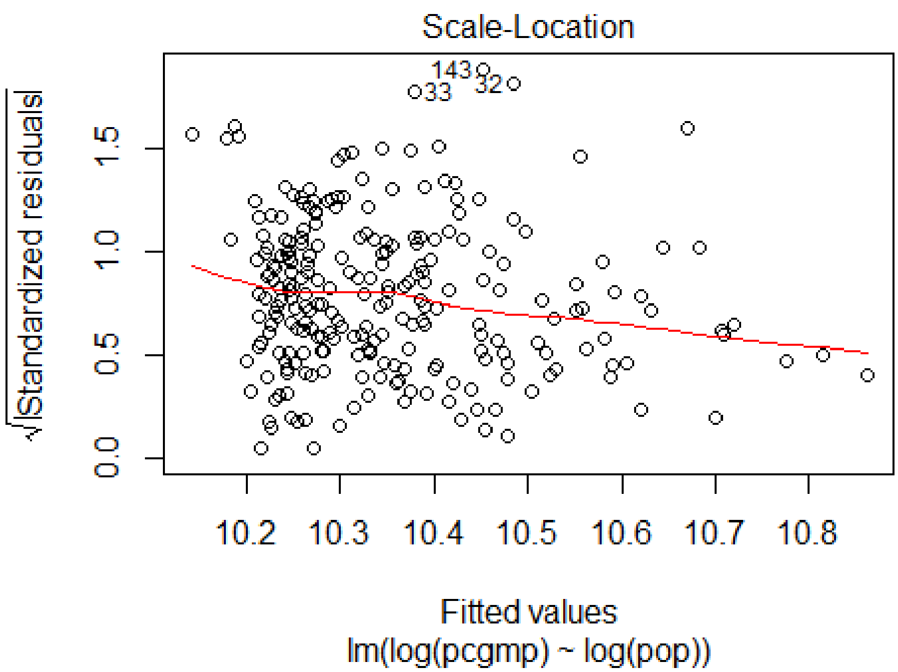




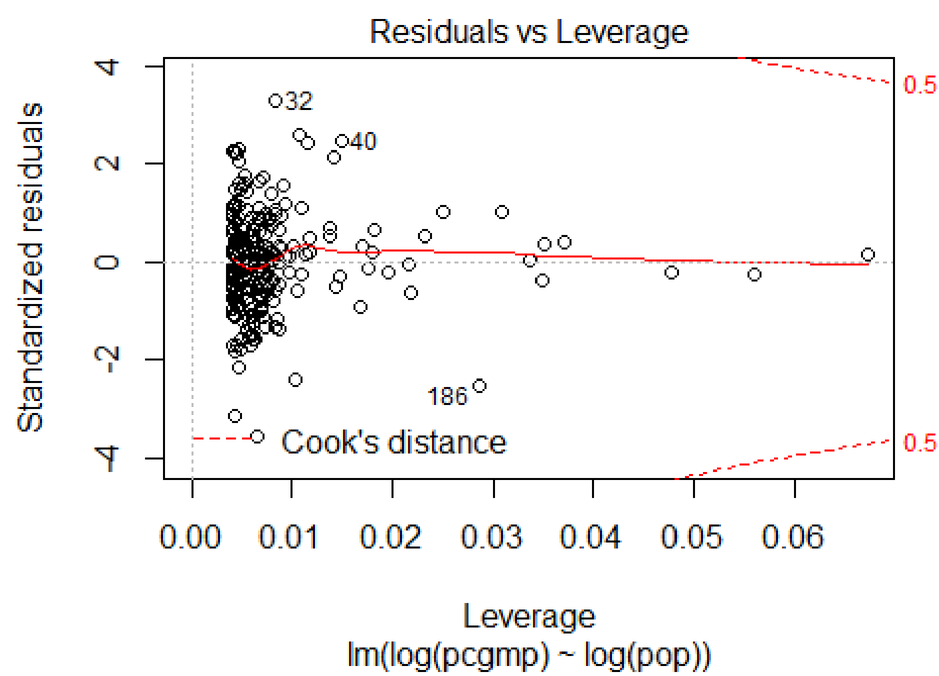

```
plot(lm_pcgmp_pop)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(pcgmp) ~ log(pop))
32
33
143
0.5
0.0
-0.5
-1.0
10.2
10.3
10.4
10.5
10.6
10.7
10.8


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(log(pcgmp) ~ log(pop))
32
33
143
4
2
0
-2
-3
-2
-1
0
1
2
3



Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
143
33
32
10.2
10.3
10.4
10.5
10.6
10.7
10.8
Fitted values
lm(log(pcgmp) ~ log(pop))



Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
32
40
186
Cook's distance
0.5
0.5
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Leverage
lm(log(pcgmp) ~ log(pop))

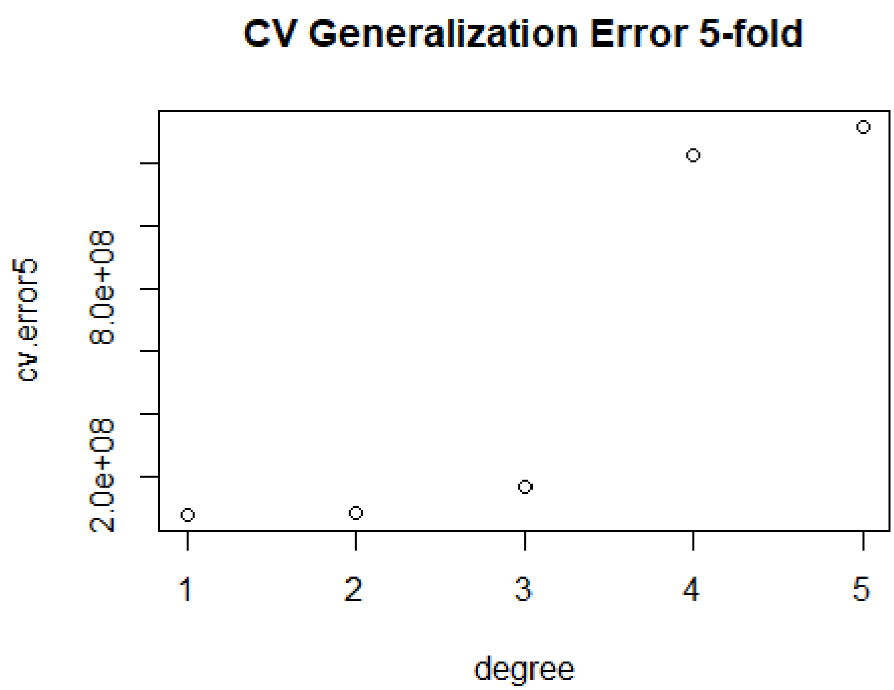
Squared error loss on the log scale

```
fit_loggmp_logpop = lm(log(gmp) ~ log(pop), data = msadata)
loss1 = mean(resid(fit_loggmp_logpop)^2)
loss1

## [1] 0.05619567

fit_loggmp_ict= lm(log(gmp) ~ ict, data = msadata)
loss2 = mean(resid(fit_loggmp_ict)^2)
loss2

## [1] 1.334869

fit.loggmp_finance = lm(log(gmp) ~ finance, data = msadata)
loss3 = mean(resid(fit.loggmp_finance)^2)
loss3

## [1] 1.055425
```

5-fold cross-validation

First applying a generalized linear model to the dataset, and see how the cross-validated error estimate changes with each degree polynomial.

```
glm.fit = glm(pcgmp~pop, data=msadata)
degree=1:5
cv.error5=rep(0,5)
for(d in degree){
  glm.fit = glm(pcgmp~poly(pop, d), data=msadata)
  cv.error5[d] = cv.glm(msadata,glm.fit,K=5)$delta[1]
}

plot(cv.error5, data = msadata, main = "CV Generalization Error 5-fold",
     xlab = "degree", ylab = "cv.error5")
```

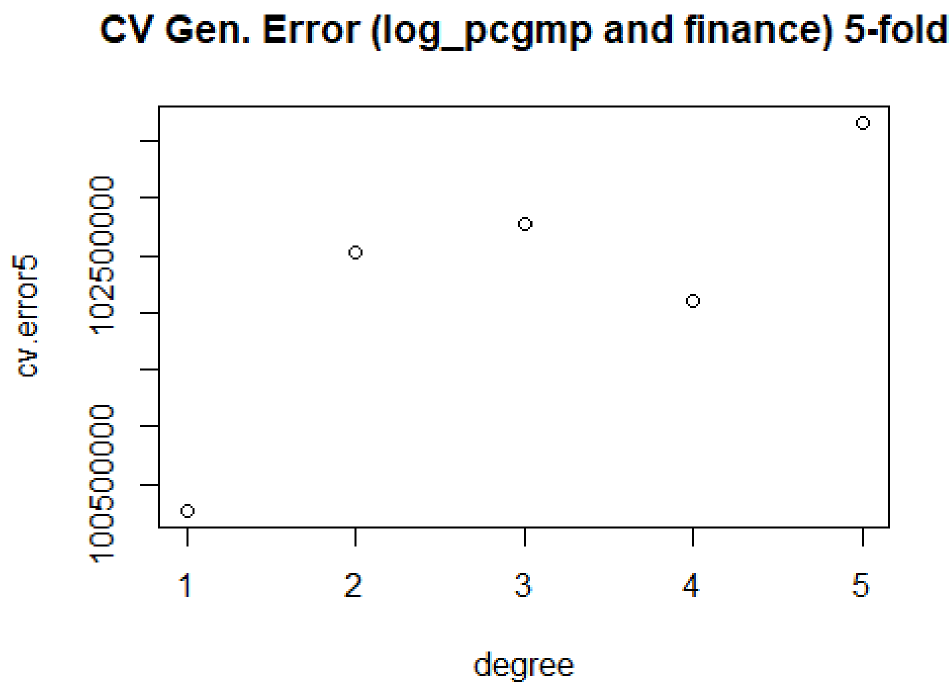


Assessment of Alternate Models (using SEL and 5-fold CV)

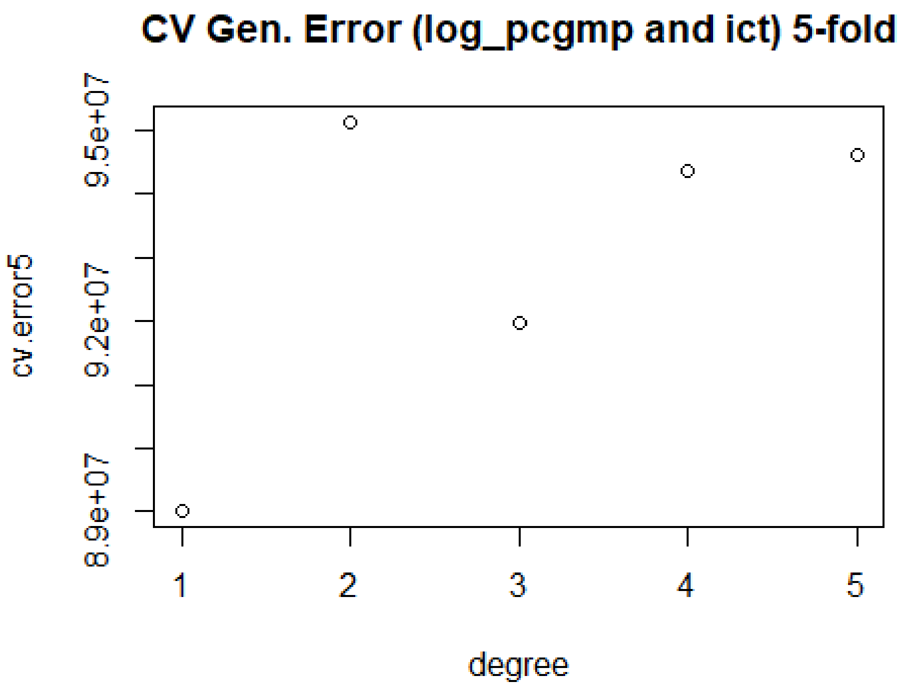
Fit the alternative models and eval using SEL and 5-fold CV

```
pcgmp_finance = merge(pcgmp, finance, by=0)
#pcgmp_finance
omit_pcgmp_finance = na.omit(pcgmp_finance)
#omit_pcgmp_finance

pcgmp1 = omit_pcgmp_finance$x
#pcgmpg1

finance1 = omit_pcgmp_finance$y
#finance1

altM_pcgmp_finance = glm(log(pcgmp1)~finance1, data=omit_pcgmp_finance)
degree=1:5
cv.error5a=rep(0,5)
for(d in degree){
  altM_pcgmp_finance = glm(pcgmp1~poly(finance1, d), data=omit_pcgmp_finance)
  cv.error5a[d] = cv.glm(omit_pcgmp_finance,altM_pcgmp_finance,K=5)$delta[1]
}
```

```r
plot(cv.error5a, data = omit_pcgmp_finance, main = "CV Gen. Error (log_pcgmp
and finance) 5-fold",
     xlab = "degree", ylab = "cv.error5")
```




```r
pcgmp_ict = merge(pcgmp, ict, by=0)
#pcgmp_finance
omit_pcgmp_ict = na.omit(pcgmp_ict)
#omit_pcgmp_finance

pcgmp2 = omit_pcgmp_ict$x
#pcgmpg1

ict1 = omit_pcgmp_ict$y
#finance1

altM_pcgmp_ict = glm(log(pcgmp2)~ict1, data=omit_pcgmp_ict)
degree=1:5
cv.error5b=rep(0,5)
for(d in degree){
  altM_pcgmp_ict = glm(pcgmp2~poly(ict1, d), data=omit_pcgmp_ict)
  cv.error5b[d] = cv.glm(omit_pcgmp_ict,altM_pcgmp_ict,K=5)$delta[1]
}

plot(cv.error5b, data = omit_pcgmp_ict, main = "CV Gen. Error (log_pcgmp and
ict) 5-fold",
     xlab = "degree", ylab = "cv.error5")
```



Standard error loss of the above two models:

```
fit_pcgmp_finance = lm(log(pcgmp1) ~ finance1, data = omit_pcgmp_finance)
loss_pcg_finance = mean(resid(fit_loggmp_logpop)^2)
loss_pcg_finance

## [1] 0.05619567

fit_pcgmp_ict = lm(log(pcgmp2) ~ ict1, data = omit_pcgmp_ict)
loss_pcg_ict = mean(resid(fit_pcgmp_ict)^2)
loss_pcg_ict

## [1] 0.06208128
```

Additional Hypothesis Test on Holdout data

```
holdout = read.csv("http://dept.stat.lsa.umich.edu/~bbh/s485/data/gmp-2006-
holdout.csv", TRUE, ",")

#holdout objects
msaName = holdout$MSA #MSA name (metropolitan statistical areas)
pcgmp_h = holdout$pcgmp #per-capita GMP
pop_h = holdout$pop #population
finance_h = holdout$finance
prof.tech_h = holdout$prof.tech
ict_h = holdout$ict
```

```r
management_h = holdout$management
gmp_h = pop_h * as.double(pcgmp_h)

anova1 = aov(log(gmp_h)~log(pop_h))
anova1$coefficients

## (Intercept)  log(pop_h)
##    8.808162    1.125124

summary(anova1)

##              Df Sum Sq Mean Sq F value Pr(>F)
## log(pop_h)    1 178.83  178.83    3624 <2e-16 ***
## Residuals   120   5.92    0.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova2 = aov(log(gmp_h)~ict_h)
anova2$coefficients

## (Intercept)       ict_h
##   22.734831    9.394557

summary(anova2)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## ict_h        1  20.47  20.468   14.42 0.000274 ***
## Residuals   85 120.65   1.419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 35 observations deleted due to missingness

anova3 = aov(log(gmp_h)~finance_h)
anova3$coefficients

## (Intercept)   finance_h
##    20.97041    13.43103

summary(anova3)

##              Df Sum Sq Mean Sq F value  Pr(>F)
## finance_h     1  62.37   62.37   70.11 1.4e-13 ***
## Residuals   117 104.09    0.89
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 3 observations deleted due to missingness
```