Unemployment Analysis Report

Cindy 300324658

2023-04-06

1.Introduction

The unemployment rate is one of the most important economic indicators used to measure the health of an economy, which can be calculated by unemployed people over the labor force. This report employs several regularization models based on the data set of "unemployment.csv," which includes 574 observations and 7 variables. It compares their performance to find an optimal model for the unemployed rate.

-date : dd/mm/yyyy

-population: population, in thousands

-aggpce: aggregate personal consumption expenditures, in billions

-psrate: personal savings rate, %

-mwunemp: median weeks unemployed, %

-inflation: inflation rate %

-unemply: in thousands

Since there is no labor force data in the data set, we compute the unemployment overpopulation as the unemployment rate in the report, which needs to be more rigorous and treated as the response variable.

2. Preliminary Analysis

2.1 Summary Statistic

We are interested in the unemployed rate and finding a suitable model to predict it. There are 574 observations in the sample data set. The min, max, mean, and median values of variables are shown in the following table.

Metric	Date	population	Aggpce	Psrate	Mwunemp	Inflation	Unemploy	Unemrate
Min	1967-07-01	198712	507.4	1.9	4.0	-2.1	2685	1.322
Median	1991-05-16	253060	3953.6	7.7	7.5	3.5	7494	2.855
Mean	1991-05-17	257189	4843.5	7.9	8.61	4.25	7772	2.989
Max	2015-04-01	320887	12161.5	17	9.10	14.8	15352	5.169

2.2 Time-Serial Analysis

The following figure shows that the unemployed rate changed over the years. The two highest unemployed rate peak, one peak occurred in 1982-12-01, when the economy had a severe recession, and their rate was about 5.17%. The other peak was found in 2009-10-01 when the inflation was negative, and the unemployed rate is about 4.98%. The smooth line shows that unemployment rates generally increased from 1967 to 2015, even though the numbers were waving over time.

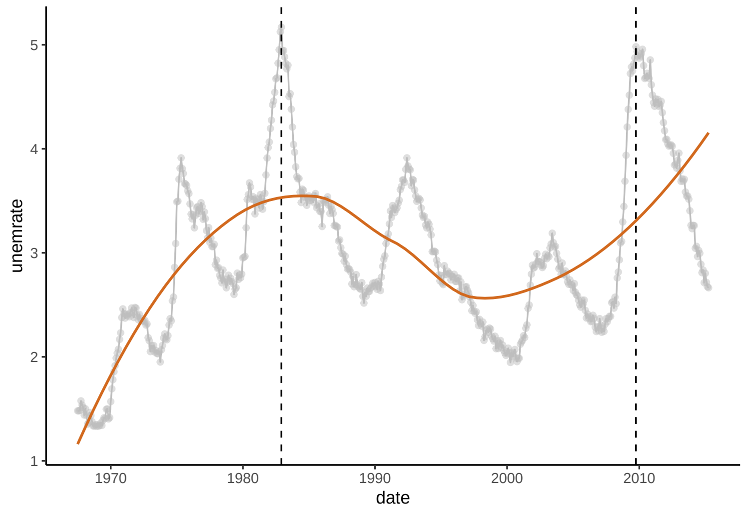


Figure 1: The Change of Unemployment Rate from 1967 to 2015

The other economic factors, such as population and aggregate personal consumption expenditures, increased stability over time. The median weeks unemployed and unemployed number increased with fluctuations.

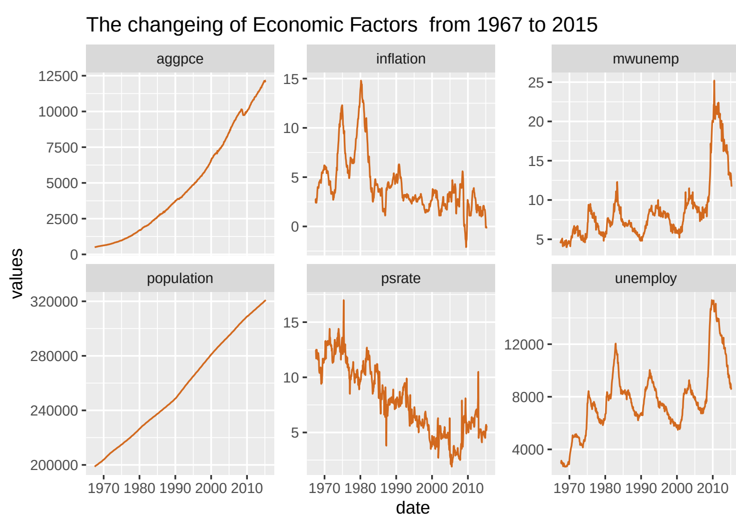


Figure 2: The Change of Economic Factors from 1967 to 2015

2.3 Correlation of Variables

The following correlation matrix shows that the unemploy rate is highly linear correlated with the unemployment number, whose coefficient is about 0.91. The numerate is moderately correlated with mwunemp, as its magnitude is between 0.5 and 0.7. The date and unemrate had a low correlation as the magnitude is between 0.3 and 0.5. The correlation between unemrate and population and aggpce had a little linear correlation, and the correlation coefficient was less than 0.3. The psrate and inflation had little negative correlation with the unemploy rate. Meanwhile, some predictor variables are highly correlated, such as population and aggpce, aggpce and psrate.

As we calculate the unemploy rate by the unemploy number over the population, the independent variable, unemployment, is highly correlated with the dependent variable. It makes no sense to use unemployment to predict the unemploy rate. We should not include the unemployment variable for the following analysis process.

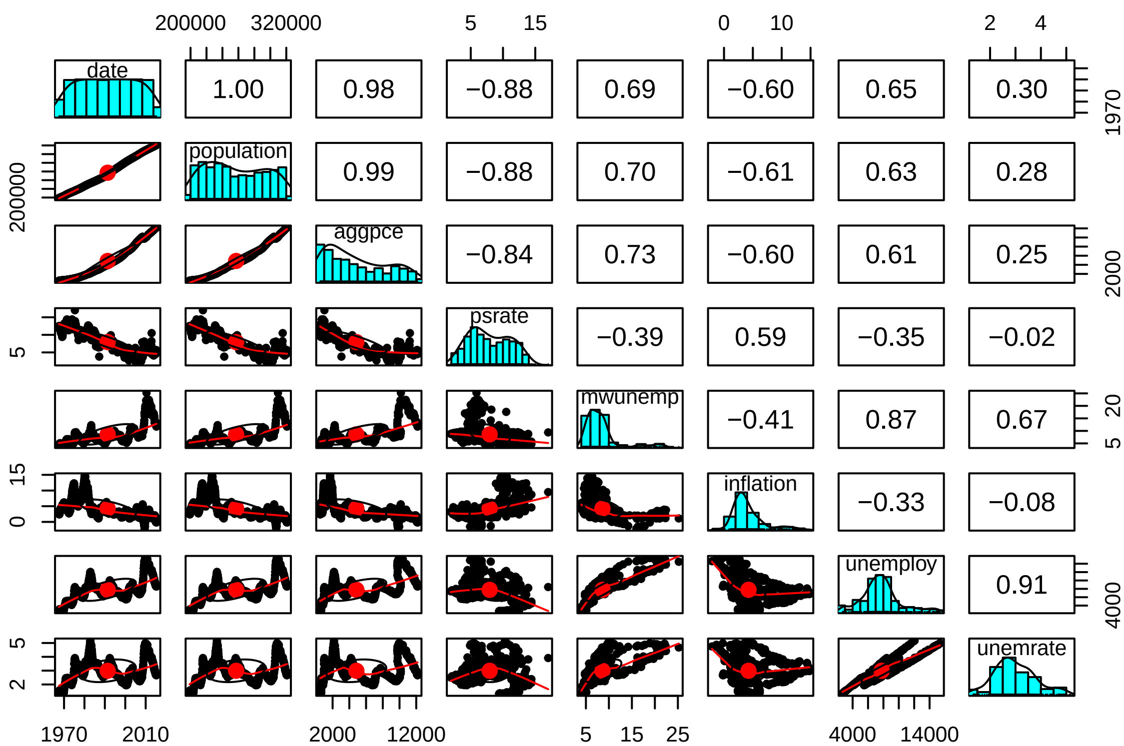


Figure 3: The Correlation Matrix Figure

3. Methodology

The report employs linear (LM) and regularized regression models to predict the unemploy rate and compare their performance. When the assumptions required for LMs are met, the estimated coefficients are unbiased and have the lowest variance. While from the previous correlation matrix figure, we can see that the response variable and predictors are not normally distributed, and some independent variables are highly correlated, which does not meet the assumption and may increase sample error. We can employ regularized regression to constrain and weighted the estimated coefficients toward zero to reduce the variance and decrease sample error.

The most common regularization method include :

- ridge regression : controls the estimated coefficients by adding $\lambda \sum_{j=1}^{p} \beta_j^2$ to the objective functions $minmize(SSE + \lambda \sum_{j=1}^{p} \beta_j^2)$.
- lasso regression: controls the estimated coefficients by adding $\lambda \sum_{j=1}^{p} |\beta_j|$ to the objective functions
- elastic-net: combines the two penalties $minmize(SSE + \lambda \sum_{j=1}^{p} \beta_j^2 + \lambda \sum_{j=1}^{p} |\beta_j|)$.

4. Modeling

We experimented with linear, ridge, lasso, and elastic-net regression to predict the unemploy rate and compare their performance. As the ranges of the predictors are very different, we scale the data before using regression algorithms.

4.1 LM Regression

There are a few ways to do linear regression. We can employ one variable or all variables to predict. The following table shows that the linear model with all 6 variables performs the best, as the Cp and BIC are the least and the Adjusted R square are great. The model can be described as

$$unemprate = 4.496*date - 2.068*population - 1.949*aggpce + 0.344*psrate + 0.99*munemp + 0.112*inflation$$

Variable_No	Cp	R2	Adj_r2	BIC
1	732.27474	0.4484839	0.4470982	-226.0508
2	473.33873	0.5760330	0.5738972	-325.2655
3	93.12321	0.7628652	0.7610687	-551.6847
4	35.78779	0.7918691	0.7897615	-597.8780
5	20.43859	0.8003497	0.7978160	-608.5263
6	7.00000	0.8078962	0.8049634	-617.9476

4.2 Ridge Regression

We employed the ridge regression, and the mathematical formula can be described as

$$unemprate = 4.348*date - 2.412*population - 1.996*aggpce + 0.346*psrate + 0.99*munemp + 0.112*inflation$$

The optimal λ is about -2.7048. The following figure shows the 10-fold CV MSE for a ridge model and coefficients for a ridge regression model as λ change.The red dot line represents the optimal λ.

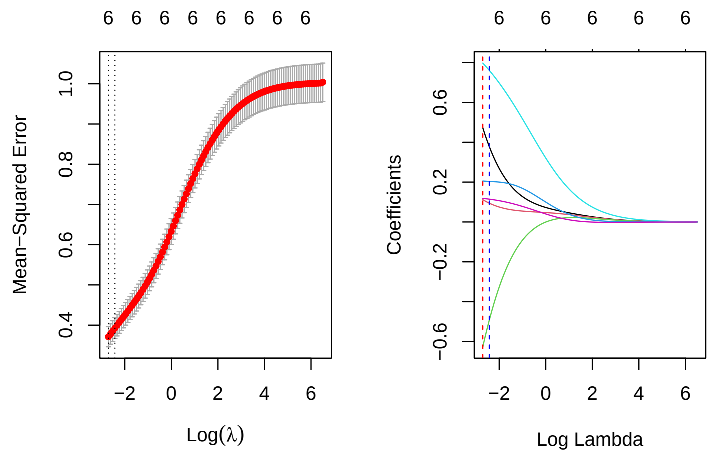


Figure 4: The Ridge Penalty

4.3 Lasso Regression

We employed the lasso regression, and the mathematical formula can be described as

$$unemprate = 4.270*date - 2.312*population - 2.019*aggpce + 0.346*psrate + 0.990*munemp + 0.112*inflation$$

The optimal λ is about -9.334. The following figure shows the 10-fold CV MSE for a lasso model and coefficients for a lasso regression model as λ change.

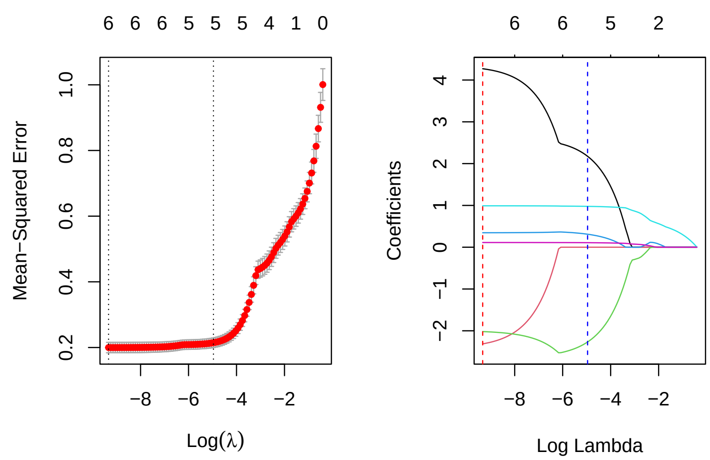


Figure 5: The Lasso Penalty

4.4 Elastic-net Regression

We employed three different Elastic-net regressions. The figure 6,figure 7 and figure 8 show their 10-fold CV MSE for Elastic models and coefficients for Elastic-net regression model as λ change. The coefficients and optimal lambda are shown in the following table. We can see that difference in coefficients is not significant.

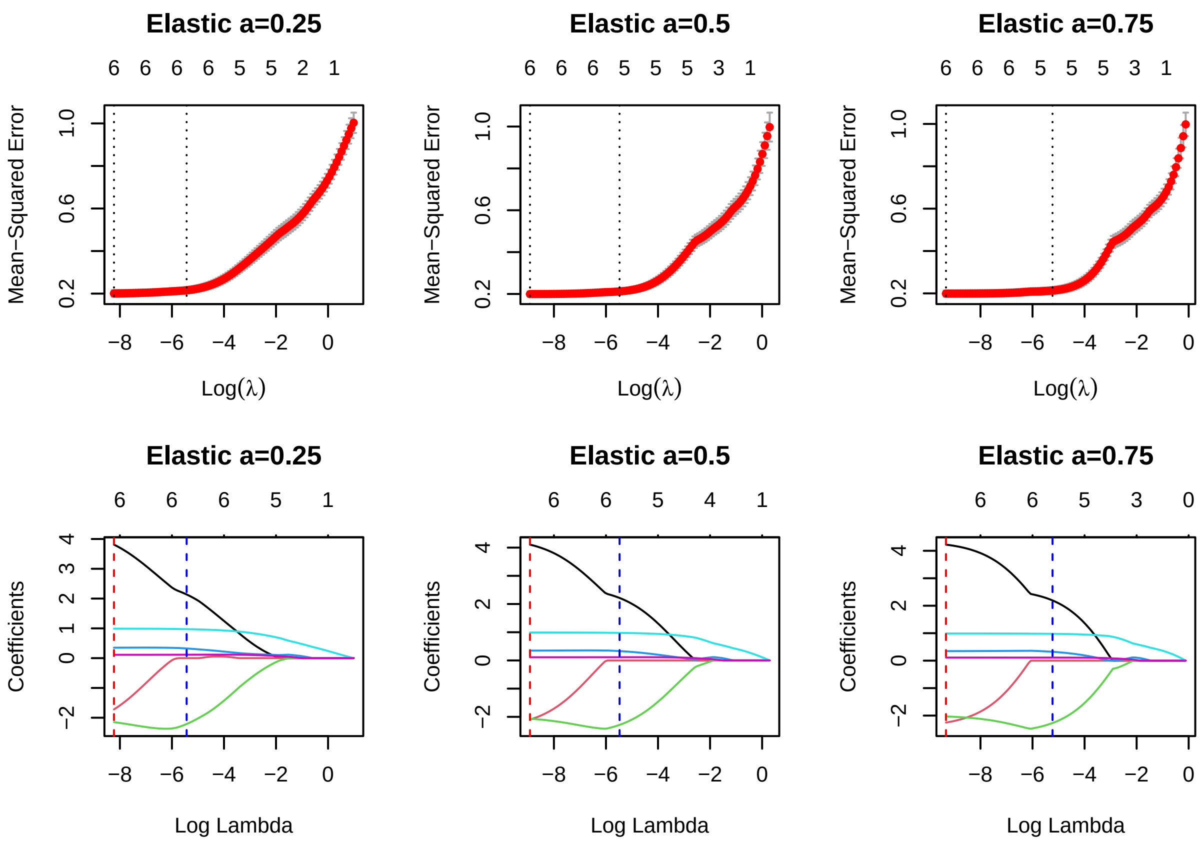


Figure 6: The Elastic Penalty

Variable	Elastic0.25	Elastic0.5	Elastic0.75
Intercept	0	0	0
date	3.8116	4.1059	4.2248
population	-1.7203	-2.098	-2.2524
aggpce	-2.149	-2.067	-2.0323
psrate	0.3514	0.3485	0.3471
mwunemp	0.9898	0.9903	0.9904
inflation	0.1126	0.1124	0.1124
lambda	-8.2262	-8.9194	-9.3249

4.5 Experiment Models

The following table shows the coefficients of six different algorithms. We can see that the ridge regression model's coefficients significantly differ from the other models.

	OLS	Ridge	Lasso	Elastic0.25	Elastic0.5	Elastic0.75
(Intercept)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
date	4.496351	0.468637	4.270166	3.811588	4.105899	4.224833
population	-2.608364	0.115470	-2.311601	-1.720292	-2.098036	-2.252384
aggpce	-1.948970	-0.625030	-2.018694	-2.148954	-2.067016	-2.032295
psrate	0.344155	0.205976	0.346582	0.351389	0.348487	0.347146
mwunemp	0.990380	0.796479	0.990434	0.989803	0.990281	0.990423
inflation	0.112440	0.118167	0.112378	0.112573	0.112449	0.112413

4.6 Model Performance Measurement

The following table shows that the linear model performs the best in the test data set among the previous 6 models, as its MSE is about 0.2466, the least.

Model	MSE
OLS	0.2465653
Ridge	0.4055879
Lasso	0.2474696
Elastic0.25	0.2498947
Elastic0.5	0.2482531
Elastic0.75	0.2476791

4.7 Model optimization

However, the economic factors interact with each other, so we employ the interaction of variables as predictors. The model can be described as

$$Y = \beta_0 + \beta_1 x_1 + \beta_1 x_2 + ... + \beta_6 x_6 + \beta_7 x_1 x_2 + \beta_8 x_1 x_3 + ...$$

The following table shows the coefficient (β_s) and the two-way variables.

	x
(Intercept)	-1.3757154
date	8.4332085
population	-2.7995356
aggpce	-6.0893606
psrate	0.3789069
mwunemp	1.2460114
inflation	0.0852220
date:population	4.0815250
date:aggpce	-20.2991788
date:psrate	-2.3262440
date:mwunemp	4.4787012
date:inflation	0.7920764
population:aggpce	17.7794282
population:psrate	3.4486080
population:mwunemp	-4.6330980
population:inflation	0.0767074
aggpce:psrate	-1.2305906
aggpce:mwunemp	-0.3796066

	x
aggpce:inflation	-0.6193308
psrate:mwunemp	-0.0077138
psrate:inflation	0.0675634
mwunemp:inflation	-0.2179177

The following table indicates that the two-way interaction linear model(LM2) perform better than the linear model. The LM2 MSE value of the test data is about 0.084 less than the MSE of the LM model.

Measurement	LM	LM.Interaction.2
Residual standard error	0.4416295	0.2467357
R-squared	0.8078962	0.9423256
Adjusted R-squared	0.8049634	0.9391215
MSE (test)	0.2465653	0.0839820

5. Conclusion

Since the parameters of the analyzed data set are not greater than the observations, and there are not many redundant predictors, the regularization regression models did not improve the performance of the linear model. The two-way interaction linear model performs the best, then is the full variable linear model. We should use the LM2 model to predict the unemployment rate (two-way interaction linear model).

Appendix

```
library(lubridate)
library(tidyr)
library(dplyr)
library(ggplot2)
library(knitr)

df<-read.csv("umemployment.csv")
head(df)
```

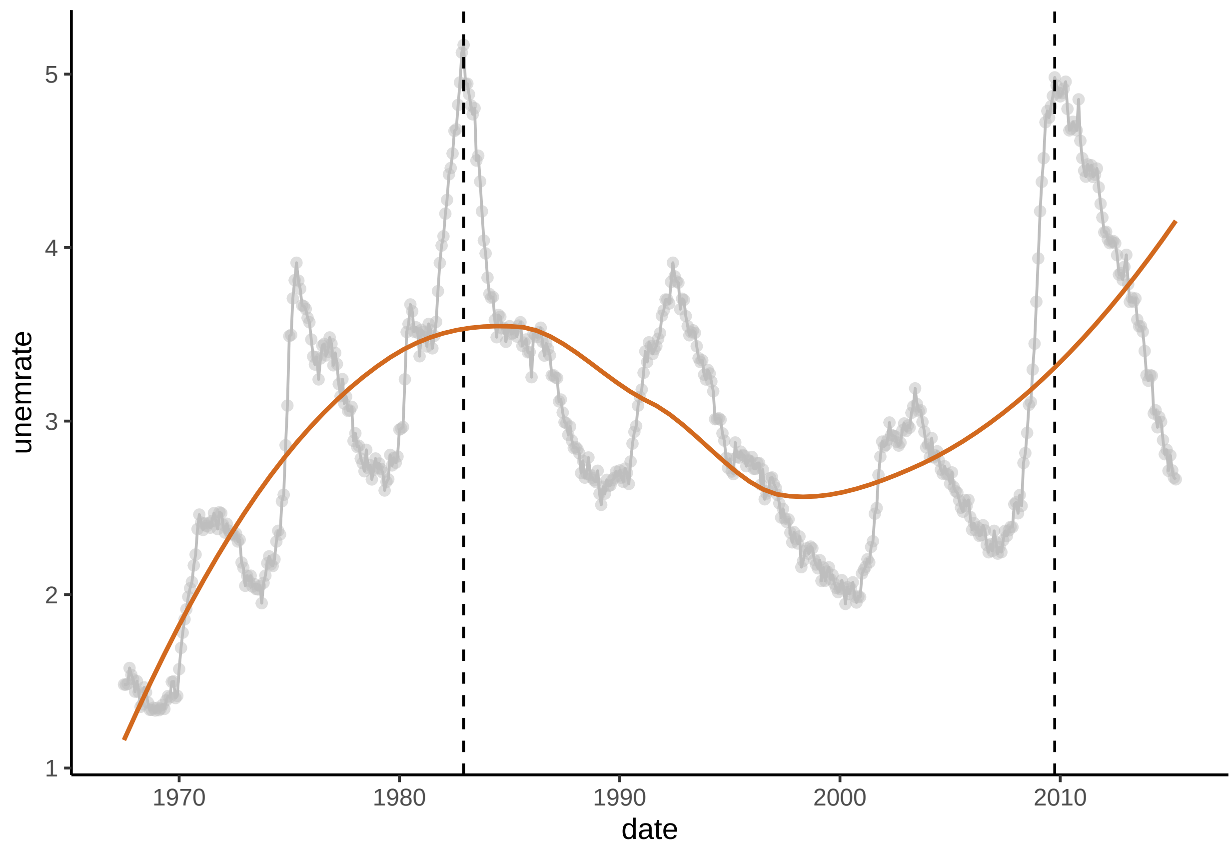
```
##         date population aggpce psrate mwunemp inflation unemploy
## 1 01/07/1967     198712  507.4   12.5     4.5       2.8     2944
## 2 01/08/1967     198911  510.5   12.5     4.7       2.4     2945
## 3 01/09/1967     199113  516.3   11.7     4.6       2.8     2958
## 4 01/10/1967     199311  512.9   12.5     4.9       2.4     3143
## 5 01/11/1967     199498  518.1   12.5     4.7       2.7     3066
## 6 01/12/1967     199657  525.8   12.1     4.8       3.0     3018
```

```
str(df)
```

```
## 'data.frame':    574 obs. of  7 variables:
##  $ date      : chr  "01/07/1967" "01/08/1967" "01/09/1967" "01/10/1967" ...
##  $ population: int  198712 198911 199113 199311 199498 199657 199808 199920 200056 200208 ...
##  $ aggpce    : num  507 510 516 513 518 ...
##  $ psrate    : num  12.5 12.5 11.7 12.5 12.5 12.1 11.7 12.2 11.6 12.2 ...
##  $ mwunemp   : num  4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ inflation : num  2.8 2.4 2.8 2.4 2.7 3 3.6 4 3.9 3.9 ...
##  $ unemploy  : int  2944 2945 2958 3143 3066 3018 2878 3001 2877 2709 ...
```

```
df$date<-dmy(df$date)

df<- df %>% mutate(unemrate=unemploy/(population)*100)

# check the unemployment rate changing with the time

df %>% ggplot(aes(date,unemrate))+
geom_line(color="grey")+
geom_point(color="grey",alpha= 0.5)+
theme_classic()+
  geom_vline(xintercept = as.numeric(ymd("1982-12-01")),linetype="dashed")+
  geom_vline(xintercept = as.numeric(ymd("2009-10-01")),linetype="dashed")+
  geom_smooth(colour="chocolate",se=FALSE,linetype=1,size=0.8)
```

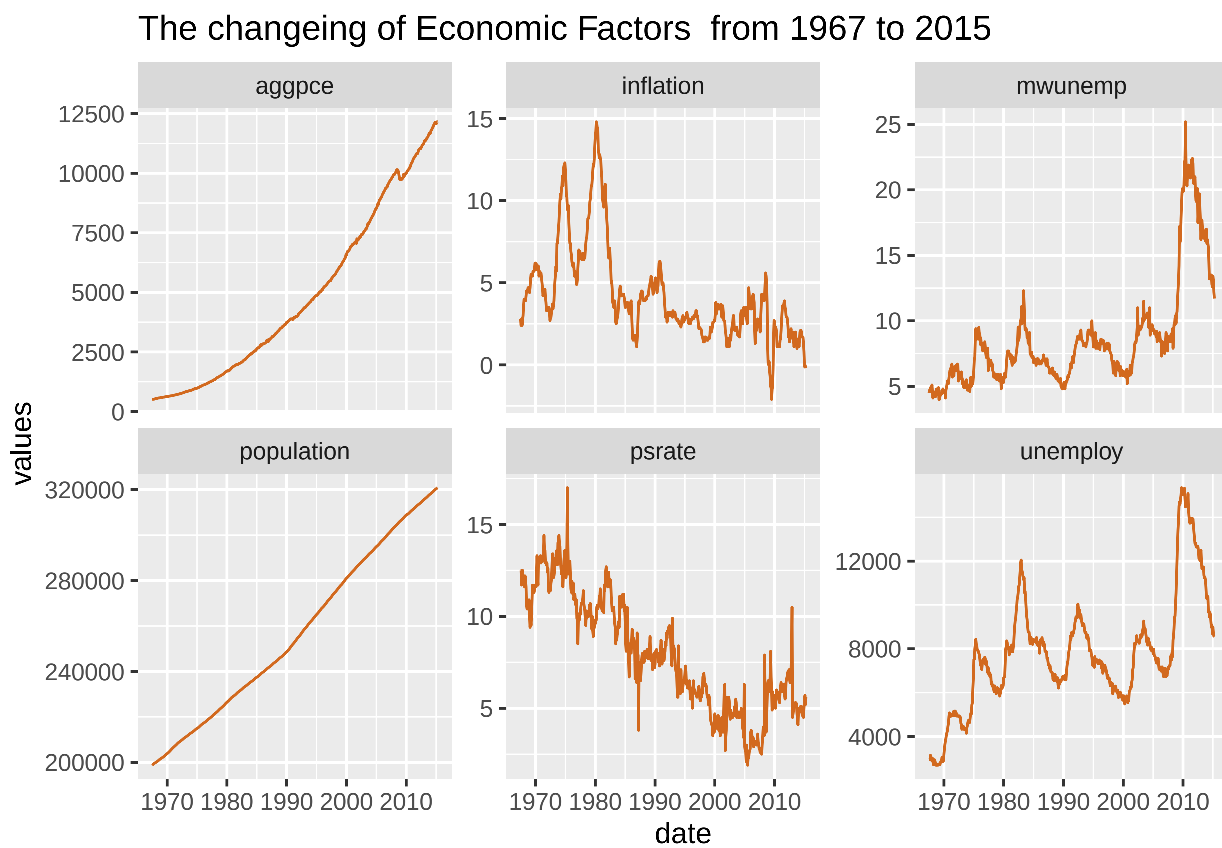
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



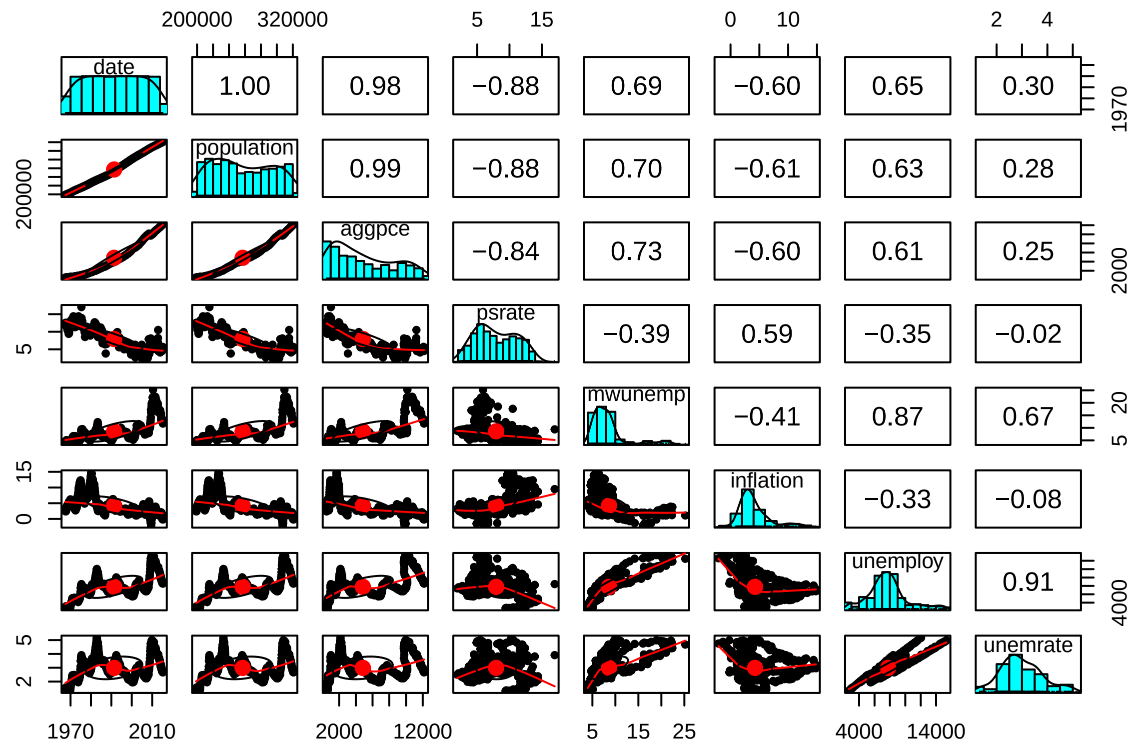

```
tail(df%>% arrange(unemrate))
```

```
##           date population  aggpce psrate mwunemp inflation unemploy unemrate
## 569 1983-02-01     233473  2183.1   10.5     9.8       3.5    11545 4.944897
## 570 1982-10-01     232816  2130.7   10.7     9.7       5.1    11529 4.951979
## 571 2010-04-01     309191 10106.9    5.6    22.1       2.2    15325 4.956483
## 572 2009-10-01     308189  9924.6    5.4    18.9      -0.2    15352 4.981359
## 573 1982-11-01     232993  2154.7   10.3    10.0       4.6    11938 5.123759
## 574 1982-12-01     233160  2167.4   10.3    10.2       3.8    12051 5.168554
```

```
# check the variable changing with the time

df_time<-df%>%pivot_longer(c("population","aggpce","psrate","mwunemp","inflation","unemploy"),names_to =

df_time %>% ggplot(aes(date,values))+
  geom_line(color="chocolate")+
  facet_wrap(~levels,scale="free_y",ncol=3)+
  ggtitle("The changeing of Economic Factors  from 1967 to 2015")
```




```
# correlation check

library(psych)
pairs.panels(df, method = "pearson")
```




```
df<-df[,-7]
```

```
# split data

library(rsample)
set.seed(311)
split<-initial_split(df,prop = 0.7,strata="unemrate")
dftrain<-training(split)
dftest<-testing(split)


# best-subset selection

library(leaps)

# scale the data
dftrain[,-1]<-scale(dftrain[,-1])

dftrain[,1]<-scale(dftrain[,1])

bestsub<-regsubsets(unemrate~.,data=dftrain)

# performance measures

sub_m<-cbind(Variable_No= 1:6,
Cp = summary(bestsub)$cp,
```

```
R2 = summary(bestsub)$rsq,
Adj_r2 = summary(bestsub)$adjr2,
BIC =summary(bestsub)$bic
)

#sub_m

knitr::kable(sub_m,full_width=FALSE)
```

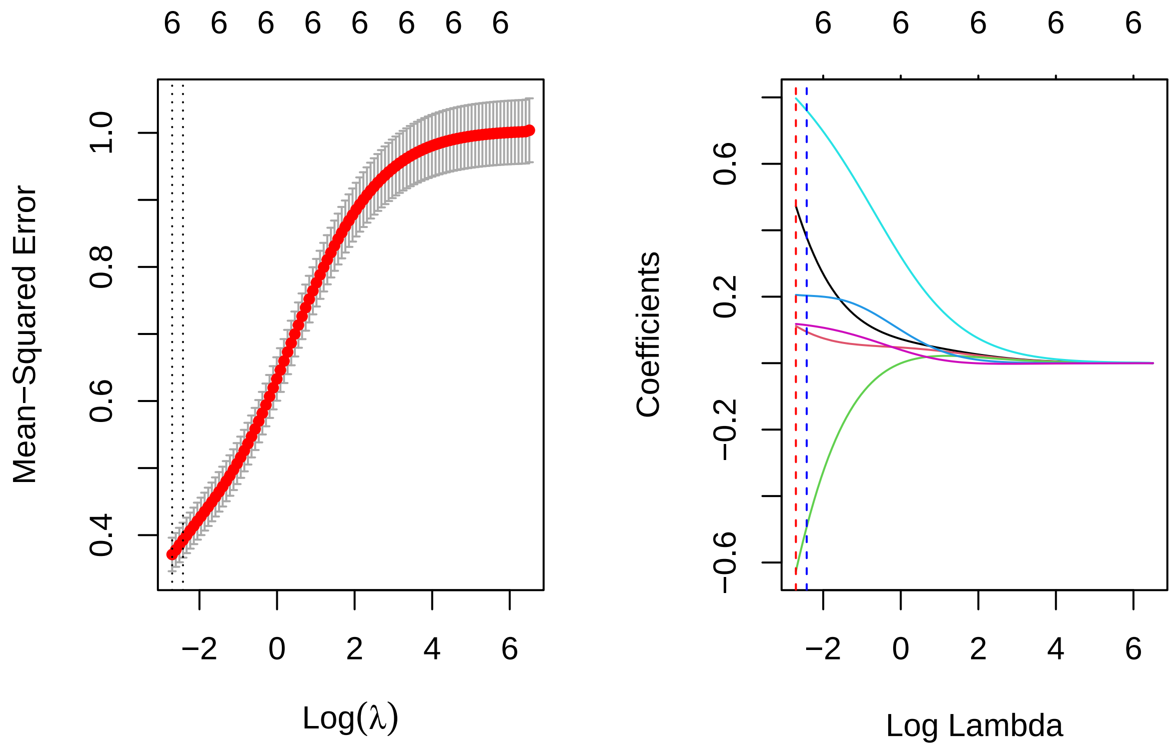
Variable_No	Cp	R2	Adj_r2	BIC
1	732.27474	0.4484839	0.4470982	-226.0508
2	473.33873	0.5760330	0.5738972	-325.2655
3	93.12321	0.7628652	0.7610687	-551.6847
4	35.78779	0.7918691	0.7897615	-597.8780
5	20.43859	0.8003497	0.7978160	-608.5263
6	7.00000	0.8078962	0.8049634	-617.9476

```
# variables full model

lm<-lm(unemrate~.,data=dftrain)
lm$coefficients
```

```
##   (Intercept)          date    population        aggpce        psrate
##  6.723046e-16  4.496351e+00 -2.608364e+00 -1.948970e+00  3.441554e-01
##       mwunemp     inflation
##  9.903804e-01  1.124397e-01
```

```
# Ridge
library(glmnet)
set.seed(311)

# define the predictors

X<- model.matrix(unemrate~.,data=dftrain)

# define the responsible variable

Y<-dftrain[,"unemrate"]

# find the optimal lambda

cv.lambda<-cv.glmnet(x=X,y=Y,alpha=0)


# visualize the lambda and related MSE

par(mfrow=c(1,2))

plot(cv.lambda)

# get the min MSE
```

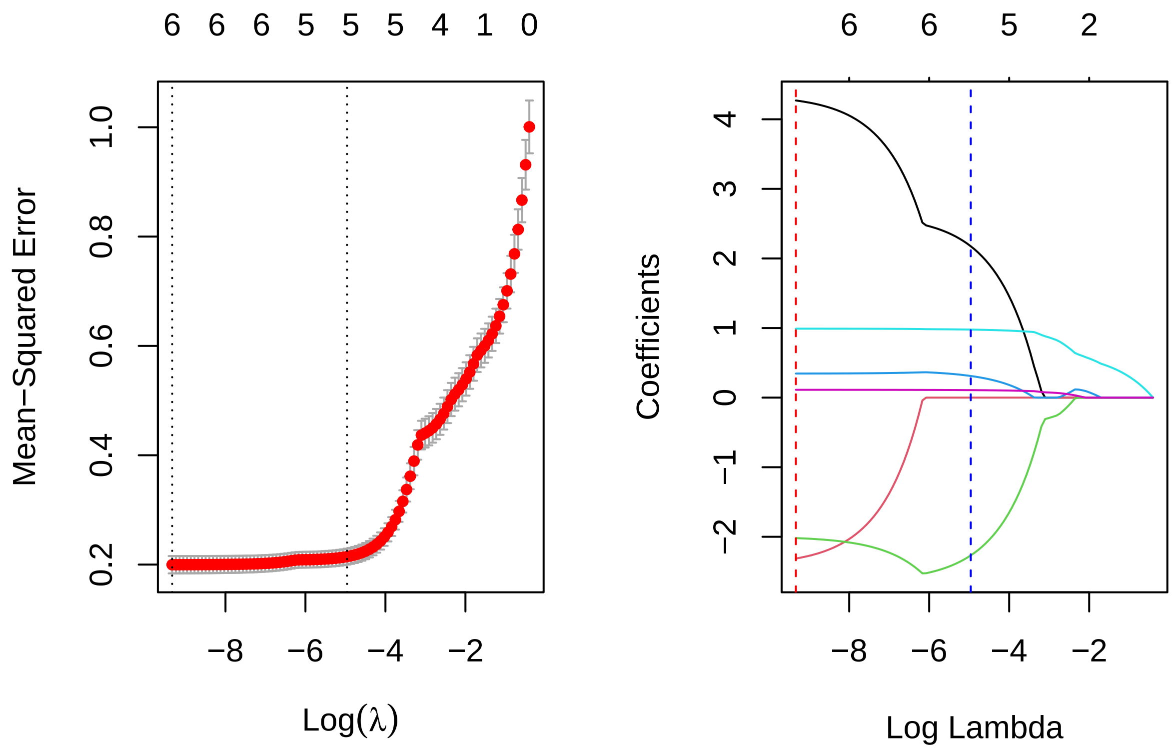
```r
# log(cv.lambda$lambda.min)

# ridge path

plot(cv.lambda$glmnet.fit,"lambda",label=FALSE)
abline(v=log(cv.lambda$lambda.min),col="red",lty="dashed")
abline(v=log(cv.lambda$lambda.1se),col="blue",lty="dashed")
```




```r
# get the ridge model

ridgemin<-cv.lambda$lambda.min
ridge<-glmnet(x=X,y=Y,alpha=0,lambda=ridgemin)

ridge$beta
```

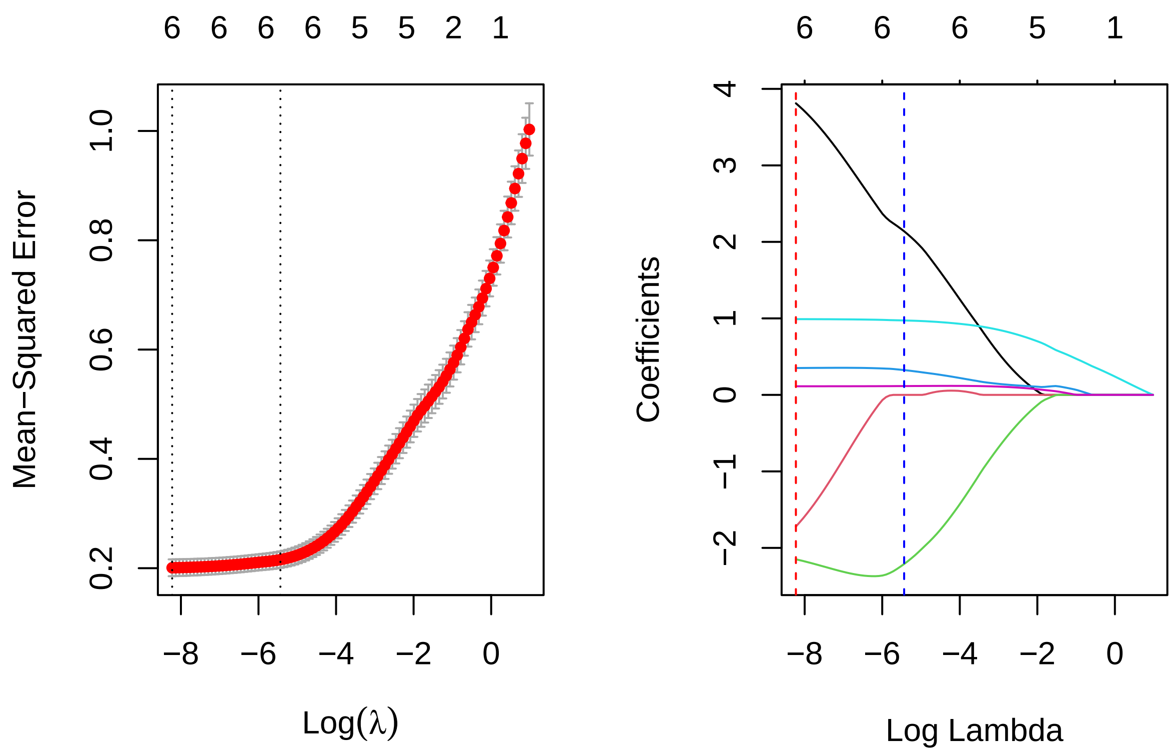
```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept)  .
## date          0.4686370
## population    0.1154697
## aggpce       -0.6250304
## psrate        0.2059756
## mwunemp       0.7964788
## inflation     0.1181670
```

```
# Lasso
set.seed(311)

cv.lambda.lasso<-cv.glmnet(x=X,y=Y,alpha=1)

par(mfrow=c(1,2))

plot(cv.lambda.lasso)

# get the optimal lambda
# log(cv.lambda.lasso$lambda.min)

# lasso path

lassomin<-cv.lambda.lasso$lambda.min
lasso<-glmnet(x=X, y=Y,alpha=1, lambda = lassomin)
# lasso$beta

plot(cv.lambda.lasso$glmnet.fit,"lambda", label = FALSE)
abline(v=log(cv.lambda.lasso$lambda.min),col="red",lty="dashed")
abline(v=log(cv.lambda.lasso$lambda.1se),col="blue",lty="dashed")
```



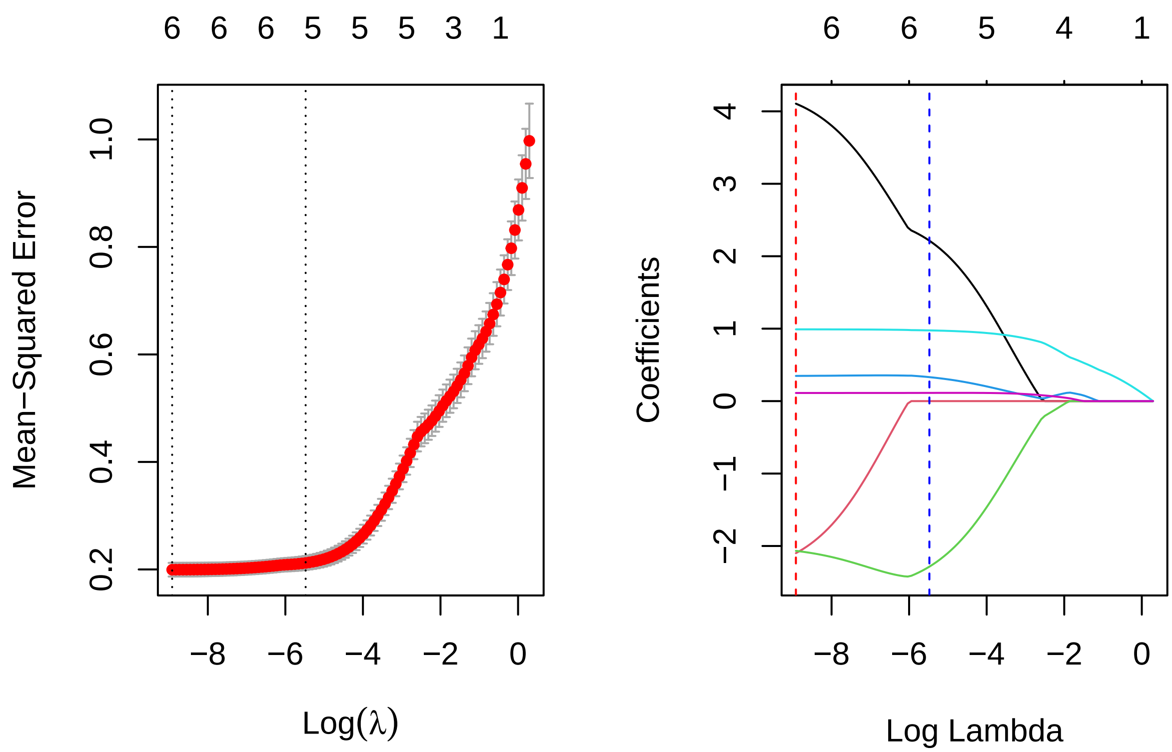

```
set.seed(311)
# elastic net
# alpha=0.25
```

```
cv.lambda.elastic1<-cv.glmnet(x=X,y=Y,alpha=0.25)

par(mfrow=c(1,2))

plot(cv.lambda.elastic1)


# get the optimal lambda
#log(cv.lambda.elastic1$lambda.min)

# elastic path 0.25

 plot(cv.lambda.elastic1$glmnet.fit,"lambda", label = FALSE)
 abline(v=log(cv.lambda.elastic1$lambda.min),col="red",lty="dashed")
 abline(v=log(cv.lambda.elastic1$lambda.1se),col="blue",lty="dashed")
```



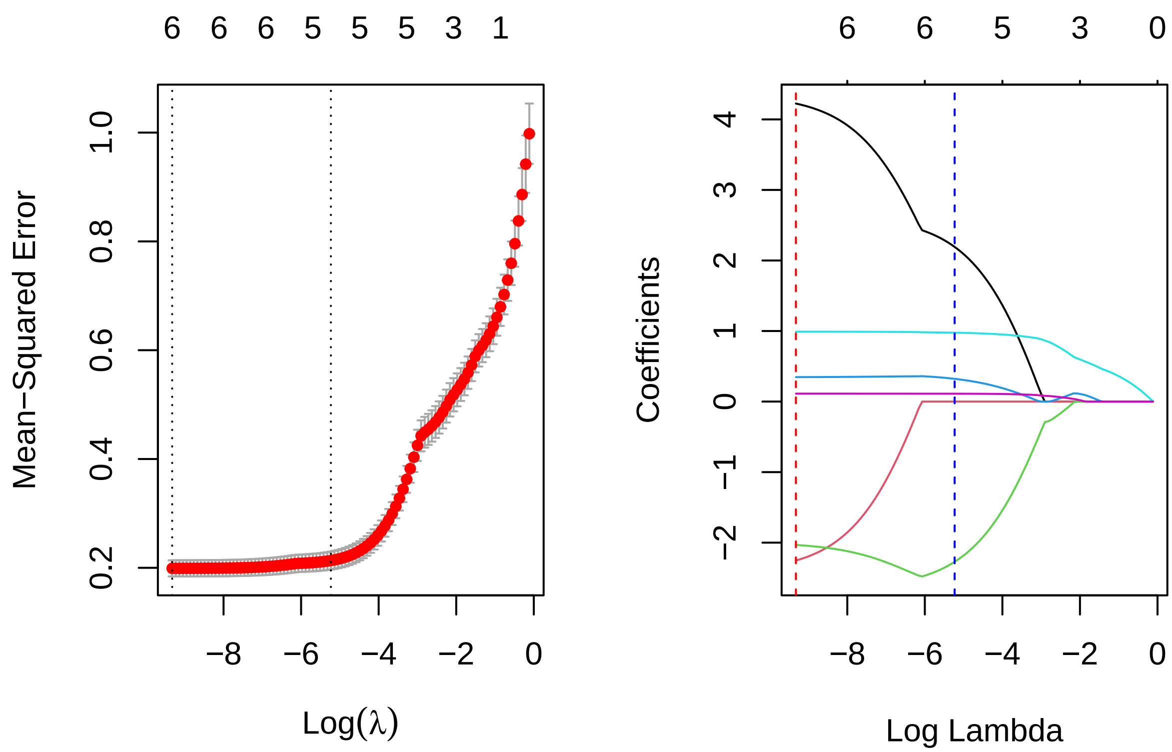

```
elasticmin1<-cv.lambda.elastic1$lambda.min
elastic1<-glmnet(x=X, y=Y,alpha=0.25, lambda = elasticmin1)
# elastic1$beta

#--------------------------------------------------

# alpha=0.5

par(mfrow=c(1,2))
```

```r
cv.lambda.elastic2<-cv.glmnet(x=X,y=Y,alpha=0.5)

plot(cv.lambda.elastic2)

# get the optimal lambda
#log(cv.lambda.elastic2$lambda.min)

# elastic path 0.5

plot(cv.lambda.elastic2$glmnet.fit,"lambda", label = FALSE)
abline(v=log(cv.lambda.elastic2$lambda.min),col="red",lty="dashed")
abline(v=log(cv.lambda.elastic2$lambda.1se),col="blue",lty="dashed")
```




```r
elasticmin2<-cv.lambda.elastic2$lambda.min
elastic2<-glmnet(x=X, y=Y,alpha=0.5, lambda = elasticmin2)
# elastic2$beta

# alpha=0.75


cv.lambda.elastic3<-cv.glmnet(x=X,y=Y,alpha=0.75)

par(mfrow=c(1,2))

plot(cv.lambda.elastic3)
```

```r
# get the optimal lambda
# log(cv.lambda.elastic3$lambda.min)

# elastic path 0.75

plot(cv.lambda.elastic3$glmnet.fit,"lambda", label = FALSE)
abline(v=log(cv.lambda.elastic3$lambda.min),col="red",lty="dashed")
abline(v=log(cv.lambda.elastic3$lambda.1se),col="blue",lty="dashed")
```




```r
elasticmin3<-cv.lambda.elastic3$lambda.min
elastic3<-glmnet(x=X, y=Y,alpha=0.75, lambda = elasticmin3)
#elastic3$beta



tabel.coef.elastic<-data.frame(Variable=c("Intercept","date","population","aggpce","psrate","mwunemp","
  Elastic0.25= round(as.vector(elastic1$beta),4),
            Elastic0.5= round(as.vector(elastic2$beta),4),
            Elastic0.75= round(as.vector(elastic3$beta),4))


lambda<-c("lambda",round(log(cv.lambda.elastic1$lambda.min),4),round(log(cv.lambda.elastic2$lambda.min)

kable(rbind(tabel.coef.elastic,lambda),full_width=FALSE)
```

Variable	Elastic0.25	Elastic0.5	Elastic0.75
Intercept	0	0	0
date	3.8116	4.1059	4.2248
population	-1.7203	-2.098	-2.2524
aggpce	-2.149	-2.067	-2.0323
psrate	0.3514	0.3485	0.3471
mwunemp	0.9898	0.9903	0.9904
inflation	0.1126	0.1124	0.1124
lambda	-8.2262	-8.9194	-9.3249

```
# coefficient of the models

tabel.coef<-round(cbind(OLS=coef(lm),
            Ridge=as.vector(ridge$beta),
            Lasso= as.vector(lasso$beta),
            Elastic0.25= as.vector(elastic1$beta),
            Elastic0.5= as.vector(elastic2$beta),
            Elastic0.75= as.vector(elastic3$beta)),6)

table.coef<-as.data.frame(tabel.coef)
table.coef[1,]<-round(c(lm$coefficients[1],ridge$a0,lasso$a0,elastic1$a0,elastic2$a0,elastic3$a0),6)

# print table

library(knitr)
kable(table.coef,full_width=FALSE)
```

	OLS	Ridge	Lasso	Elastic0.25	Elastic0.5	Elastic0.75
(Intercept)	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
date	4.496351	0.468637	4.270166	3.811588	4.105899	4.224833
population	-2.608364	0.115470	-2.311601	-1.720292	-2.098036	-2.252384
aggpce	-1.948970	-0.625030	-2.018694	-2.148954	-2.067016	-2.032295
psrate	0.344155	0.205976	0.346582	0.351389	0.348487	0.347146
mwunemp	0.990380	0.796479	0.990434	0.989803	0.990281	0.990423
inflation	0.112440	0.118167	0.112378	0.112573	0.112449	0.112413

```
# measure the performance of the models in test data set

dftest[,-1]<-scale(dftest[,-1])
dftest[,1]<-scale(dftest[,1])

# measure the lm model

lm.pred <- predict(lm,dftest,type="response")
lm.mse<-mean((lm.pred-dftest$unemrate)^2)

# measure the ridge model

X1<- model.matrix(unemrate~.,data=dftest)

Y1<-dftest[,"unemrate"]
```

```r
ridge.pred <- predict(ridge, s = ridgemin, newx = X1)

ridge.mse<-mean((ridge.pred - dftest$unemrate)^2)


# measure the lasso model

lasso.pred <- predict(lasso, s = lassomin, newx = X1)

lasso.mse<-mean((lasso.pred - dftest$unemrate)^2)

# measure the elastic models

elastic1.pred <- predict(elastic1, s = elasticmin1, newx = X1)

elastic1.mse<-mean((elastic1.pred - dftest$unemrate)^2)


elastic2.pred <- predict(elastic2, s = elasticmin2, newx = X1)

elastic2.mse<-mean((elastic2.pred - dftest$unemrate)^2)


elastic3.pred <- predict(elastic3, s = elasticmin3, newx = X1)

elastic3.mse<-mean((elastic3.pred - dftest$unemrate)^2)

# compare the mse

table.mse<-data.frame(Model=c("OLS","Ridge","Lasso","Elastic0.25","Elastic0.5","Elastic0.75"),
           MSE=c(lm.mse,ridge.mse,lasso.mse,elastic1.mse,elastic2.mse,elastic3.mse))

kable(table.mse,full_width=FALSE)
```

Model	MSE
OLS	0.2465653
Ridge	0.4055879
Lasso	0.2474696
Elastic0.25	0.2498947
Elastic0.5	0.2482531
Elastic0.75	0.2476791

```r
lm2<-lm(unemrate~.^2,data=dftrain)

kable(lm2$coefficients)
```

	x
(Intercept)	-1.3757154
date	8.4332085

	x
population	-2.7995356
aggpce	-6.0893606
psrate	0.3789069
mwunemp	1.2460114
inflation	0.0852220
date:population	4.0815250
date:aggpce	-20.2991788
date:psrate	-2.3262440
date:mwunemp	4.4787012
date:inflation	0.7920764
population:aggpce	17.7794282
population:psrate	3.4486080
population:mwunemp	-4.6330980
population:inflation	0.0767074
aggpce:psrate	-1.2305906
aggpce:mwunemp	-0.3796066
aggpce:inflation	-0.6193308
psrate:mwunemp	-0.0077138
psrate:inflation	0.0675634
mwunemp:inflation	-0.2179177

```
# measure the prediction

lm2.pred <- predict(lm2,dftest,type="response")
lm2.mse<-mean((lm2.pred-dftest$unemrate)^2)


lm_table<-data.frame(Measurement=c("Residual standard error","R-squared","Adjusted R-squared","MSE (tes
                     LM=c(summary(lm)$sigma,summary(lm)$r.squared,summary(lm)$adj.r.squared,lm.mse),
                     LM.Interaction.2=c(summary(lm2)$sigma,summary(lm2)$r.squared,summary(lm2)$adj.r.squa

kable(lm_table,full_width=FALSE)
```

Measurement	LM	LM.Interaction.2
Residual standard error	0.4416295	0.2467357
R-squared	0.8078962	0.9423256
Adjusted R-squared	0.8049634	0.9391215
MSE (test)	0.2465653	0.0839820