Analysis of World COVID-19 Vaccination Progress

Linpeng Sun, Priya Venkadesh

May 11, 2021

Abstract

This project is about finding the most significant factors that influenced the number of population vaccinated against the COVID-19 in various countries around the world. And demonstrate the vaccination progress by data visualization methods. Two datasets were used in this project, *country_profile_variables* and *country_vaccinations*. Through the analysis, Gibraltar, Seychelles, Israel have the fastest vaccination progress, with the achievement of 211.08, 129.88, and 121.03 vaccinations per hundred people.

1 Introduction

The COVID-19 pandemic has been a global challenge to everyone and a destruction to world's economy. The vaccination of COVID-19 vaccine plays a critical role in the maintenance of social and economic stability. The goals of this project are:

1. Tracking the progress of vaccination around the world by visualization techniques.
2. Identify the key factors affecting the vaccination process using supervised learning methods like ridge, least absolute shrinkage and selection operator(LASSO), Principal components regression(PCR), Partial Least Squares(PLS) and unsupervised method like best subset selection.

R is used as the only coding method for achieving goals of this project. This project involved with two datasets, *country_profile_variables* and *country_vaccinations*. Both datasets came from *Kaggle.com*, which is an online community of data scientists and machine learning practitioners. The *country_vaccinations* data is collected from Our World in Data GitHub repository for covid-19, merged and uploaded. It consisting of categorical and quantitative variables such as *date*, *Total number of vaccinations*, *Total vaccinations per hundred*, etc. The data is from 12/14/2020 to 05/04/2021. Dataset *country_profile_variables* is an internet based data service launched by the United Nations Statistics Division (UNSD) of the Department of Economic and Social Affairs (DESA). Which contains key statistical indicators of the countries. It covers 4 major sections: General Information, Economic Indicators, Social Indicators, Environmental & Infrastructure Indicators. The data is from 2017 when available or the most recent data previous to the year.

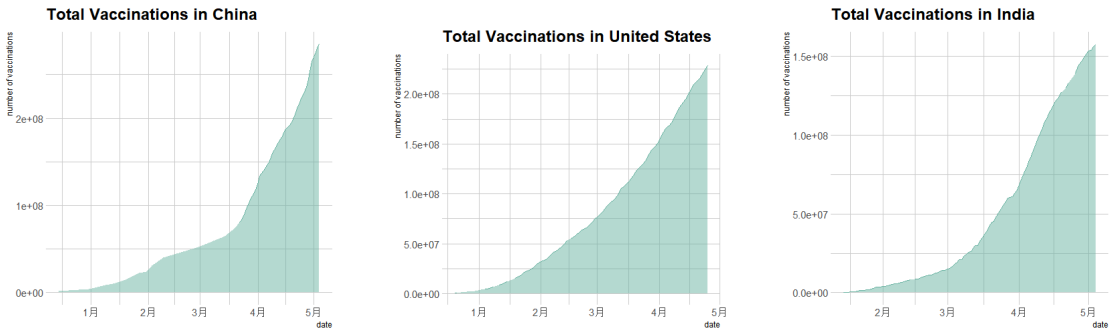
2 Materials and Methods

Before starting the analysis process, data cleaning and rearrangement is fundamental to the project. Due to the limitation of data collecting and distinction between each country's public health system. There are many *NA* values in both datasets. In the process of visualizing vaccination progress, *NA* values does not effect much. Yet, when analysing the key factors of vaccination difference between countries and applying statistical method to the datasets, *NA* values need to be eliminated. Since the datasets were collected individually, it is needed to merge them together. Combine the total vaccination per hundred people to corresponding country variables and get rid of the countries without vaccination information.

In this project, best subset selection, and ridge regression, LASSO, principle components regression, and partial least squares methods are used in this project. One advantage of the above methods is that the organized dataset has 37 predictors. Other methods can be used are cross-validation and bootstrap.

3 Result

3.1 Vaccination Progress Visualization



(a) (b) (c)

Figure 1: Total vaccinations

Table 1: Countries with highest total vaccination per hundred

Country	Gibraltar	Seychelles	Israel	United Arab Emirates	Cayman Islands
Region	SouthernEurope	EasternAfrica	WesternAsia	WesternAsia	Caribbean
Total Vaccination/100	211.08	129.88	121.03	108.99	99.96

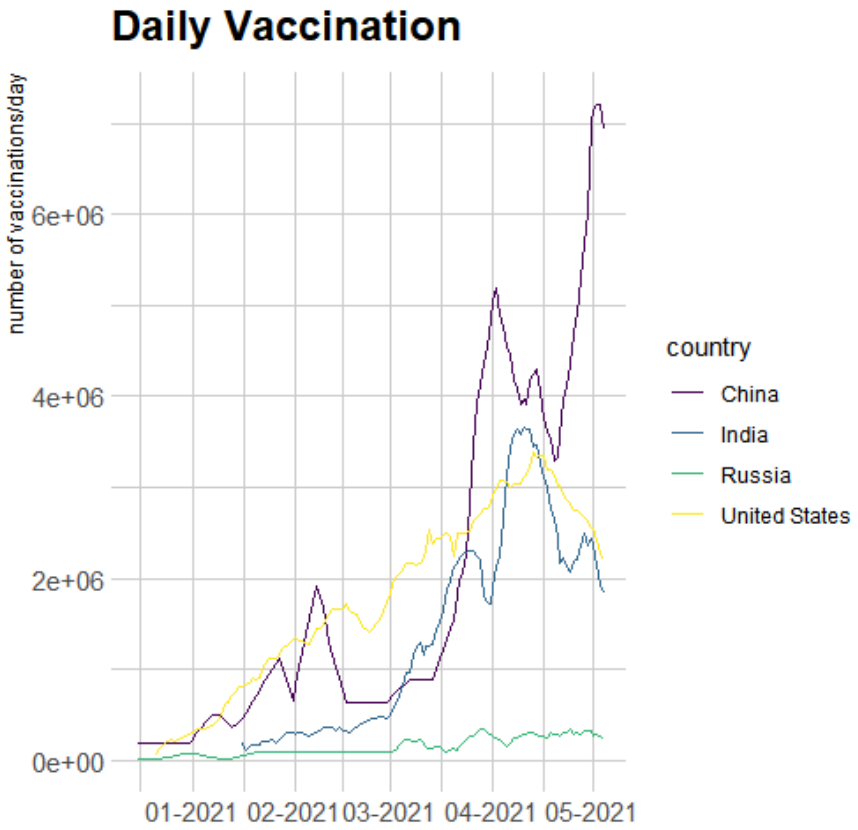


Figure 2: Daily vaccinations in US, China, India, and Russia.

3.2 Subset Selection

3.2.1 Best Subset Selection

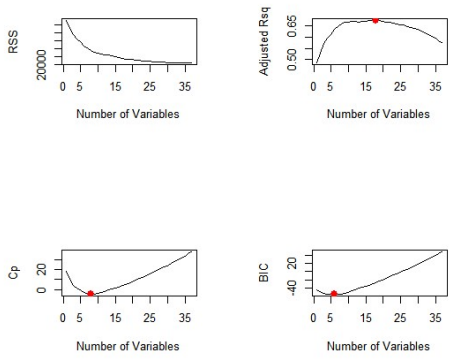


Figure 3: Plot of RSS, adjusted R^2, Cp, and BIC.

```
> coef(regfit.full, 6)
                                (Intercept)        Population.density..per.km2..2017.
                                -27.8033254                                 0.0254487
          Sex.ratio..m.per.100.f..2017.          Economy..Agriculture....of.GVA.
                                 0.4310897                                -0.7973808
    Employment..Industry....of.employed.      Employment..Services....of.employed.
                                -0.7190735                                 0.5024387
Population.growth.rate..average.annual...
                                -4.7753256
```

Figure 4: Coefficient estimates associated with 6-predictor model.

3.3 Shrinkage

3.3.1 Ridge Regression

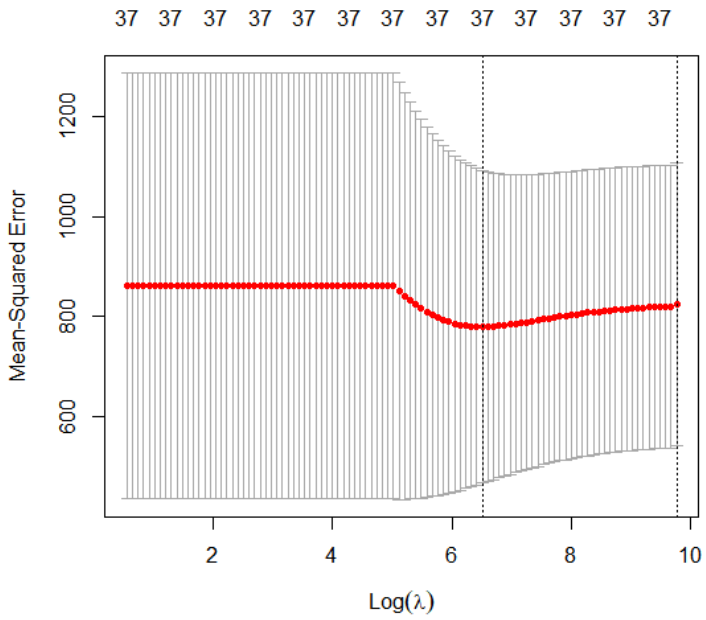


Figure 5: Ridge Regression: $log(\lambda)$ vs. MSE

```
> set.seed(1)
> cv.out=cv.glmnet(x[train ,],y[ train],alpha=0)
> plot(cv.out)
> bestlam =cv.out$lambda.min
> bestlam
[1] 672.4823
```

Figure 6: The value of λ that results in the smallest cross-validation error is 672.4823.

```
> # test MSE associated with λ = 672.4823
> ridge.pred=predict (ridge.mod ,s=bestlam ,newx=x[test ,])
> mean((ridge.pred -y.test)^2)
[1] 1245.566
> out=glmnet(x,y,alpha=0)
```

Figure 7: The test MSE associated with $\lambda = 672.4823$ is 1245.566.

3.3.2 The LASSO

```
> lasso.pred=predict(lasso.mod,s=bestlam2,newx=x[test,])
> mean((lasso.pred-y.test)^2)
[1] 1439.908
```

Figure 8: The test MSE of LASSO method is 1439.908.

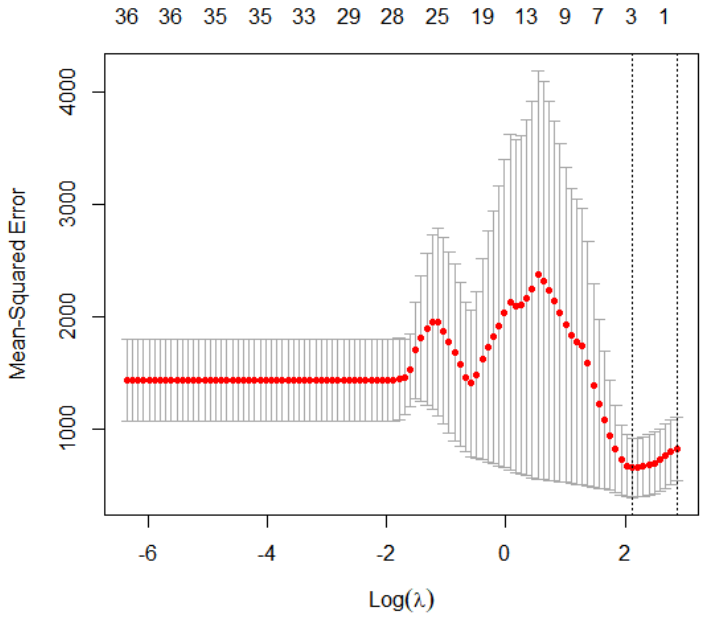


Figure 9: LASSO: $log(\lambda)$ vs. MSE

```
> lasso.coef[lasso.coef!=0]
(Intercept)
   29.41595
```

Figure 10: Null-zero coefficient.

3.3.3 Principal Components Regression

```
> pcr.pred=predict(pcr.fit,x[test,],ncomp=25)
> mean((pcr.pred-y.test)^2)
[1] 320.5759
> pcr.fit=pcr(y~x,scale=TRUE,ncomp=25)
> summary(pcr.fit)
Data:   X dimension: 79 37
        Y dimension: 79 1
Fit method: svdpc
Number of components considered: 25
TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X    27.24    43.10    55.94    63.89    69.67    74.27    78.27    81.50    84.27     86.52
y    46.17    46.63    49.09    55.07    56.16    57.85    58.26    58.49    60.19     60.38
   11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
X     88.36     90.09     91.59     92.95      94.1     94.99     95.76     96.47     97.05
y     61.86     61.93     62.23     64.08      64.7     64.79     64.79     65.58     66.85
   20 comps  21 comps  22 comps  23 comps  24 comps  25 comps
X     97.51     97.95     98.33     98.65     98.94     99.17
y     66.89     66.89     67.01     67.20     69.19     70.01
```

Figure 11: The the lowest cross-validation error occurs when $M = 25$ component are used, and the associated test MSE is 320.5459.

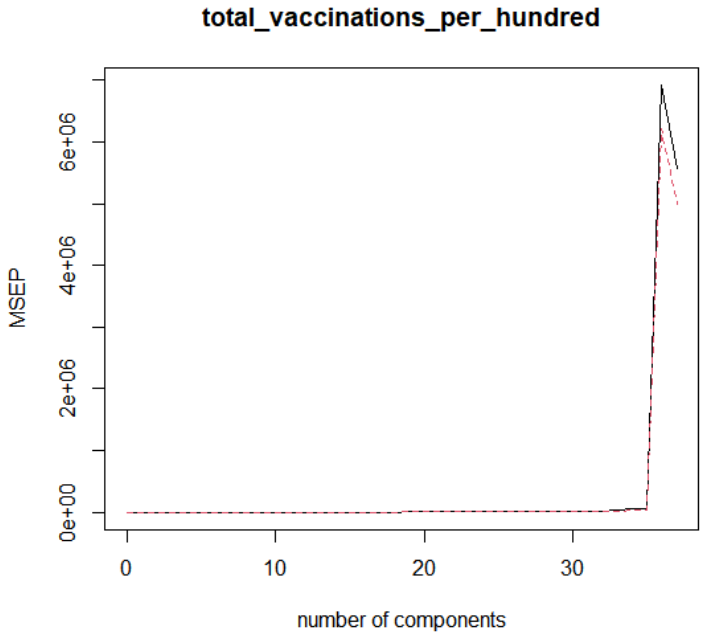


Figure 12: PCR: No. of component vs. MSE

3.3.4 Partial Least Squares

```
> pls.pred=predict(pls.fit,x[test,],ncomp=1)
> mean((pls.pred-y.test)^2)
[1] 698.4724
> pls.fit=plsr(total_vaccinations_per_hundred~.,data=CP_2,scale=TRUE,ncomp=1)
> summary(pls.fit)
Data:   X dimension: 79 37
        Y dimension: 79 1
Fit method: kernelpls
Number of components considered: 1
TRAINING: % variance explained
                                1 comps
X                                 26.76
total_vaccinations_per_hundred    53.61
```

Figure 13: The the lowest cross-validation error occurs when $M = 1$ component are used, and the associated test MSE is 698.4724.

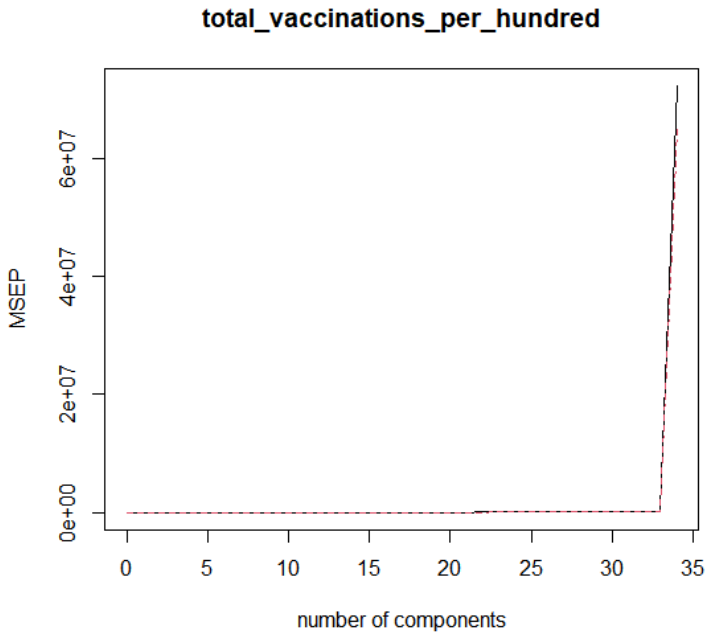


Figure 14: PLS: No. of component vs. MSE

4 Conclusions

From the result we can see, the principal components regression has the smallest MSE value. Which is 320.5459 when 25 components are used. It is good for deriving a low-dimensional set of features from a large set of variables.

Since $p = 37$, best subset selection method becomes computationally infeasible. Consequently, it should not be the best model for analysing the dataset in this project. However, the lasso has a substantial advantage over ridge regression in that the resulting coefficient

```
> coef(pcr.fit, 25)
, , 25 comps

                                                                y
Surface.area..km2.                                     -2.63222789
Population.in.thousands..2017.                         -2.75397284
Population.density..per.km2..2017.                      9.81185487
Sex.ratio..m.per.100.f..2017.                           8.90261291
GDP..Gross.domestic.product..million.current.US..      -0.55033752
GDP.growth.rate..annual....const..2005.prices.          5.32799658
GDP.per.capita..current.US..                            0.23855785
Economy..Agriculture....of.GVA.                        -6.54561903
Economy..Industry....of.GVA.                           -6.59062813
Economy..Services.and.other.activity....of.GVA.        -1.37127249
Employment..Agriculture....of.employed.                -2.41917345
Employment..Industry....of.employed.                    3.19081079
Employment..Services....of.employed.                    6.94027894
Unemployment....of.labour.force.                        0.26716672
Agricultural.production.index..2004.2006.100.          -2.68520822
Food.production.index..2004.2006.100.                  -3.01084803
International.trade..Exports..million.US..              1.58251796
International.trade..Imports..million.US..              1.91325988
International.trade..Balance..million.US..              0.05208554
Balance.of.payments..current.account..million.US..     -5.83272376
Population.growth.rate..average.annual...              -8.39006756
Urban.population....of.total.population.               -2.40536501
Urban.population.growth.rate..average.annual...         8.24911555
Fertility.rate..total..live.births.per.woman.          -5.90370121
Refugees.and.others.of.concern.to.UNHCR..in.thousands.  0.47069422
Infant.mortality.rate..per.1000.live.births            -1.69323525
Health..Total.expenditure....of.GDP.                   -2.75652912
Health..Physicians..per.1000.pop..                     -2.69268773
Education..Government.expenditure....of.GDP.            2.82913628
Seats.held.by.women.in.national.parliaments..           0.47095134
Mobile.cellular.subscriptions..per.100.inhabitants.     1.25317665
Mobile.cellular.subscriptions..per.100.inhabitants..1   0.87223393
Individuals.using.the.Internet..per.100.inhabitants.   -3.67848830
Threatened.species..number.                             3.15538350
CO2.emission.estimates..million.tons.tons.per.capita.   6.01621201
Energy.production..primary..Petajoules.                -0.58025196
Pop..using.improved.sanitation.facilities..urban.rural.... -3.25954507
Mobile.cellular.subscriptions..per.100.inhabitants      1.25317665
```

Figure 15: Best PCR model

estimates are sparse. Here we see that all of the 37 coefficient estimates are exactly zero. So the lasso model with λ chosen by cross-validation contains no variables.

Gibraltar, Seychelles, Israel are the top-3 countries with the highest vaccination per hundred people. China, United States, and India are the top-3 countries with the highest number of total vaccination of 270406000, 243463471, and 153626325 by 2021-05-01. More information on public health care system would help us to establish a greater degree of accuracy on this matter. Further research should be undertaken to explore how country-of-origin of vaccine affected the vaccination progress.

References

[1] COVID-19 World Vaccination Progress, `https://www.kaggle.com/gpreda/covid-world-vaccination-progress`.

[2] Country Statistics - UNData, `https://www.kaggle.com/sudalairajkumar/undata-country-profiles`.

[3] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics) (1st ed. 2013, Corr. 7th printing 2017 ed.). Springer. `https://doi.org/10.1007/978-1-4614-7138-7`.

5 Appendix

```
library(tidyverse)
library(hrbrthemes)
library(plotly)
library(patchwork)
library(babynames)
library(viridis)
CV <- read.csv("country_vaccinations.csv")
head(CV)
CP <- read.csv("country_profile_variables.csv")
CP <- subset(CP, CP$country %in% unique(CV$country))
CV <- CV[!is.na(CV$total_vaccinations_per_hundred),]
CV <- CV[ order(CV$date , decreasing = TRUE ),]
CP$total_vaccinations_per_hundred <-
CV$total_vaccinations_per_hundred[match(CP$country,C
V$country)]
CP <- na.omit(CP)
drop <- c("Region",
"Labour.force.participation..female.male.pop....",
"Life.expectancy.at.birth..females.males..years.",
"Population.age.distribution..0.14...60..years....",
"International.migrant.stock..000...of.total.pop..",
"Education..Primary.gross.enrol..ratio..f.m.per.100.pop..",
"Education..Secondary.gross.enrol..ratio..f.m.per.100.pop..
", "Education..Tertiary.gross.enrol..ratio..f.m.per.100.pop..",
"Forested.area....of.land.area.",
"Energy.supply.per.capita..Gigajoules.",
"Pop..using.improved.drinking.water..urban.rural....",
"Net.Official.Development.Assist..received....of.GNI.")
CP_2 <- CP[,!(names(CP) %in% drop)]
CP_2 <- as.data.frame(lapply(CP_2[,-1], as.numeric))
CP_2 <- CP_2[complete.cases(CP_2), ]
x = model.matrix(total_vaccinations_per_hundred~., data
= CP_2)[,-1]
y = CP_2$total_vaccinations_per_hundred
## Ridge Regression
library(glmnet)
grid = 10^seq(10,-2,length = 100)
ridge.mod = glmnet(x,y, alpha=0,lambda=grid)
# access the ridge regression coefficients
coef(ridge.mod)
# 38 x 100 matrix: each row for one predictor, each column
for one lambda value
dim(coef(ridge.mod))
ridge.mod$lambda[50]
ridge.mod$lambda[60]
coef(ridge.mod)[,50]
sqrt(sum(coef(ridge.mod)[-1,50]^2))
sqrt(sum(coef(ridge.mod)[-1,60]^2))
set.seed(1)
train=sample (1: nrow(x), nrow(x)/2)
test=(-train)
y.test=y[test]
ridge.mod=glmnet(x[train ,],y[ train],alpha=0, lambda
=grid ,
                 thresh =1e-12)
ridge.pred=predict (ridge.mod ,s=4, newx=x[test ,])
# test MSE
mean((ridge.pred -y.test)^2)
# with a very large lambda
ridge.pred=predict (ridge.mod ,s=1e10 ,newx=x[test ,])
mean((ridge.pred -y.test)^2)
set.seed(1)
cv.out=cv.glmnet(x[train ,],y[ train],alpha=0)
plot(cv.out)
bestlam =cv.out$lambda.min
bestlam
# the value of λ that results in the smallest
cross validation error is 672.4823
# test MSE associated with λ = 672.4823
ridge.pred=predict (ridge.mod ,s=bestlam ,newx=x[test ,])
mean((ridge.pred -y.test)^2)
out=glmnet(x,y,alpha=0)
predict(out,type="coefficients",s=bestlam) [1:20,]

## LASSO
lasso.mod=glmnet(x[train ,],y[ train],alpha=1, lambda
=grid)
plot(lasso.mod)
set.seed(1)
cv.out=cv.glmnet(x[train ,],y[ train],alpha=1)
plot(cv.out)
bestlam2=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam2,newx=x[test,])
mean((lasso.pred-y.test)^2)
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(out,type="coefficients",s=bestlam)[1:37,
]
lasso.coef
lasso.coef[lasso.coef!=0]

## Principal Components Regression
library (pls)
set.seed(2)
pcr.fit=pcr(total_vaccinations_per_hundred~.,data=CP_2,s
cale=TRUE,validation ="CV")
summary (pcr.fit)
validationplot(pcr.fit,val.type="MSEP")
set.seed(1)
```

```
pcr.fit=pcr(total_vaccinations_per_hundred~.,data=CP_2,subset=train,scale=TRUE,validation="CV")
validationplot(pcr.fit,val.type="MSEP")
pcr.pred=predict(pcr.fit,x[test,],ncomp=25)
mean((pcr.pred-y.test)^2)
pcr.fit=pcr(y~x,scale=TRUE,ncomp=25)
summary(pcr.fit)

## Partial Least Squares
set.seed(1)
pls.fit=plsr(total_vaccinations_per_hundred~.,data=CP_2,subset=train,scale=TRUE,validation="CV")
summary(pls.fit)
validationplot(pls.fit,val.type="MSEP")
pls.pred=predict(pls.fit,x[test,],ncomp=1)
mean((pls.pred-y.test)^2)
pls.fit=plsr(total_vaccinations_per_hundred~.,data=CP_2,scale=TRUE,ncomp=1)
summary(pls.fit)

## Best Subset Selection
library(leaps)
regfit.full = regsubsets(total_vaccinations_per_hundred~., CP_2, nvmax=37)
reg.summary = summary(regfit.full)
reg.summary$rsq
#plot adjusted R2 and select the best model
# type=”l” connects the plotted points with lines
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="RSS",
     type="l")
# identify the location of a maximum point of a vector
which.max(reg.summary$adjr2)
#plot RSS, adjusted R2, Cp, BIC for all of the models in one picture
par(mfrow=c(2,2))
#plot RSS
plot(reg.summary$rss, xlab="Number of Variables", ylab="RSS", type="l")
#plot adjusted R2
plot(reg.summary$adjr2, xlab="Number of Variables", ylab="Adjusted Rsq",
     type="l")
which.max(reg.summary$adjr2)
points(18, reg.summary$adjr2[18], col="red", cex=2, pch=20)
#plot Cp
plot(reg.summary$cp, xlab="Number of Variables", ylab="Cp", type="l")
which.min(reg.summary$cp)
points(8, reg.summary$cp[8], col="red", cex=2, pch=20)
#plot bic
plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC", type="l")
which.min(reg.summary$bic)
points(6, reg.summary$bic[6], col="red", cex=2, pch=20)
plot(regfit.full, scale="r2")
plot(regfit.full, scale="adjr2")
plot(regfit.full, scae="Cp")
plot(regfit.full, scale="bic")
coef(regfit.full, 18)
```