```
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)
```