peru_glm_corona.r

Roy Costilla and Freddy A. Rojas Cama

17 de mayo 2020

```
##################################################
# Scripts de R para reproducir los analisis de
# Costilla y Rojas (2020)
# Predicción de corto plazo del número de fallecimientos por COVID19 en Perú:
# enfoque usando modelos lineales generalizados
##################################################


# Setting up
rm(list=ls())
set.seed(12345678)
require(MASS)
```

```
## Loading required package: MASS
```

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages ----------------------------------------------------------------- tidyverse 1.3
```

```
## v ggplot2 3.2.1     v purrr   0.3.4
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts -------------------------------------------------------------------- tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```
require(readxl)
```

```
## Loading required package: readxl
```

```
require(ggplot2)
require(glm.predict)
```

```
## Loading required package: glm.predict
```

```
## Loading required package: parallel
```

```
# Reading data directly from repository
reportes_minsa.xlsx =tempfile()
download.file("https://github.com/jincio/COVID_19_PERU/blob/master/docs/reportes_minsa.xlsx?raw=true",
alldata=read_excel("reportes_minsa.xlsx", sheet = "Sheet2")
str(alldata)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    72 obs. of  18 variables:
##  $ Dia                      : POSIXct, format: "2020-03-06" "2020-03-07" ...
##  $ Hora                     : POSIXct, format: "1899-12-31 18:00:00" "1899-12-31 18:00:00" ...
##  $ Total_Pruebas            : num  155 219 250 318 346 ...
##  $ Descartados              : num  154 213 243 309 335 ...
##  $ Positivos                : num  1 6 7 9 11 17 22 38 43 71 ...
##  $ Nuevos_Positivos         : num  1 5 1 2 2 6 5 16 5 28 ...
##  $ TasaPositivos            : num  0.645 2.74 2.8 2.83 3.179 ...
##  $ Pruebas_dia              : num  155 64 31 68 28 368 141 377 313 277 ...
##  $ Recuperados              : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Fallecidos               : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Hospitalizados           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Hospitalizados_UCI       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Hospitalizados_ventilador: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PruebasRapidas           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ RapidasPositivos         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Pruebas_diaPR            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PR_nuevos                : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ PM_PR                    : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
alldata$Dia[nrow(alldata)]
```

```
## [1] "2020-05-16 UTC"
```

```
dataperu=data.frame(alldata[,
              c("Dia","Fallecidos","Hospitalizados","Pruebas_dia","TasaPositivos")])
colnames(dataperu)=c("date","ycum","hosp","pruebasd","tpositivos")

# Plotting Parameters
myvar="y"
# Uncomment line below for models with moving averages
#myvar="y_3dma"
maxpred=ifelse(myvar=="y",160,125)

# subsetting data from first deaths (19 March)
dataperu=dataperu[!is.na(dataperu$ycum),]
dataperu$y=dataperu$ycum-lag(dataperu$ycum, 1)
dataperu$y[1]=dataperu$ycum[1]
# until 16 May
dataperu <- dataperu %>% filter(date<"2020-05-17")

# set prediction date (7 days in future)
npred=dim(dataperu)[1]+7

# 3d ma
dataperu$y_3dma=round((dataperu$y+lag(dataperu$y,1)+lag(dataperu$y,2))/3,0)

# temporal variables
dataperu$t=1:nrow(dataperu)
dataperu$t2=(1:nrow(dataperu))^2
dataperu$t3=(1:nrow(dataperu))^3
dataperu$t4=(1:nrow(dataperu))^4

# weekdays effect for Poisson
dataperu$date=as.Date(dataperu$date)
```

```
dataperu$weekdays=as.factor(weekdays(dataperu$date, abbr=T))
dataperu$dia = recode_factor(dataperu$weekdays,
                             Fri = "Viernes",
                             Sat = "Sabado",
                             Sun = "Domingo",
                             Mon = "Lunes",
                             Tue = "Martes",
                             Wed = "Miercoles",
                             Thu = "Jueves",
                             .default = levels(dataperu$weekdays))

#####   Poisson
mymodel=glm(get(myvar) ~ 1+t+t2+dia,
            data=dataperu, family=poisson)
summary(mymodel)
```

```
##
## Call:
## glm(formula = get(myvar) ~ 1 + t + t2 + dia, family = poisson,
##     data = dataperu)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8420  -1.0376  -0.3997   0.6552   4.7255
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0970276  0.1903982  -0.510 0.610329
## t             0.1628887  0.0097334  16.735  < 2e-16 ***
## t2           -0.0013847  0.0001224 -11.317  < 2e-16 ***
## diaSabado     0.0690781  0.0678478   1.018 0.308614
## diaDomingo   -0.2637065  0.0798973  -3.301 0.000965 ***
## diaLunes     -0.2934699  0.0793769  -3.697 0.000218 ***
## diaMartes    -0.0606894  0.0731908  -0.829 0.406995
## diaMiercoles  0.0461303  0.0702948   0.656 0.511669
## diaJueves    -0.0190572  0.0706298  -0.270 0.787300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2263.02  on 58  degrees of freedom
## Residual deviance:  153.62  on 50  degrees of freedom
## AIC: 450.21
##
## Number of Fisher Scoring iterations: 5
```

```
# out-of-sample prediction
myt=seq(1,npred, 1)
newdates=as.Date(seq(dataperu$date[1], dataperu$date[1]+npred-1, by=1))
newweekend=ifelse(weekdays(newdates, abbr=T)%in%c("Sun","Mon"),1,0)
newweekdays=as.factor(weekdays(newdates, abbr=T))
newdata=data.frame(1,t=myt, t2=myt^2, t3=myt^3, t4=myt^4,
                   weekend=newweekend, weekdays=newweekdays)
```

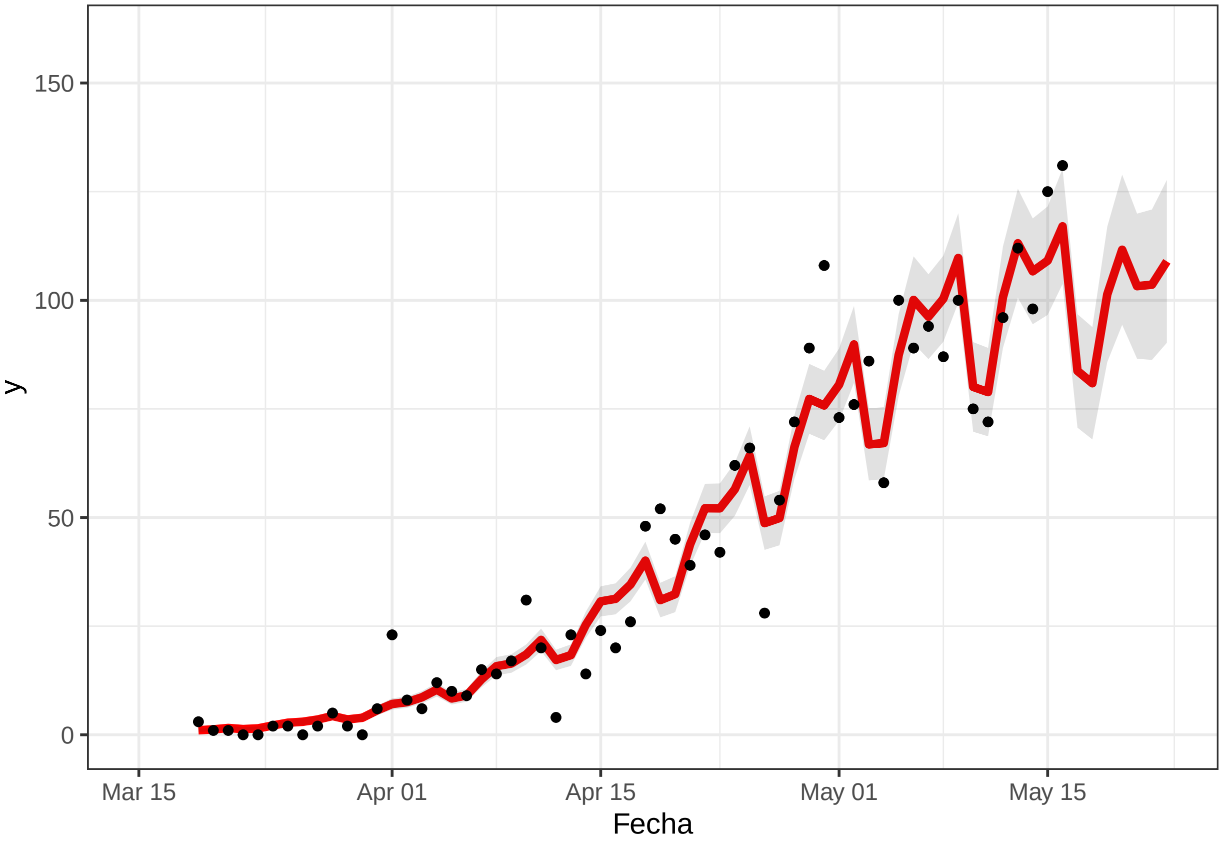
```r
newdata$dia = recode_factor(newdata$weekdays,
                            Fri = "Viernes",
                            Sat = "Sabado",
                            Sun = "Domingo",
                            Mon = "Lunes",
                            Tue = "Martes",
                            Wed = "Miercoles",
                            Thu = "Jueves",
                            .default = levels(newdata$weekdays))
newdata$date=dataperu$date[1]+(newdata$t-1)

# Prediction
mypred=predict.glm(mymodel,type="response", se.fit = T,
                   newdata = newdata)
newdata$mypred=mypred$fit
newdata$mypred.min=ifelse(mypred$fit-1.96*mypred$se.fit>0,mypred$fit-1.96*mypred$se.fit,0)
newdata$mypred.max=mypred$fit+1.96*mypred$se.fit

# max point
yhatmax=which(mypred$fit==max(mypred$fit[!is.na(mypred$fit)]))
as.Date(dataperu$date[1])+as.numeric(yhatmax)
```

```
## [1] "2020-05-17"
```

```r
# plot out-of-sample
plt <- ggplot(newdata,aes(x = date, y = mypred)) +
  geom_line(color="red", size=1.5)+
  geom_ribbon(aes(ymin = mypred.min, ymax = mypred.max),
              alpha = 0.15, size=2)+
  labs(x = 'Fecha', y = myvar)+
  ylim(0,maxpred) + xlim(as.Date(c("2020-03-15",max(newdata$date))))+
  theme_bw()+
  geom_point(data=dataperu, aes(x=date, y=get(myvar)), size=1.25)
plt
```



```r
pdf(paste0("poisson_peru_",
           myvar,'_',paste(attr(mymodel$terms,"term.labels"), collapse = '_'),
           "_n",nrow(dataperu),"_",npred-nrow(dataperu),"days.pdf"),
    height = 3, width = 4.5)
plt
dev.off()
```

```
## pdf
##   2
```

```r
# GOF competing models
mycovars=c("1","t","t2","t3","t4")
mygof=matrix(NA,nrow=length(mycovars),ncol=4, dimnames=list(mycovars, c("npar","DF","AIC","BIC")))
thiscovars=NULL
for (mycovar in mycovars){
  if (mycovar=="1") thiscovars=paste(mycovar,"dia",sep="+") else
    thiscovars=paste(thiscovars, mycovar, sep = "+")
  myformula=as.formula(paste(myvar, " ~ ", thiscovars))
  message("Model: ",myformula, '\n')
  mymodel=glm(myformula,
                   data=dataperu,family=poisson)
  npar=nrow(dataperu)-mymodel$df.residual
  mygof[mycovar,]=c(npar,mymodel$df.residual,AIC(mymodel),BIC(mymodel))
  rownames(mygof)[which(rownames(mygof)==mycovar)]=thiscovars
}
```

```
## Model: ~y1 + dia
```

```
## Model: ~y1 + dia + t
## Model: ~y1 + dia + t + t2
```

```
## Model: ~y1 + dia + t + t2 + t3
```

```
## Model: ~y1 + dia + t + t2 + t3 + t4
```

```
round(mygof,1)
```

```
##                   npar DF    AIC    BIC
## 1+dia                7 52 2487.0 2501.5
## 1+dia+t              8 51  599.0  615.6
## 1+dia+t+t2           9 50  450.2  468.9
## 1+dia+t+t2+t3       10 49  451.5  472.2
## 1+dia+t+t2+t3+t4    11 48  452.8  475.6
```

```
write.csv(mygof,paste0("poi_gof_",myvar,"_",thiscovars,".csv"), quote=F)


#############  NB
mymodel.nb=glm.nb(get(myvar)~1+t+t2+dia,
                data=dataperu)
rbind(summary(mymodel.nb)$coef,
      theta=c(summary(mymodel.nb)$theta,summary(mymodel.nb)$SE.theta,NA,NA))
```

```
##                   Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept)  -0.157718133 0.249210785 -0.6328704 5.268183e-01
## t             0.166954608 0.013853574 12.0513744 1.907427e-33
## t2           -0.001439941 0.000189951 -7.5805910 3.439843e-14
## diaSabado     0.079329368 0.133814471  0.5928310 5.532943e-01
## diaDomingo   -0.302082749 0.145356537 -2.0782192 3.768917e-02
## diaLunes     -0.292941747 0.143878457 -2.0360362 4.174672e-02
## diaMartes    -0.088037880 0.139756266 -0.6299387 5.287347e-01
## diaMiercoles  0.101763576 0.136455177  0.7457656 4.558090e-01
## diaJueves    -0.045479339 0.136866553 -0.3322896 7.396706e-01
## theta        23.811136880 9.711506228         NA           NA
```

```
write.csv(summary(mymodel.nb)$coef,paste0("nb_coef_",myvar,".csv"), quote=F)


# out-of-sample prediction
myt=seq(1,npred, 1)
newdates=as.Date(seq(dataperu$date[1], dataperu$date[1]+npred-1, by=1))
newweekend=ifelse(weekdays(newdates, abbr=T)%in%c("Sun","Mon"),1,0)
newweekdays=as.factor(weekdays(newdates, abbr=T))
newdata=data.frame(1,t=myt, t2=myt^2, t3=myt^3, t4=myt^4,
                   weekend=newweekend, weekdays=newweekdays)
newdata$dia = recode_factor(newdata$weekdays,
                            Fri = "Viernes",
                            Sat = "Sabado",
                            Sun = "Domingo",
                            Mon = "Lunes",
                            Tue = "Martes",
                            Wed = "Miercoles",
                            Thu = "Jueves",
                            .default = levels(newdata$weekdays))

newdata$date=dataperu$date[1]+(newdata$t-1)
```

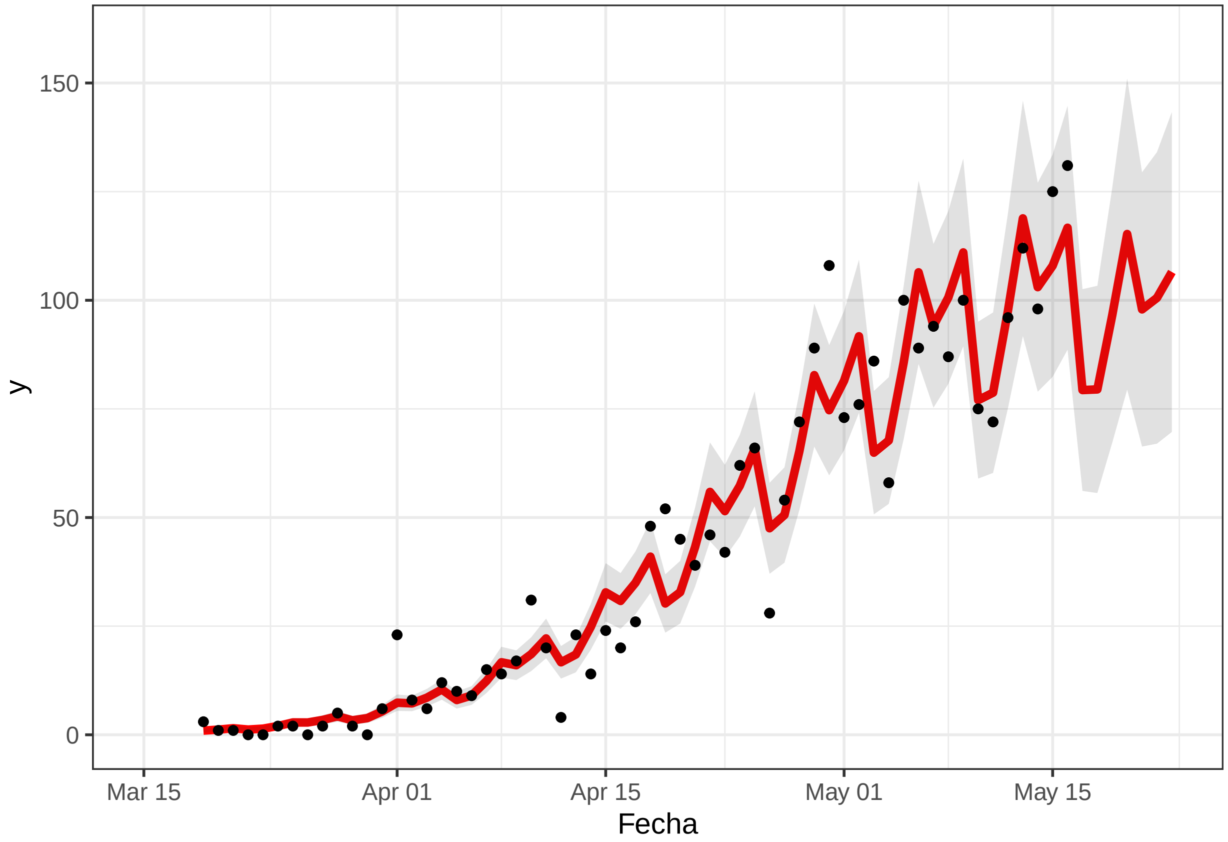
```
# Prediction
mypred=predict.glm(mymodel.nb,type="response", se.fit = T,
                   newdata = newdata)
# prediction
newdata$mypred=mypred$fit
newdata$mypred.min=ifelse(mypred$fit-1.96*mypred$se.fit>0,mypred$fit-1.96*mypred$se.fit,0)
newdata$mypred.max=mypred$fit+1.96*mypred$se.fit
# max point
yhatmax=which(mypred$fit==max(mypred$fit[!is.na(mypred$fit)]))
newdata[yhatmax,]
```

```
##    X1  t   t2     t3      t4 weekend weekdays       dia       date   mypred
## 56  1 56 3136 175616 9834496       0      Wed Miercoles 2020-05-13 118.8485
##    mypred.min mypred.max
## 56   91.75062   145.9464
```

```
as.Date(dataperu$date[1])+as.numeric(yhatmax)
```

```
## [1] "2020-05-14"
```

```
plt <- ggplot(newdata,aes(x = date, y = mypred)) +
  geom_line(color="red", size=1.5)+
  geom_ribbon(aes(ymin = mypred.min, ymax = mypred.max),
              alpha = 0.15)+
  labs(x = 'Fecha', y = myvar)+
  ylim(0,maxpred) + xlim(as.Date(c("2020-03-15",max(newdata$date))))+
  theme_bw()+
  geom_point(data=dataperu, aes(x=date, y=get(myvar)), size=1.25)
plt
```



```
pdf(paste0("nb_peru_",myvar,'_',
           paste(attr(mymodel.nb$terms,"term.labels"), collapse = '_'),
           "_n",nrow(dataperu),"_",npred-nrow(dataperu),"days.pdf"),
    height = 3, width = 4.5)
plt
dev.off()
```

```
## pdf
##   2
```

```
# GOF NB
mycovars=c("1","t","t2","t3","t4")
mygof.nb=matrix(NA,nrow=length(mycovars),ncol=4, dimnames=list(mycovars, c("npar","DF","AIC","BIC")))
thiscovars=NULL
for (mycovar in mycovars){
#       if (myvar=="1") mycovars=myvar else
      if (mycovar=="1") thiscovars=paste(mycovar,"dia",sep="+") else
      thiscovars=paste(thiscovars, mycovar, sep = "+")
  myformula=as.formula(paste(myvar, " ~ ", thiscovars))
  message("Model: ",myformula, '\n')
  mymodel.nb=glm.nb(myformula,
                    data=dataperu)
  theta=c(summary(mymodel.nb)$theta, summary(mymodel.nb)$SE.theta, NA,NA)
  npar=nrow(dataperu)-mymodel.nb$df.residual+1
  mygof.nb[mycovar,]=c(npar,mymodel.nb$df.residual,AIC(mymodel.nb),BIC(mymodel.nb))
  rownames(mygof.nb)[which(rownames(mygof.nb)==mycovar)]=thiscovars
}
```

```
## Model: ~y1 + dia
```

```
## Model: ~y1 + dia + t
```

```
## Model: ~y1 + dia + t + t2
```

```
## Model: ~y1 + dia + t + t2 + t3
```

```
## Model: ~y1 + dia + t + t2 + t3 + t4
```

```
write.csv(mygof.nb,paste0("nb_gof_",myvar,"_",thiscovars,".csv"), quote=F)

cat("GOF Poisson \n")
```

```
## GOF Poisson
```

```
round(mygof,1)
```

```
##                    npar DF    AIC    BIC
## 1+dia                 7 52 2487.0 2501.5
## 1+dia+t               8 51  599.0  615.6
## 1+dia+t+t2            9 50  450.2  468.9
## 1+dia+t+t2+t3        10 49  451.5  472.2
## 1+dia+t+t2+t3+t4     11 48  452.8  475.6
```

```
cat("GOF Negative Binomial \n")
```

```
## GOF Negative Binomial
```

```
round(mygof.nb,1)
```

```
##                    npar DF   AIC   BIC
## 1+dia                 8 52 573.6 590.2
## 1+dia+t               9 51 469.1 487.8
## 1+dia+t+t2           10 50 428.8 449.5
## 1+dia+t+t2+t3        11 49 430.5 453.3
## 1+dia+t+t2+t3+t4     12 48 432.5 457.4
```