

¿Quiénes protestan? - Trabajo Maestría

Luis Alberto Chávez

Preparando la data

```
library(tinytex)
```

```
rm(list = ls()) # limpiar el working environment
```

```
linkADrive='https://docs.google.com/spreadsheets/d/e/2PACX-1vQoN_1tfvAClKC06wjHLy7rGNdbZx2nuvTm99d2Fzrw'
```

```
protestas=read.csv(linkADrive)
```

```
head(protestas)
```

##	UBIGEO	NOM_PROV	NOM_DIST	CONFLICTOS	PP_Nac_Loc	PP_Nom	POBLACION	ALTITUD
## 1	80201	ACOMAYO	ACOMAYO	0	1	SP	4523	3235
## 2	80202	ACOMAYO	ACOPIA	0	2	MRIP	3121	3724
## 3	80203	ACOMAYO	ACOS	0	1	SP	2328	3106
## 4	80204	ACOMAYO	MOSOC LLACTA	0	1	SP	983	3820
## 5	80205	ACOMAYO	POMACANCHI	0	2	APP	7515	3709
## 6	80206	ACOMAYO	RONDOCAN	1	1	SP	1800	3394

##	TRANS_TOTAL	TRANS_TOTAL_MILL	RENTA_PER	PORC_EJEC	DIAS_ATRASO
## 1	5077807	5.08	467.52	NA	NA
## 2	3469203	3.47	462.37	NA	NA
## 3	2474700	2.47	442.53	NA	NA
## 4	2418484	2.42	1026.44	NA	NA
## 5	8357551	8.36	463.01	79.2	870
## 6	1686511	1.69	390.64	NA	NA

```
str(protestas)
```

```
## 'data.frame': 116 obs. of 13 variables:
## $ UBIGEO : int 80201 80202 80203 80204 80205 80206 80207 80302 80301 80303 ...
## $ NOM_PROV : chr "ACOMAYO" "ACOMAYO" "ACOMAYO" "ACOMAYO" ...
## $ NOM_DIST : chr "ACOMAYO" "ACOPIA" "ACOS" "MOSOC LLACTA" ...
## $ CONFLICTOS : int 0 0 0 0 0 1 0 0 14 1 ...
## $ PP_Nac_Loc : int 1 2 1 1 2 1 2 2 2 2 ...
## $ PP_Nom : chr "SP" "MRIP" "SP" "SP" ...
## $ POBLACION : int 4523 3121 2328 983 7515 1800 3362 7900 26916 2854 ...
## $ ALTITUD : int 3235 3724 3106 3820 3709 3394 3788 3479 2800 3424 ...
## $ TRANS_TOTAL : int 5077807 3469203 2474700 2418484 8357551 1686511 4783121 10110480 24131530 ...
## $ TRANS_TOTAL_MILL: num 5.08 3.47 2.47 2.42 8.36 ...
```

```
## $ RENTA_PER      : num  468 462 443 1026 463 ...
## $ PORC_EJEC      : num   NA NA NA NA 79.2 NA NA 56.2 NA NA ...
## $ DIAS_ATRASO     : int   NA NA NA NA 870 NA NA 841 NA NA ...
```

Le damos formato a nuestras variables

```
protestas$PP_Nac_Loc = as.factor(protestas$PP_Nac_Loc) #lo convertimos a una variable categórica ya que
protestas$PP_Nom = as.factor(protestas$PP_Nom) #también lo convertimos a una variable categórica
```

Exploración Univariada de nuestra variable dependiente: Conflictos (contando a Cusco)

```
summary(protestas$CONFLICTOS)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00   0.00    1.00    7.44   5.00   294.00
```

Vemos los estadígrafos:

```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.4.2
```

```
allStats=c(summary(protestas$CONFLICTOS),
  sd=sd(protestas$CONFLICTOS), # variabilidad (en relacion a la media)
  skew=Skew(protestas$CONFLICTOS), # asimetria (positiva o negativa)
  kurt=Kurt(protestas$CONFLICTOS), # concentración (enpinada / aplanada)
  cv=CoefVar(protestas$CONFLICTOS)) # variabilidad (mayor o menor que uno)
allStats
```

```
##      Min.    1st Qu.    Median      Mean   3rd Qu.     Max.      sd
##      0.000000  0.000000  1.000000   7.439655  5.000000  294.000000  28.709728
##      skew      kurt      cv
##      8.698648  82.793186  3.859013
```

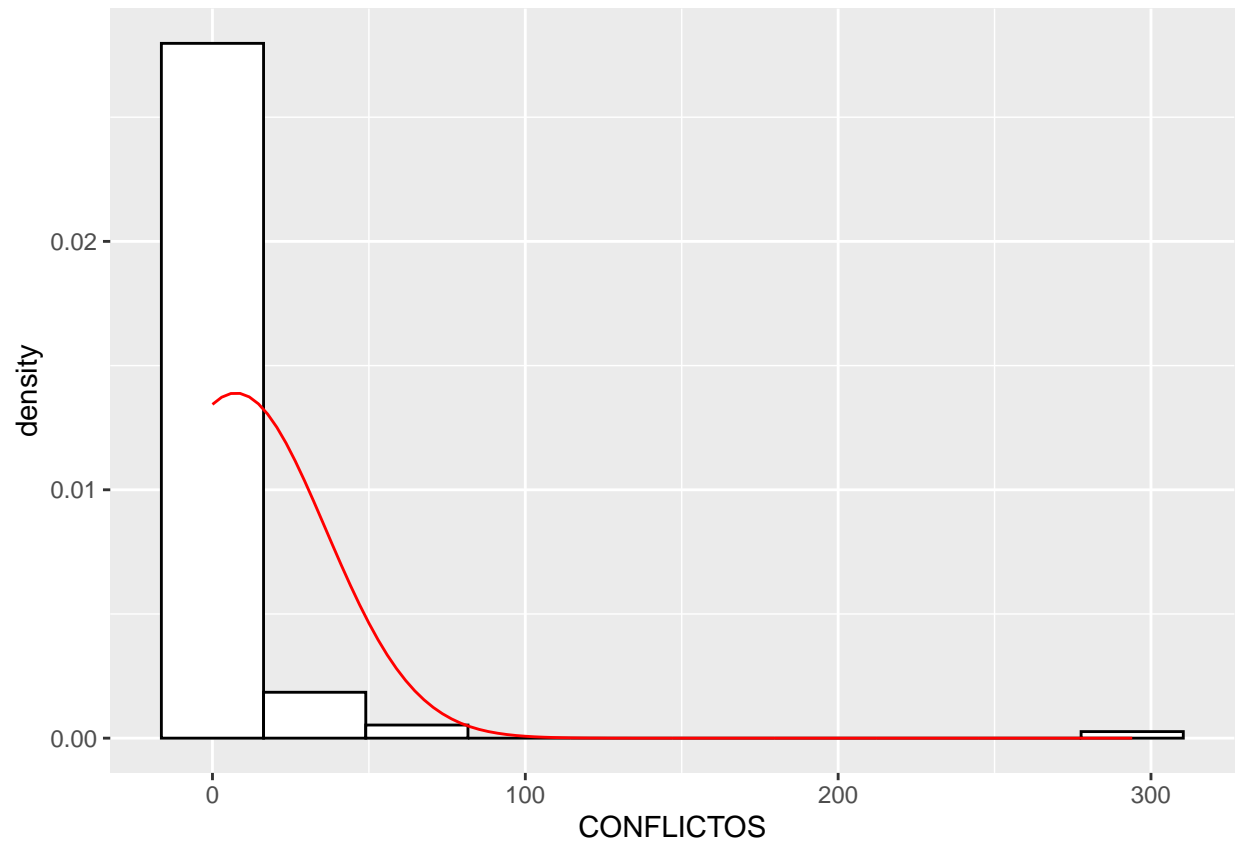
Empleando un histograma:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

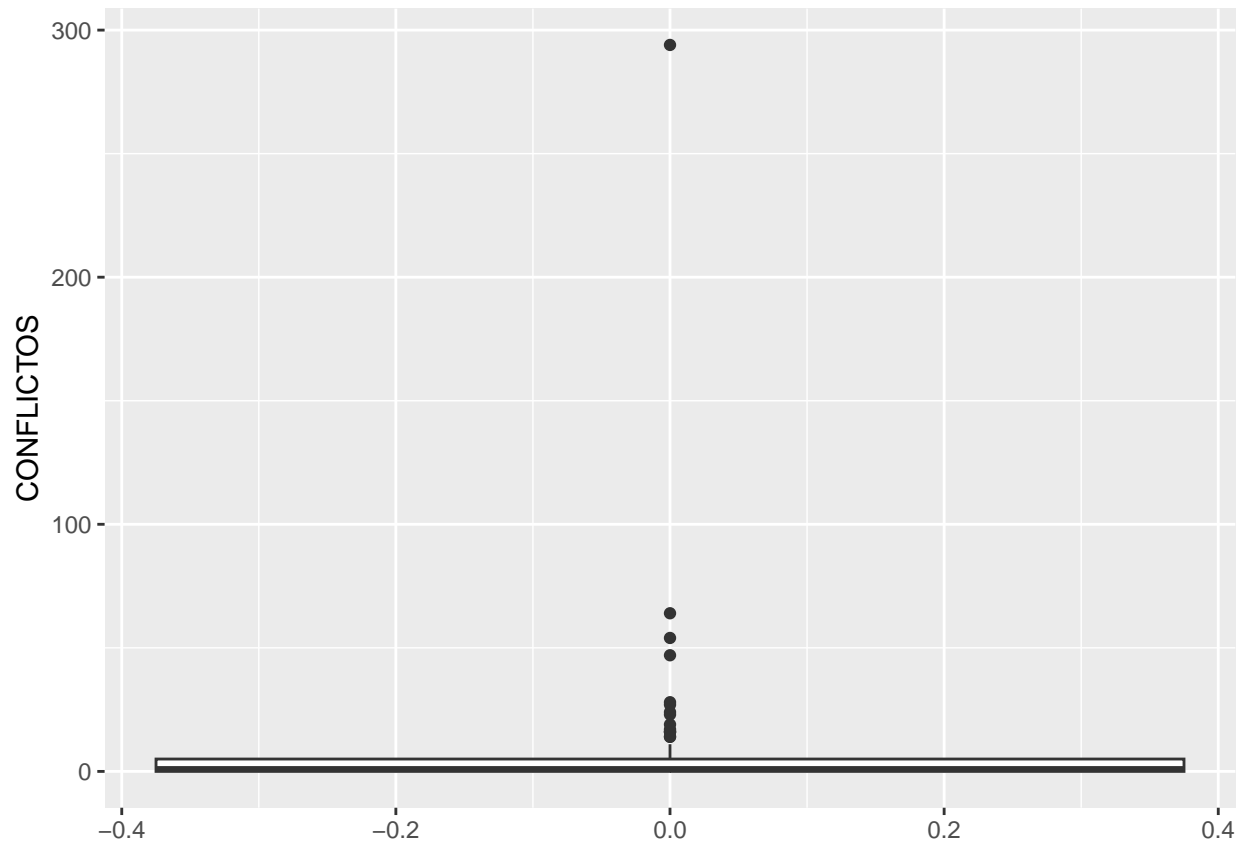
```
base=ggplot(data=protestas,
  aes(x=CONFLICTOS))
histogram= base + geom_histogram(aes(y = after_stat(density)),
  colour = 1, fill = "white",bins=10) +
  stat_function(fun = dnorm,
    args = list(mean = allStats['Mean'],
      sd = allStats['sd']),col='red')

histogram
```



Y en un boxplot:

```
base=ggplot(data=protestas,  
            aes(y=CONFLICTOS))  
boxplot=base + geom_boxplot()  
  
boxplot
```



```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
```

```
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
protestas %>%
  arrange(desc(CONFLICTOS)) %>%
  slice_head(n = 5)
```

```
##   UBIGEO      NOM_PROV  NOM_DIST CONFLICTOS PP_Nac_Loc PP_Nom POBLACION
## 1  80101      CUSCO     CUSCO      294        2    MRIP    119148
## 2  80801      ESPINAR   ESPINAR      64        2     AA     40781
## 3  80601      CANCHIS   SICUANI      54        2    MRIP    65864
## 4  80901 LA CONVENCION SANTA ANA      47        1     APP    29456
```

```
## 5  80702  CHUMBIVILCAS CAPACMARCA      28      1      SP      4064
##    ALTITUD TRANS_TOTAL TRANS_TOTAL_MILL RENTA_PER PORC_EJEC DIAS_ATRASO
## 1    3439    47619888          47.62    166.30      NA      NA
## 2    3976    27802717          27.80    283.34      NA      NA
## 3    3593    43903355          43.90    277.32      NA      NA
## 4    1086    69821925          69.82    987.15      NA      NA
## 5    3563     4617367           4.62    473.05      NA      NA
```

Podemos observar como Cusco concentra un mayor número de conflictos. Esto se debe a las limitaciones en la creación de la base de datos. Es por ello que, por motivos metodológicos, vamos a obviar al distrito de Cusco y el resto de filas que cuenten con valores vacíos para nuestros cálculos.

Borramos los NAs, y nos percatamos que nos quedan 81 observaciones/distritos. Nuestro trabajo aún sigue en pie!

```
protestas <- protestas[complete.cases(protestas), ]
```

Exploración Univariada de nuestra variable dependiente: Conflictos (sin Cusco, ni NA's)

Aquí podemos ver el valor mínimo, máximo, los cuartiles.

```
summary(protestas$CONFLICTOS)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000   0.000   1.000   3.074   4.000  27.000
```

Pero nos faltan algunos estadígrafos:

```
library(DescTools)
allStats=c(summary(protestas$CONFLICTOS),
  sd=sd(protestas$CONFLICTOS), # variabilidad (en relacion a la media)
  skew=Skew(protestas$CONFLICTOS), # asimetria (positiva o negativa)
  kurt=Kurt(protestas$CONFLICTOS), # concentración (enpinada / aplanada)
  cv=CoefVar(protestas$CONFLICTOS)) # variabilidad (mayor o menor que uno)
allStats
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.      sd      skew
## 0.000000 0.000000 1.000000 3.074074 4.000000 27.000000 5.212432 2.536538
##      kurt      cv
## 6.924679 1.695610
```

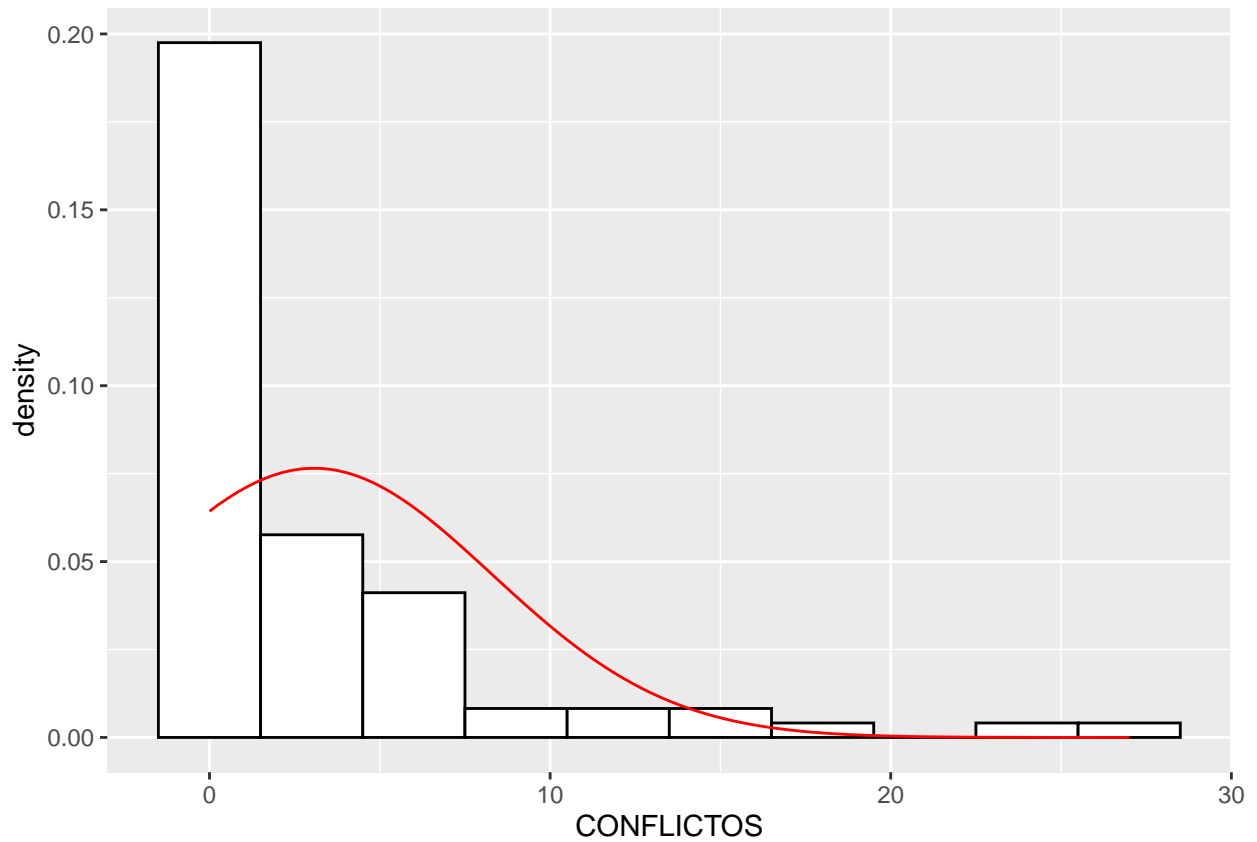
Podemos verlo en el siguiente histograma:

```
library(ggplot2)

base=ggplot(data=protestas,
  aes(x=CONFLICTOS))
histogram= base + geom_histogram(aes(y = after_stat(density)),
  colour = 1, fill = "white",bins=10) +
```

```
stat_function(fun = dnorm,
              args = list(mean = allStats['Mean'],
                          sd = allStats['sd']),col='red')
```

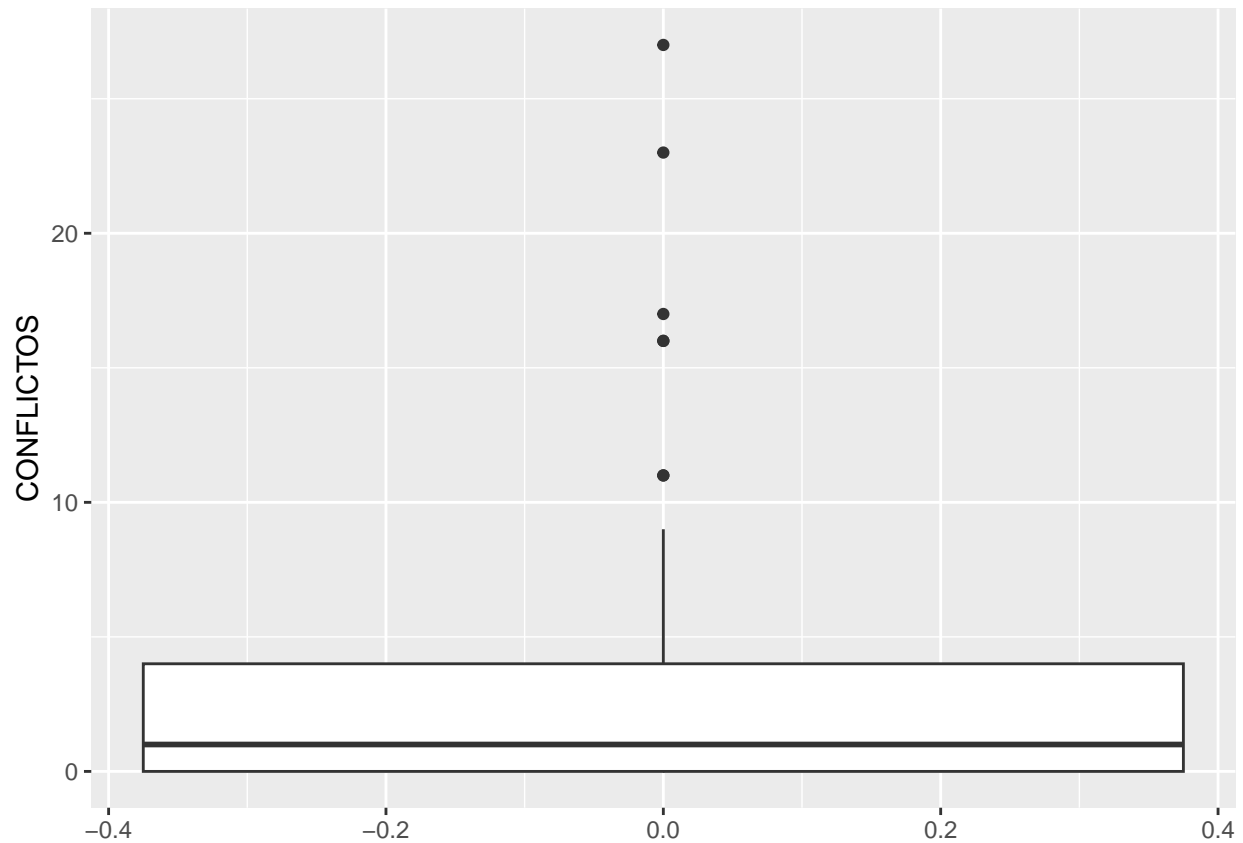
histogram



Podemos verlo en un boxplot:

```
base=ggplot(data=protestas,
            aes(y=CONFLICTOS))
boxplot=base + geom_boxplot()
```

boxplot



Observamos como la data mejora, así que vamos por un buen camino.

Podemos identificar a los atípicos:

```
valuesFromBox=ggplot_build(boxplot)$data[[1]]
valuesFromBox
```

```
##   ymin lower middle upper ymax      outliers notchupper notchlower
## 1    0    0      1     4    9 16, 27, 23, 17, 11, 11, 16  1.702222 0.2977778
##   x flipped_aes PANEL group ymin_final ymax_final  xmin  xmax xid newx
## 1 0      FALSE     1    -1          0        27 -0.375 0.375   1    0
##   new_width weight colour fill alpha shape linetype linewidth
## 1      0.75     1 grey20 white  NA    19    solid      0.5
```

```
outliersLocales=valuesFromBox$outliers[[1]]
outliersLocales
```

```
## [1] 16 27 23 17 11 11 16
```

Podemos calcular los outliers usando los máximos y mínimos teóricos:

```
valuesFromBox[c('ymin','ymax')]
```

```
##   ymin ymax
## 1    0    9
```

Entonces, los valores que excedan 9 serán considerados atípicos: Tabla Atípicos

```
protestas[protestas$CONFLICTOS>9,]
```

##	UBIGEO	NOM_PROV	NOM_DIST	CONFLICTOS	PP_Nac_Loc	PP_Nom	POBLACION
## 43	80704	CHUMBIVILCAS	COLQUEMARCA	16	1	PL	6923
## 48	80708	CHUMBIVILCAS	VELILLE	27	1	PL	8587
## 56	80108	CUSCO	WANCHAQ	23	1	PFE	61530
## 59	80803	ESPINAR	COPORAQUE	17	1	SP	8688
## 76	80907	LA CONVENCION	QUIMBIRI	11	2	MRIP	14056
## 96	81106	PAUCARTAMBO	KOSÑIPATA	11	1	SP	4924
## 114	81306	URUBAMBA	OLLANTAYTAMBO	16	1	SP	12061
##	ALTITUD	TRANS_TOTAL	TRANS_TOTAL_MILL	RENTA_PER	PORC_EJEC	DIAS_ATRASO	
## 43	3606	9432339	9.43	567.48	85.2		783
## 48	3766	9960410	9.96	482.60	91.0		973
## 56	3424	11035252	11.04	74.64	55.5		689
## 59	3600	8205436	8.21	394.71	70.1		1067
## 76	590	129322852	129.32	3824.54	79.5		712
## 96	544	16973835	16.97	1434.45	58.3		715
## 114	2871	18055156	18.06	622.75	83.3		1019

Podemos decir que las protestas se distribuyen por lo general entre 0 y 9 entre todos los distritos del departamento del Cusco. Asimismo, a través de la mediana, notamos que la mitad de los distritos sostiene 1 conflicto en el año 2023. Por otro lado, los distritos destacados son 7.

Podemos agrupar las protestas por provincias:

```
library(dplyr)
protestas_provincias <- protestas %>%
  group_by(NOM_PROV) %>%
  summarise(total_conflictos = sum(CONFLICTOS, na.rm = TRUE))

protestas_provincias
```

```
## # A tibble: 13 x 2
##   NOM_PROV      total_conflictos
##   <chr>          <int>
## 1 ACOMAYO             0
## 2 ANTA                1
## 3 CALCA              4
## 4 CANAS              0
## 5 CANCHIS           15
## 6 CHUMBIVILCAS       70
## 7 CUSCO             46
## 8 ESPINAR           27
## 9 LA CONVENCION      38
## 10 PARURO             1
## 11 PAUCARTAMBO        11
## 12 QUISPICANCHI      13
## 13 URUBAMBA          23
```


Regresión Poisson

En nuestro caso, lo ideal resulta ser correr una regresión Poisson, ya que nuestra variable dependiente se trata de un conteo, el conteo de los conflictos a nivel distrital.

Nuestra hipótesis destaca que el número de conflictos en los distritos del Cusco, se encuentran afectados por el tipo de partido (ya sea nacionalista o local), la transferencia total por canon gasífero en millones, la renta per cápita, la altitud, el porcentaje de ejecución presupuestal y el número de días de atraso en obras públicas.

```
library("modelsummary")

## 'modelsummary' 2.0.0 now uses 'tinytable' as its default table-drawing
## backend. Learn more at: https://vincentarelbundock.github.io/tinytable/
##
## Revert to 'kableExtra' for one session:
##
## options(modelsummary_factory_default = 'kableExtra')
## options(modelsummary_factory_latex = 'kableExtra')
## options(modelsummary_factory_html = 'kableExtra')
##
## Silence this message forever:
##
## config_modelsummary(startup_message = FALSE)

##
## Adjuntando el paquete: 'modelsummary'

## The following objects are masked from 'package:DescTools':
##
## Format, Mean, Median, N, SD, Var

hipo = formula(CONFLICTOS ~ PP_Nac_Loc + TRANS_TOTAL_MILL + RENTA_PER + ALTITUD + PORC_EJEC + DIAS_ATRASO)

rp = glm(hipo, data = protestas,
        offset = log(POBLACION), # variable de control: población
        family = poisson(link = "log"))

modelo = list('Poisson asegurado (I)' = rp)

modelsummary(modelo,
             title = "Regresión Poisson",
             stars = TRUE,
             output = "kableExtra")
```

Si bien esta primera tabla nos da un vistazo inicial, es necesario exponenciar los valores.

```
formatoNum <- function(x) format(x, digits = 4, scientific = FALSE)

modelsummary(modelo,
             fmt=formatoNum, # uso mi formula
             exponentiate = T, # exponenciar!!!!)
```

Table 1: Regresión Poisson

	Poisson asegurado (I)
(Intercept)	-7.668*** (0.502)
PP_Nac_Loc2	-0.934*** (0.188)
TRANS_TOTAL_MILL	-0.019*** (0.003)
RENTA_PER	0.000*** (0.000)
ALTITUD	0.000 (0.000)
PORC_EJEC	-0.006 (0.005)
DIAS_ATRASO	0.001* (0.000)
Num.Obs.	81
AIC	578.2
BIC	595.0
Log.Lik.	-282.108
F	16.994
RMSE	4.72
+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}	

```

statistic = 'conf.int',
title = "Regresión Poisson - coeficientes exponenciados",
stars = TRUE,
output = "kableExtra")

```

Equidispersión

Sin embargo, uno de los supuestos de la regresión Poisson señala que tanto la media y la varianza deben ser iguales. Para conocer si nuestra regresión cumple esa condición corremos la siguiente prueba:

```

library(magrittr)
library(kableExtra)

```

```
## Warning: package 'kableExtra' was built under R version 4.4.3
```

```
##
```

```
## Adjuntando el paquete: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## group_rows
```

Table 2: Regresión Poisson - coeficientes exponenciados

	Poisson asegurado (I)
(Intercept)	0.000 467 5*** [0.000 171 6, 0.001 228]
PP_Nac_Loc2	0.392 885 6*** [0.268 376 8, 0.561 627]
TRANS_TOTAL_MILL	0.980 856 5*** [0.974 663 1, 0.986 808]
RENTA_PER	1.000 364 8*** [1.000 251 5, 1.000 483]
ALTITUD	0.999 961 4 [0.999 812 5, 1.000 119]
PORC_EJEC	0.994 496 9 [0.985 381 4, 1.003 962]
DIAS_ATRASO	1.000 658 1* [1.000 017 5, 1.001 289]
Num.Obs.	81
AIC	578.2
BIC	595.0
Log.Lik.	-282.108
F	16.994
RMSE	4.72

+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}

```

overdispersion=AER::dispersiontest(rp,alternative='greater')$ p.value<0.05
underdispersion=AER::dispersiontest(rp,alternative='less')$ p.value<0.05
# tabla
testResult=as.data.frame(rbind(overdispersion,underdispersion))
names(testResult)='Es probable?'
testResult%>%kable(caption = "Test de Equidispersión")%>%kableExtra::kable_styling()

```

Table 3: Test de Equidispersión

	Es probable?
overdispersion	TRUE
underdispersion	FALSE

Hemos hallado la presencia de sobredispersión, por lo que debemos correr una regresión Quasi Poisson y una Binomial Negativa

Quasi Poisson

Útil para la sobredispersión y subdispersión.

```

rqp = glm(hipo, data = protestas,
          offset = log(POBLACION),
          family = quasipoisson(link="log"))

```

Table 4: Regresión QuasiPoisson

	(1)
(Intercept)	-7.668*** (1.480)
PP_Nac_Loc2	-0.934+ (0.554)
TRANS_TOTAL_MILL	-0.019* (0.009)
RENTA_PER	0.000* (0.000)
ALTITUD	0.000 (0.000)
PORC_EJEC	-0.006 (0.014)
DIAS_ATRASO	0.001 (0.001)
Num.Obs.	81
F	1.954
RMSE	4.72
+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}	

```
modelsummary(rqp,
  title = "Regresión QuasiPoisson",
  stars = TRUE,
  output = "kableExtra")
```

Binomial Negativa

Útil solo para la sobredispersión

```
library(MASS)
```

```
##
```

```
## Adjuntando el paquete: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## select
```

```
hipo_bn = formula(CONFLICTOS ~ PP_Nac_Loc + TRANS_TOTAL_MILL + RENTA_PER + ALTITUD + PORC_EJEC + DIAS_A
```

```
rbn = glm.nb(hipo_bn, data = protestas)
```

```
modelsummary(rbn,
  title = "Regresión BinomialNegativa",
  stars = TRUE,
  output = "kableExtra")
```

Table 5: Regresión BinomialNegativa

	(1)
(Intercept)	-7.721*** (1.378)
PP_Nac_Loc2	-0.664 (0.441)
TRANS_TOTAL_MILL	-0.008 (0.008)
RENTA_PER	0.000 (0.000)
ALTITUD	0.000 (0.000)
PORC_EJEC	-0.002 (0.013)
DIAS_ATRASO	0.000 (0.001)
Num.Obs.	81
AIC	347.3
BIC	366.5
Log.Lik.	-165.656
F	0.696
RMSE	5.80
+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}	

Vemos nuestras regresiones

```

todos_modelos = list('Poisson'= rp,
                     'Quasi Poisson'= rqp,
                     'Binomial Negativa'= rbn)

modelsummary(todos_modelos, fmt=formatoNum,
             exponentiate = T,
             statistic = 'conf.int',
             title = "EXP de las Regresiones",
             stars = TRUE,
             output = "kableExtra")

```

Comparando modelos

```

# poisson case
performance::check_overdispersion(rp)

## # Overdispersion test
##
##      dispersion ratio =    8.698

```

Table 6: EXP de las Regresiones

	Poisson	Quasi Poisson	Binomial Negativa
(Intercept)	0.000 467 5*** [0.000 171 6, 0.001 228]	0.000 467 5*** [0.000 021 61, 0.007 372]	0.000 443 4*** [0.000 031 96, 0.007 732]
PP_Nac_Loc2	0.392 885 6*** [0.268 376 8, 0.561 627]	0.392 885 6+ [0.116 847 71, 1.068 867]	0.514 537 5 [0.206 640 15, 1.329 637]
TRANS_TOTAL_MILL	0.980 856 5*** [0.974 663 1, 0.986 808]	0.980 856 5* [0.961 969 87, 0.997 736]	0.992 106 8 [0.975 640 08, 1.013 426]
RENTA_PER	1.000 364 8*** [1.000 251 5, 1.000 483]	1.000 364 8* [1.000 040 32, 1.000 731]	1.000 160 6 [0.999 755 63, 1.000 569]
ALTITUD	0.999 961 4 [0.999 812 5, 1.000 119]	0.999 961 4 [0.999 541 48, 1.000 454]	0.999 991 8 [0.999 584 66, 1.000 358]
PORC_EJEC	0.994 496 9 [0.985 381 4, 1.003 962]	0.994 496 9 [0.968 304 50, 1.023 753]	0.997 763 2 [0.968 810 51, 1.026 263]
DIAS_ATRASO	1.000 658 1* [1.000 017 5, 1.001 289]	1.000 658 1 [0.998 743 60, 1.002 492]	1.000 316 6 [0.998 776 02, 1.001 894]
Num.Obs.	81	81	81
AIC	578.2		347.3
BIC	595.0		366.5
Log.Lik.	-282.108		-165.656
F	16.994	1.954	0.696
RMSE	4.72	4.72	5.80

+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}

```
## Pearson's Chi-Squared = 643.645
## p-value = < 0.001
```

```
## Overdispersion detected.
```

```
# quasipoisson case
performance::check_overdispersion(rqp)
```

```
## # Overdispersion test
##
## dispersion ratio = 8.698
## Pearson's Chi-Squared = 643.645
## p-value = < 0.001
```

```
## Overdispersion detected.
```

```
# negative binomial case
library(DHARMA)
```

```
## Warning: package 'DHARMA' was built under R version 4.4.2
```

```
## This is DHARMA 0.4.7. For overview type '?DHARMA'. For recent changes, type news(package = 'DHARMA')
```

```
performance::check_overdispersion(rbn)
```

```
## # Overdispersion test
##
## dispersion ratio = 0.376
##           p-value = 0.464

## No overdispersion detected.
```

Por el momento vemos que la mejor regresión resulta ser la Binomial Negativa, ya que elimina la sobredispersión.

Asimismo, como no son modelos anidados, usamos la tabla anova con cuidado, pidiendo un test chi-cuadrado, para comparar los modelos:

```
anova(rp, rqp, rbn, test = "Chisq") %>%
kable(caption = "Tabla ANOVA para comparar modelos") %>% kableExtra::kable_styling(full_width = FALSE)
```

Table 7: Tabla ANOVA para comparar modelos

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
74	424.10534	NA	NA	NA
74	424.10534	0	0.0000	NA
74	79.48912	0	344.6162	NA

Aquí nos fijamos en el Resid.Dev, si decae de una regresión a otra suele ser mejor la última opción.

Podemos emplear otra prueba:

```
lmtest::lrtest(rp, rbn) %>%
kable(caption = "loglikelihood ratio test") %>% kableExtra::kable_styling(full_width = FALSE)
```

Table 8: loglikelihood ratio test

#Df	LogLik	Df	Chisq	Pr(>Chisq)
7	-282.1084	NA	NA	NA
8	-165.6562	1	232.9045	0

Aquí nos fijamos en el LogLik, si el número se acerca más al 0 de una regresión a otra, suele ser mejor.

Nuestra tabla de interés

Por todos los resultados, podemos quedarnos con la regresión Binomial Negativa para comprender los resultados:

Table 9: Regresión Binomial Negativa - coeficientes exponenciados

Binomial Negativa asegurado (I)	
(Intercept)	0.000 443 4*** [0.000 031 96, 0.007 732]
PP_Nac_Loc2	0.514 537 5 [0.206 640 15, 1.329 637]
TRANS_TOTAL_MILL	0.992 106 8 [0.975 640 08, 1.013 426]
RENTA_PER	1.000 160 6 [0.999 755 63, 1.000 569]
ALTITUD	0.999 991 8 [0.999 584 66, 1.000 358]
PORC_EJEC	0.997 763 2 [0.968 810 51, 1.026 263]
DIAS_ATRASO	1.000 316 6 [0.998 776 02, 1.001 894]
Num.Obs.	81
AIC	347.3
BIC	366.5
Log.Lik.	-165.656
F	0.696
RMSE	5.80
+ p \num{< 0.1}, * p \num{< 0.05}, ** p \num{< 0.01}, *** p \num{< 0.001}	

```

modelo_bn = list('Binomial Negativa asegurado (I)' = rbn)

formatoNum <- function(x) format(x, digits = 4, scientific = FALSE)

modelsummary(modelo_bn,
  fmt=formatoNum, # uso mi formula
  exponentiate = T, # exponenciar!!!!
  statistic = 'conf.int',
  title = "Regresión Binomial Negativa - coeficientes exponenciados",
  stars = TRUE,
  output = "kableExtra")

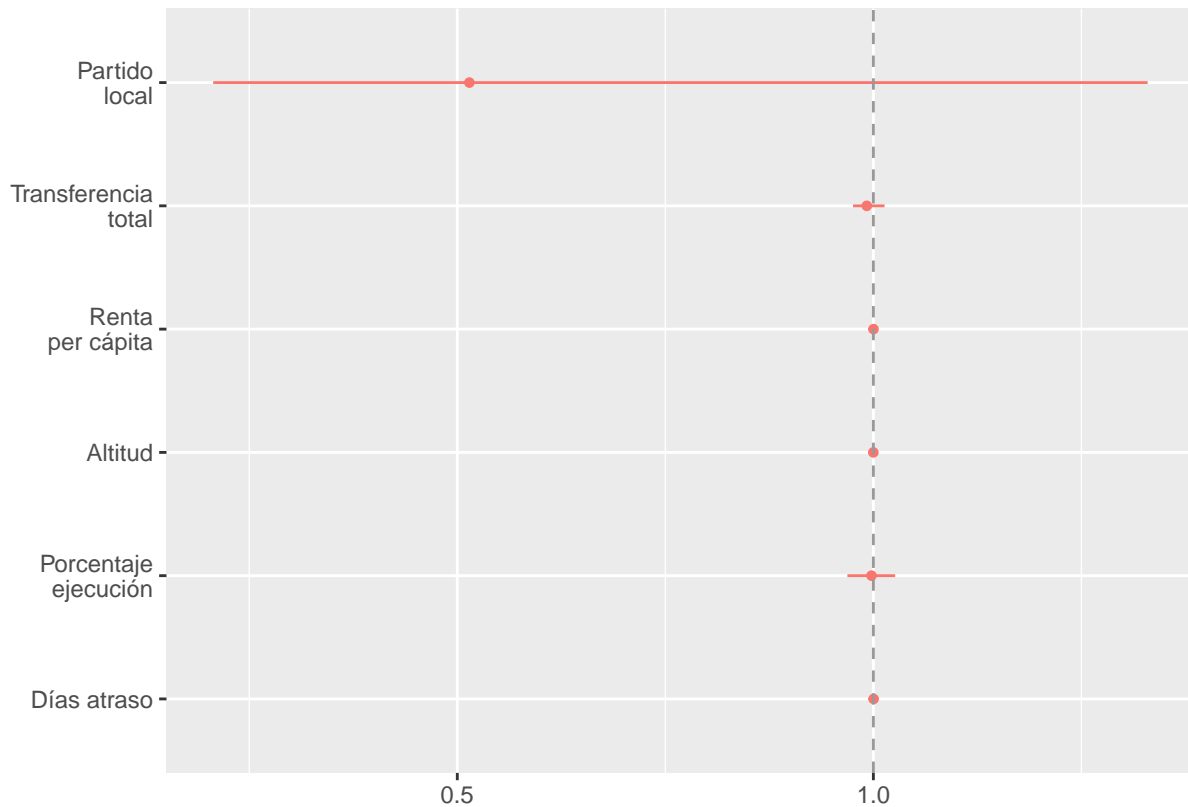
```

Podemos verlo mejor con el siguiente gráfico:

```

library(ggplot2)
dotwhisker::dwplot(list(BinomialNegativa=rbn),exp=T) + scale_y_discrete(labels=c("Días atraso", "Porcent
  xintercept = 1,
  colour = "grey60",
  linetype = 2
)

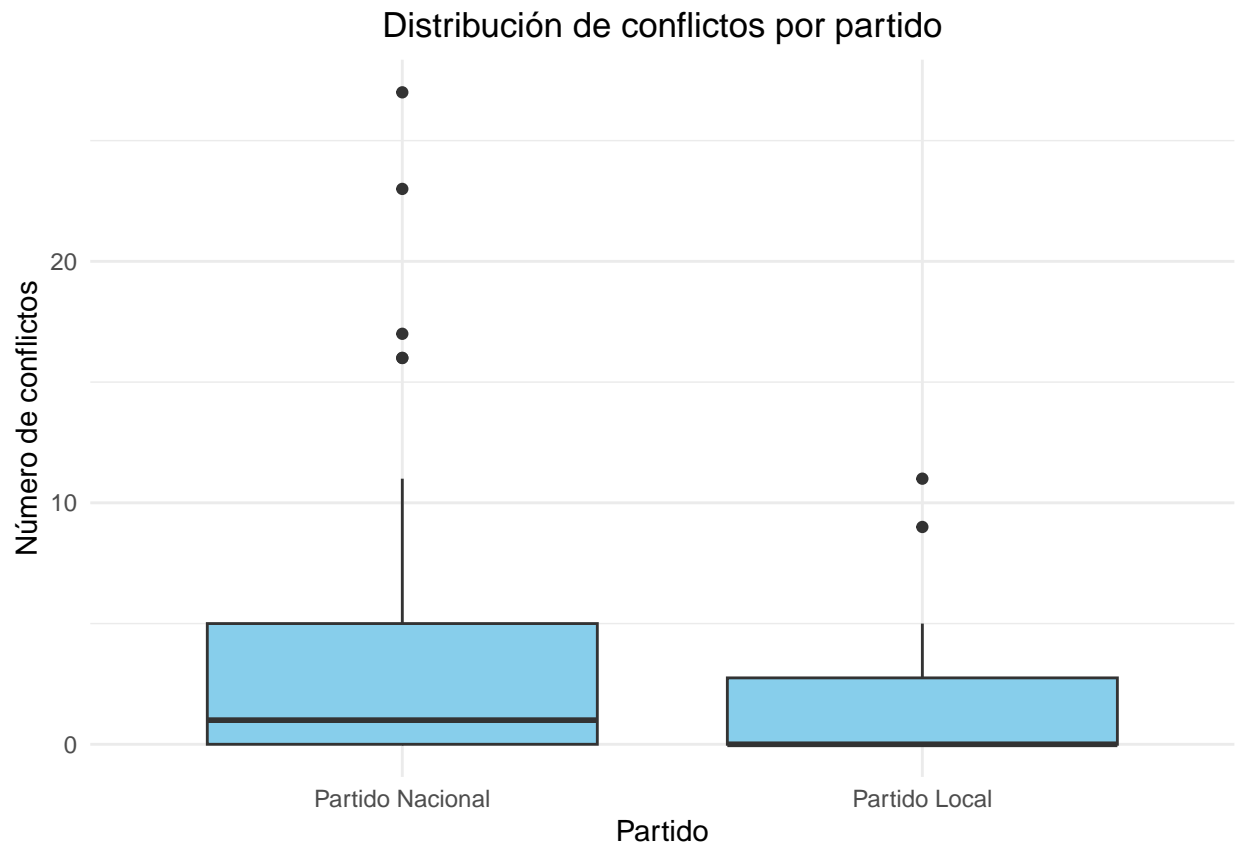
```

Como podemos apreciar, ninguna de nuestras variables resultan significativas.

Distribución de protestas por tipo de partido (nacional - local)

```
ggplot(protestas, aes(x = PP_Nac_Loc, y = CONFLICTOS)) +
  geom_boxplot(fill = "skyblue") +
  scale_x_discrete(labels = c(
    "1" = "Partido Nacional",
    "2" = "Partido Local")) +
  labs(title = "Distribución de conflictos por partido",
       x = "Partido",
       y = "Número de conflictos") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



¿Y cómo se distribuye el tipo de partido en todos los distritos?

```
library(ggplot2)
library(dplyr)

protestas %>%
  count(PP_Nac_Loc) %>%
  ggplot(aes(x= PP_Nac_Loc, y = n, fill = PP_Nac_Loc)) +
  geom_col() +
  scale_x_discrete(labels = c(
    "1" = "Partido Nacional",
    "2" = "Partido Local")) +
  labs(tittle= "Distribución de distritos por tipo de partido",
       x = "Tipo de partido",
       y = "Número de distritos") +
  theme_minimal()
```

