Trb Clase 12

Julián Estivent Serna Triana

27/4/2022

Carga de datos

```
datos_xp <- read_excel("C:/Users/admin/Downloads/XPABLO (2).xlsx")
View(datos_xp)
```

Modelo de regresión simple

```
names(datos_xp)
```

```
##  [1] "id"   "Long" "Lat"  "z"    "MO"   "Ca"   "Mg"   "K"    "Na"   "CICE"
## [11] "CE"   "Fe"   "Cu"   "Zn"   "cos"  "mod1" "mod2" "mod3" "mod4"
```

Redefiniendo df1

```
df1 <- datos_xp[-c(15,16,17,18,19)]
names(df1)
```

```
##  [1] "id"   "Long" "Lat"  "z"    "MO"   "Ca"   "Mg"   "K"    "Na"   "CICE"
## [11] "CE"   "Fe"   "Cu"   "Zn"
```

K

Trabajando con K/Cu

```
model_1 <- lm(K ~ Cu, data = df1)
summary(model_1)
```

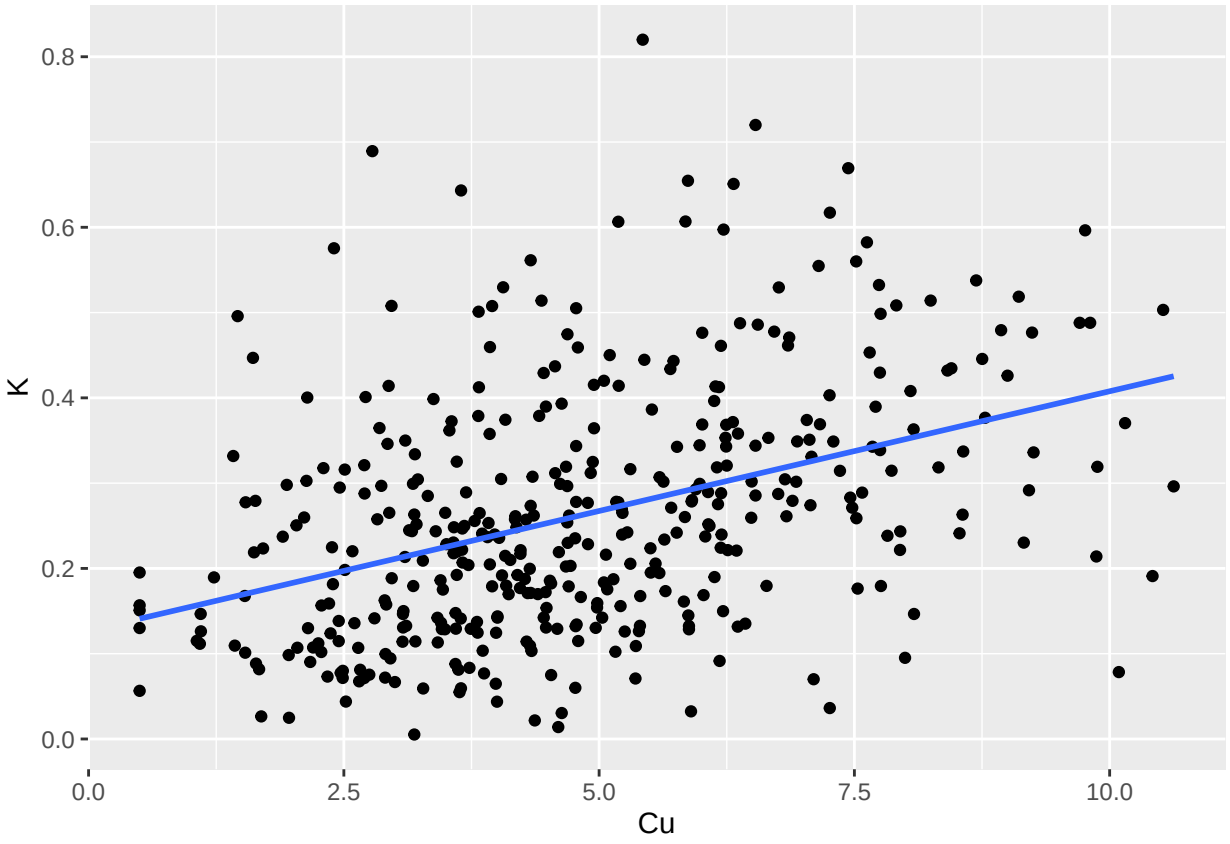
```
##
## Call:
## lm(formula = K ~ Cu, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33182 -0.08915 -0.01915  0.06818  0.54068
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.126940   0.016671   7.615 1.92e-13 ***
## Cu          0.028079   0.003176   8.840  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1322 on 401 degrees of freedom
## Multiple R-squared:  0.1631, Adjusted R-squared:  0.161
## F-statistic: 78.15 on 1 and 401 DF,  p-value: < 2.2e-16
```

$$Y_K = 0.127 + 0.028 X_{Cu}$$

```
ggplot(df1, aes(y = K, x = Cu)) +
  geom_point()+
  geom_smooth(method='lm', se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Filtrado Cu > 10

```r
df_2 <- df1 |>
  filter(Cu <= 10)
df_2
```

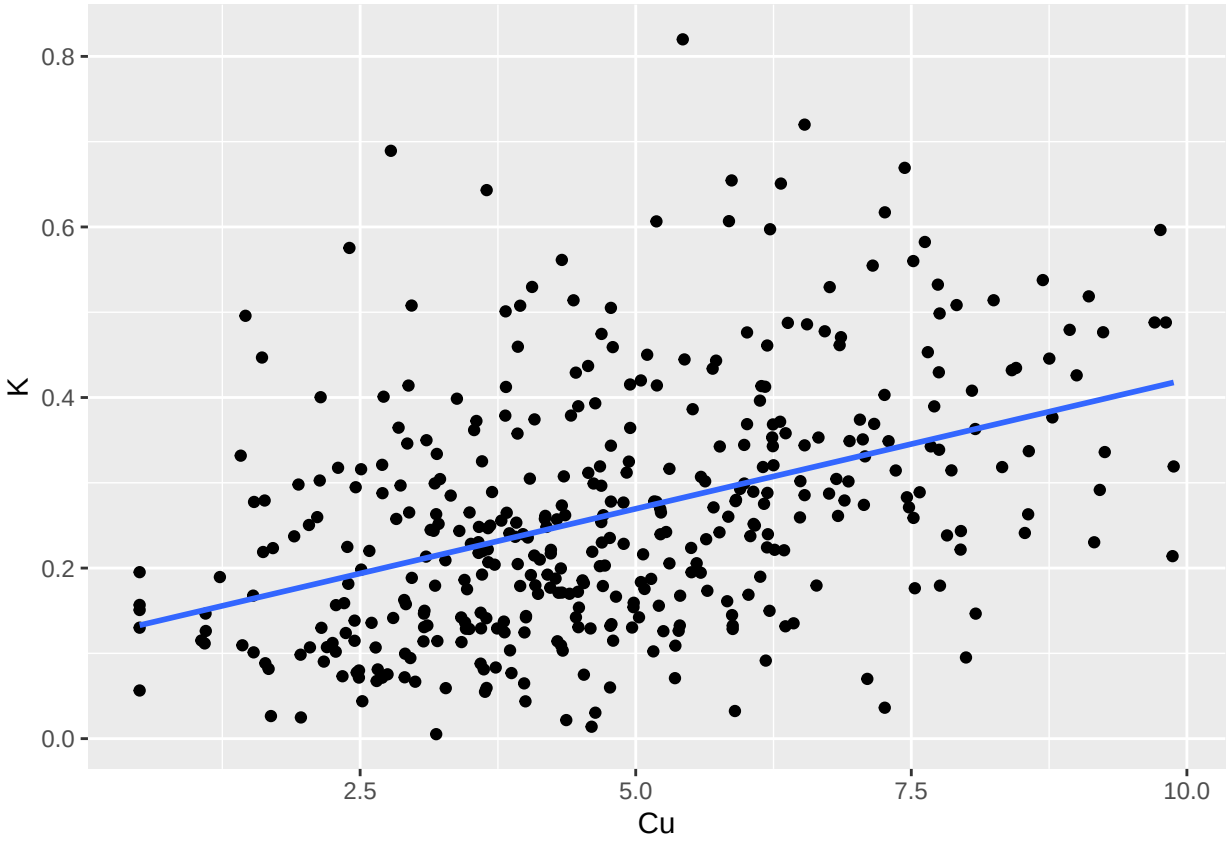
```
## # A tibble: 398 x 14
##       id  Long   Lat     z    MO    Ca    Mg      K    Na  CICE    CE    Fe
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
##  1     1 -72.6  8.08   120  2.09  7.83 1.56  0.175  0.291  9.85 0.130 133.
##  2     2 -72.6  8.08   119  1.65  3.95 0.771 0.496  0.136  5.36 0.126  29.7
##  3     3 -72.6  8.08   111  1.65  5.88 1.23  0.273  0.135  7.52 0.287 237.
##  4     4 -72.6  8.08   114  2.48  5.62 1.13  0.217  0.163  7.13 0.415 331.
##  5     5 -72.6  8.09   115  3.01 11.4  2.36  0.501  0.292 14.6  0.269 281.
##  6     6 -72.6  8.09   109  1.93  7.49 1.56  0.244  0.115  9.41 0.410 258.
##  7     7 -72.6  8.09   116  2.86 10.9  2.40  0.195  0.282 13.8  0.141 167.
##  8     8 -72.6  8.10   109  2.20 12.1  2.73  0.0438 0.420 15.3  0.163  54.5
##  9     9 -72.6  8.10   109  2.64 15.7  5.54  0.265  0.454 22.9  0.173  96.4
## 10    10 -72.6  8.10   115  2.06  7.96 1.78  0.133  0.308 10.2  0.245 446.
## # ... with 388 more rows, and 2 more variables: Cu <dbl>, Zn <dbl>
```

```r
model_2 <- lm(K ~ Cu, data = df_2)
summary(model_2)
```

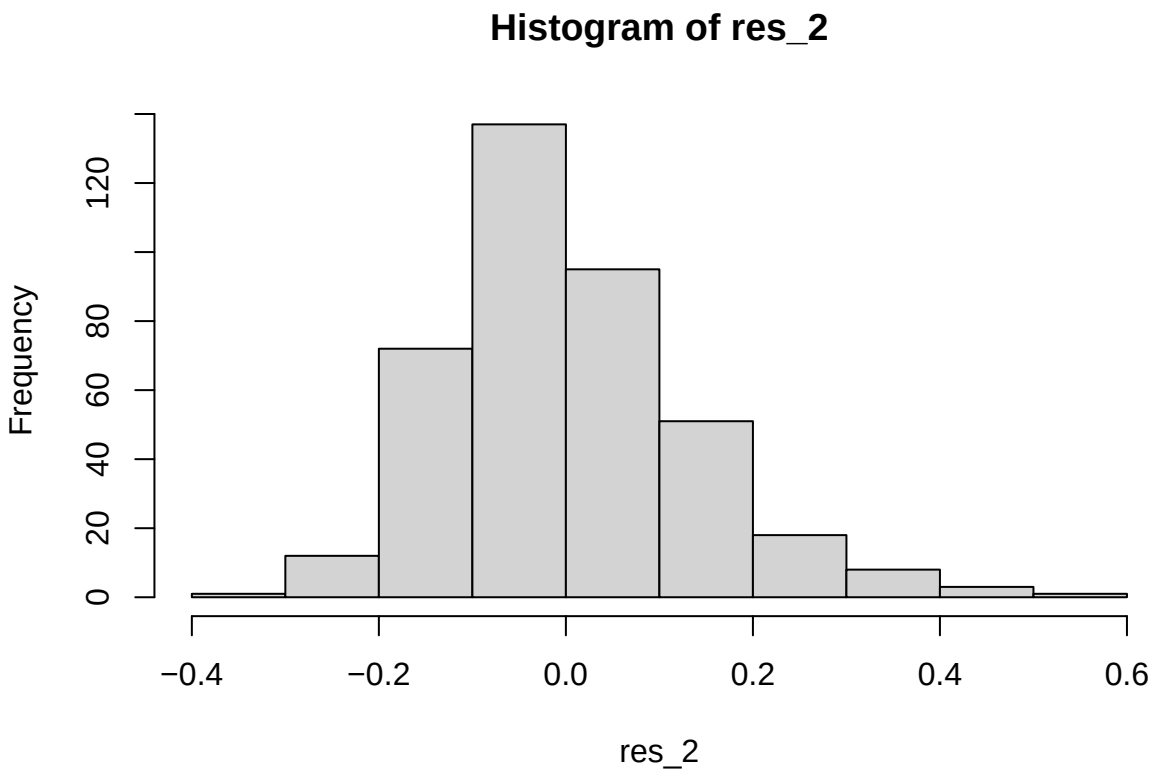
```
##
## Call:
## lm(formula = K ~ Cu, data = df_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30186 -0.08899 -0.01960  0.06393  0.53751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.117765   0.017018    6.92 1.82e-11 ***
## Cu          0.030353   0.003303    9.19  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1311 on 396 degrees of freedom
## Multiple R-squared:  0.1758, Adjusted R-squared:  0.1737
## F-statistic: 84.45 on 1 and 396 DF,  p-value: < 2.2e-16
```

```r
ggplot(df_2, aes(y = K, x = Cu)) +
  geom_point()+
  geom_smooth(method='lm', se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



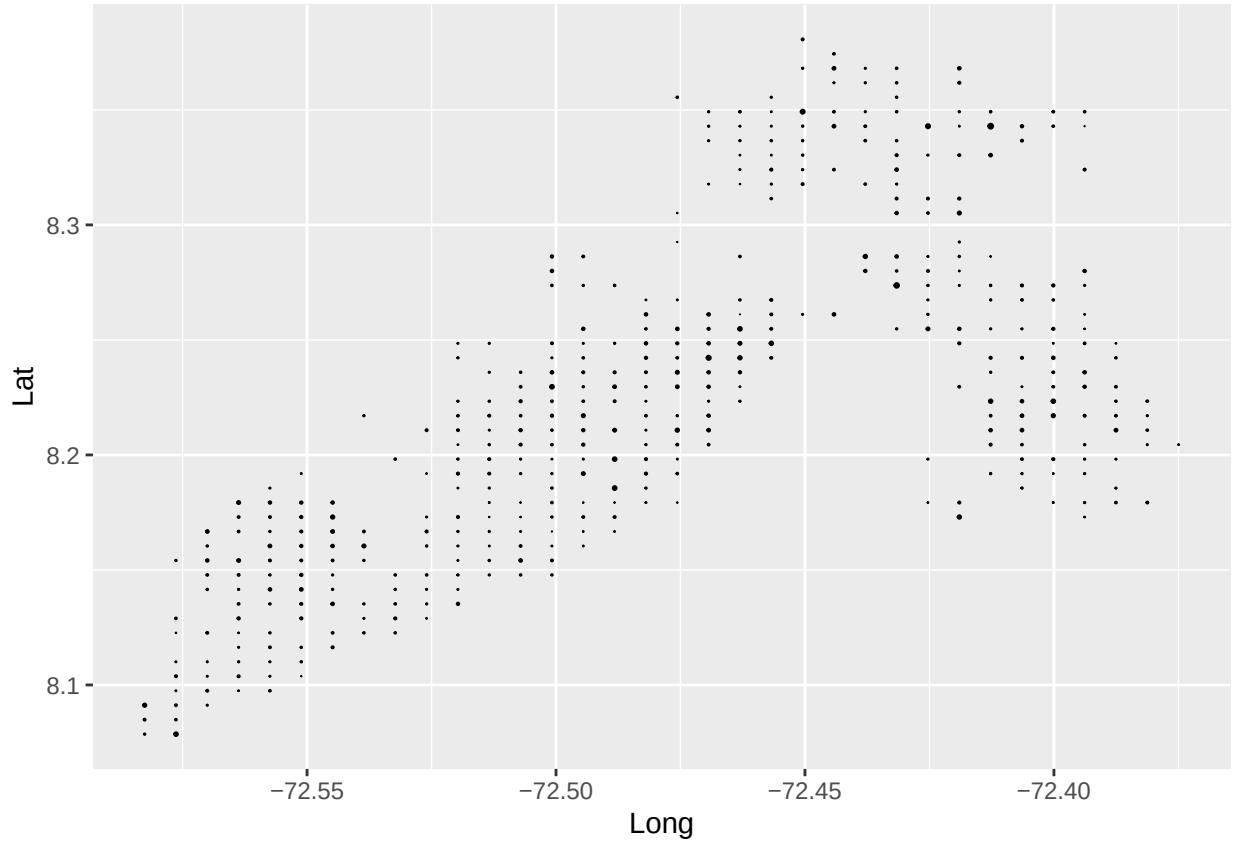

```
res_2 <- model_2$residuals
hist(res_2)
```



Sin valor absoluto (Con valores negativos)

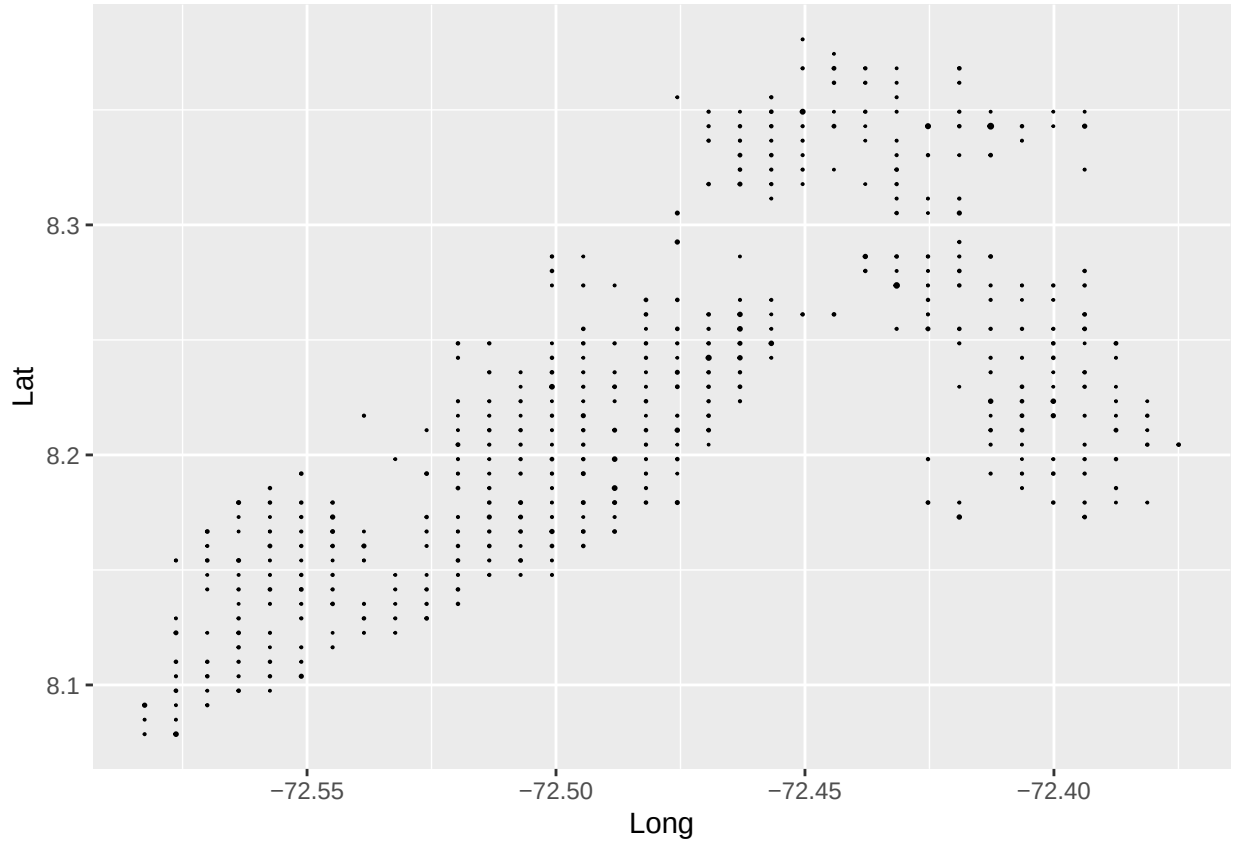
```
res_2 <- model_2$residuals

ggplot(df_2, aes(Long, Lat))+
  geom_point(size = res_2)
```

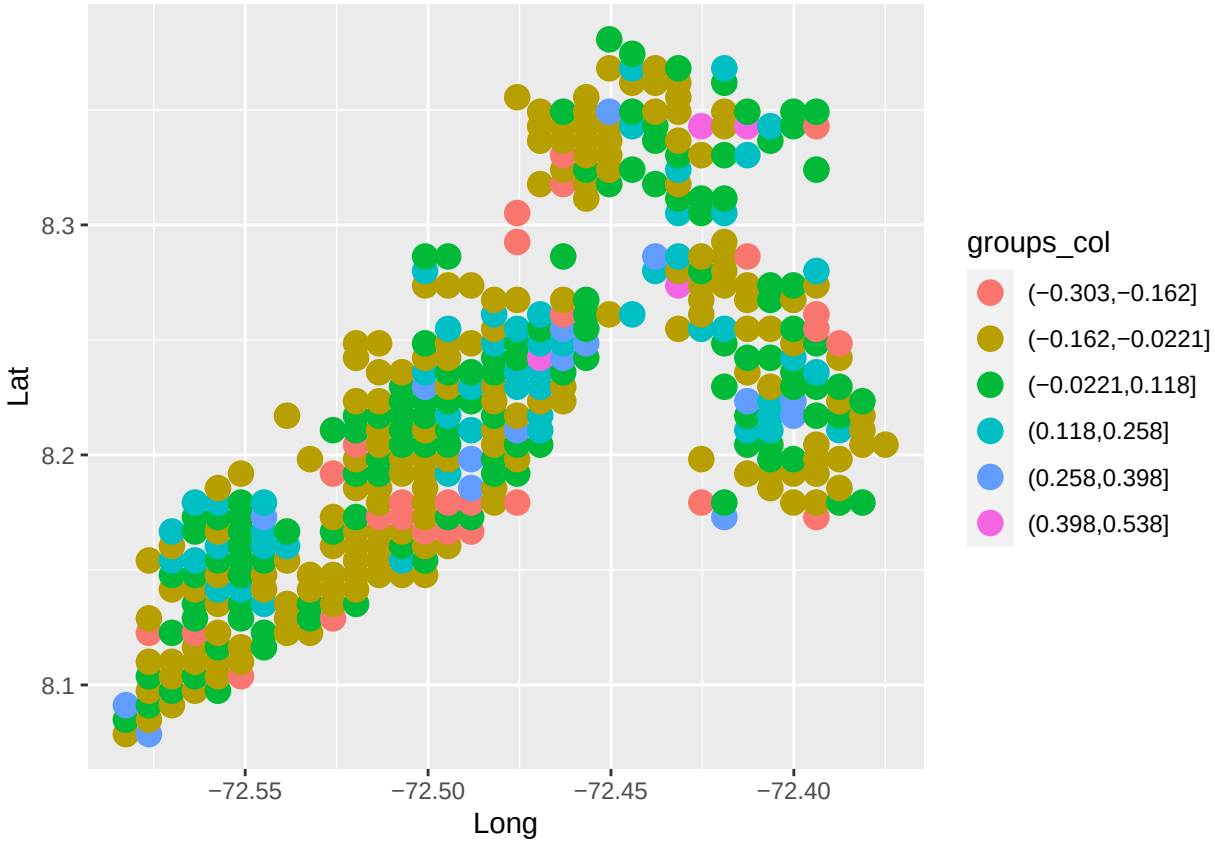


Con correción

```
ggplot(df_2, aes(Long, Lat))+
  geom_point(size = abs(res_2))
```




```
groups_col <- cut(res_2, breaks = 6)
#color <-
ggplot(df_2, aes(Long, Lat, color = groups_col))+
  geom_point(size = 4)
```



Moran Index para residuales

```
matriz_dist <- as.matrix(dist(cbind(x = df_2$Long, y = df_2$Lat)))

dim(matriz_dist)
```

```
## [1] 398 398
```

```
m_dist_inv <- 1/matriz_dist
m_dist_inv[is.infinite(m_dist_inv)] <- 0
diag(m_dist_inv) <- 0
```

```
Moran.I(res_2, m_dist_inv)
```

```
## $observed
## [1] 0.03271635
##
## $expected
## [1] -0.002518892
##
## $sd
## [1] 0.004317647
```

```
##
## $p.value
## [1] 2.220446e-16
```

Modelo de regresión multiple

```
model_3 <- lm(K ~ Cu + CE, data = df1)
summary(model_3)
```

```
##
## Call:
## lm(formula = K ~ Cu + CE, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32677 -0.08781 -0.02418  0.06364  0.51077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.096703   0.020290   4.766 2.63e-06 ***
## Cu          0.027628   0.003159   8.746  < 2e-16 ***
## CE          0.101197   0.039263   2.577   0.0103 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1312 on 400 degrees of freedom
## Multiple R-squared:  0.1768, Adjusted R-squared:  0.1727
## F-statistic: 42.95 on 2 and 400 DF,  p-value: < 2.2e-16
```

$$Y_K = 0.097 + 0.028 X_{Cu} + 0.101 z$$

```
res_3 <- model_3$residuals
```

Moran Index para residuales model 3

```
matriz_dist <- as.matrix(dist(cbind(x = df1$Long, y = df1$Lat)))

dim(matriz_dist)
```

```
## [1] 403 403
```

```
m_dist_inv <- 1/matriz_dist
m_dist_inv[is.infinite(m_dist_inv)] <- 0
diag(m_dist_inv) <- 0
```

```r
Moran.I(res_3, m_dist_inv)
```

```
## $observed
## [1] 0.031757
##
## $expected
## [1] -0.002487562
##
## $sd
## [1] 0.004255965
##
## $p.value
## [1] 8.881784e-16
```

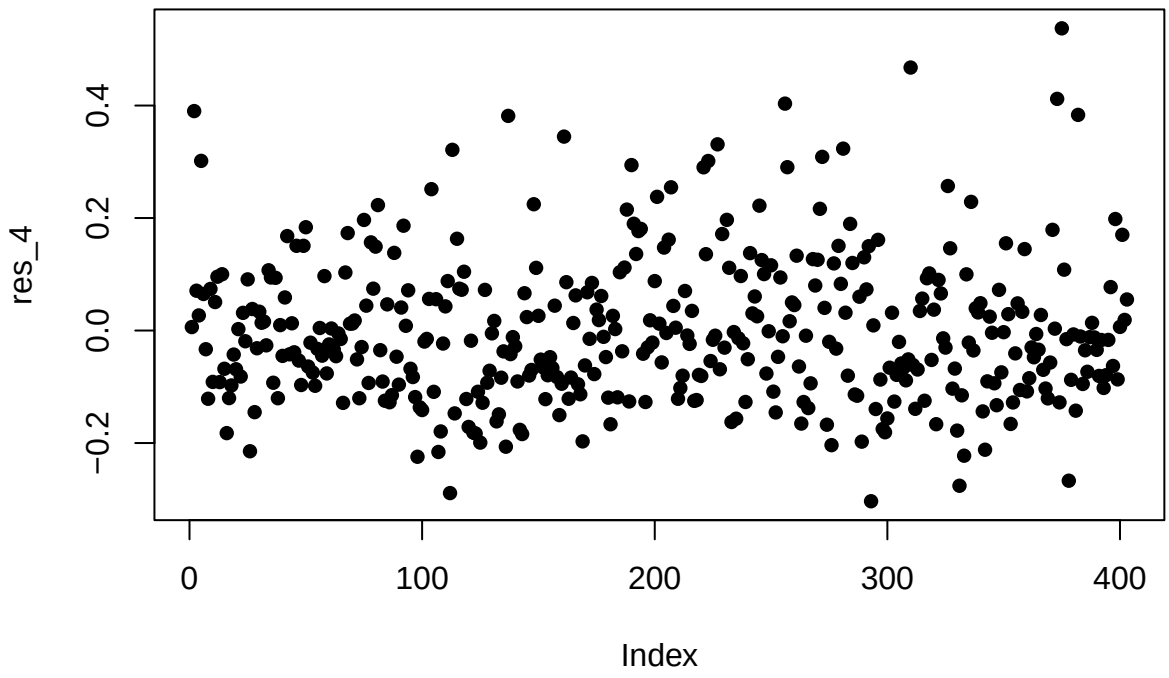
```r
model_4 <- lm(K ~ Cu + Long + Lat + I(Long**2) + I(Lat**2), data = df1) #datos georrefenciados
summary(model_4)
```

```
##
## Call:
## lm(formula = K ~ Cu + Long + Lat + I(Long^2) + I(Lat^2), data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30345 -0.09064 -0.01688  0.07188  0.53720
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.008e+04  1.538e+04   1.956  0.05121 .
## Cu           2.667e-02  3.294e-03   8.095 7.06e-15 ***
## Long         8.384e+02  4.259e+02   1.969  0.04968 *
## Lat          7.020e+01  2.428e+01   2.891  0.00405 **
## I(Long^2)    5.787e+00  2.939e+00   1.969  0.04963 *
## I(Lat^2)    -4.237e+00  1.470e+00  -2.883  0.00415 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.131 on 397 degrees of freedom
## Multiple R-squared:  0.1857, Adjusted R-squared:  0.1754
## F-statistic:  18.1 on 5 and 397 DF,  p-value: 3.465e-16
```

```r
res_4 <- model_4$residuals
shapiro.test(res_4)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_4
## W = 0.95988, p-value = 4.923e-09
```

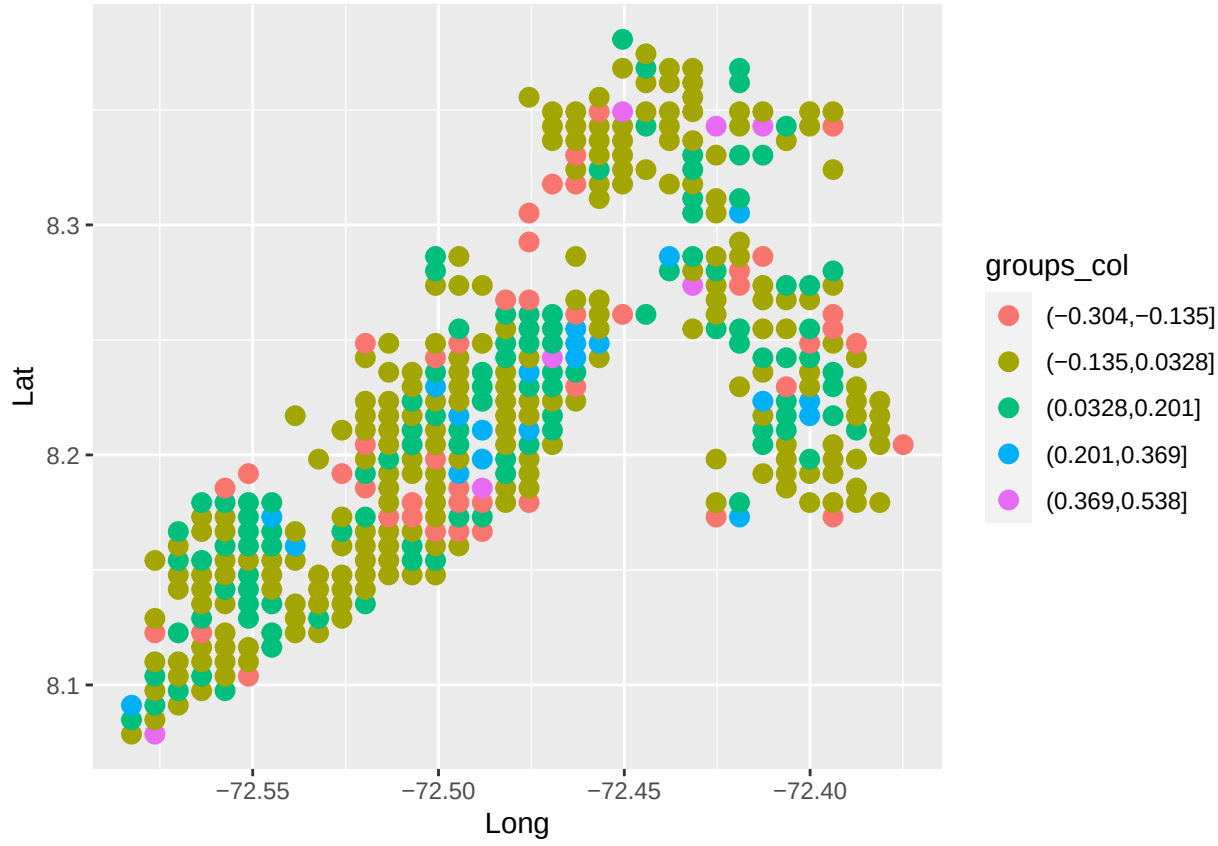
```r
plot(res_4, pch = 16)
```




```
Moran.I(res_4, m_dist_inv)
```

```
## $observed
## [1] 0.02114097
##
## $expected
## [1] -0.002487562
##
## $sd
## [1] 0.0042552
##
## $p.value
## [1] 2.810341e-08
```

```
groups_col <- cut(res_4, breaks = 5)
ggplot(df1, aes(Long, Lat, color = groups_col))+
  geom_point(size = 3)
```




```
model_5 <- lm(K ~ Cu + I(Long**2) + I(Lat**2) + I(Cu**2)+ Long + Lat , data = df1) #datos georrefenciad
summary(model_5)
```

```
##
## Call:
## lm(formula = K ~ Cu + I(Long^2) + I(Lat^2) + I(Cu^2) + Long +
##     Lat, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30330 -0.09048 -0.01736  0.07082  0.53773
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.053e+04  1.574e+04   1.940  0.05308 .
## Cu           2.489e-02  1.316e-02   1.892  0.05926 .
## I(Long^2)    5.873e+00  3.007e+00   1.953  0.05149 .
## I(Lat^2)    -4.253e+00  1.476e+00  -2.882  0.00417 **
## I(Cu^2)      1.700e-04  1.217e-03   0.140  0.88895
## Long         8.510e+02  4.358e+02   1.953  0.05153 .
## Lat          7.046e+01  2.439e+01   2.889  0.00407 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1312 on 396 degrees of freedom
## Multiple R-squared:  0.1857, Adjusted R-squared:  0.1734
## F-statistic: 15.05 on 6 and 396 DF,  p-value: 1.553e-15
```

```
res_5 <- model_5$residuals
Moran.I(res_5, m_dist_inv)
```

```
## $observed
## [1] 0.02115172
##
## $expected
## [1] -0.002487562
##
## $sd
## [1] 0.004255156
##
## $p.value
## [1] 2.769056e-08
```

Modelos de regresión espacial

```
xy = as.matrix(df1[,c(2,3)])
```

```
contnb <- dnearneigh(coordinates(xy),0,380000,longlat = F)
dlist <- nbdists(contnb, xy)
dlist <- lapply(dlist, function(x) 1/x)            #inverse distance
Wve <- nb2listw(contnb,glist=dlist,style = "W")       #W matriz-standarized
```

Modelo autoregresivo puro

```
model_auto <- spautolm(K ~ 1,data = df1,listw=Wve)
summary(model_auto)
```

```
##
## Call: spautolm(formula = K ~ 1, data = df1, listw = Wve)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.261475 -0.106784 -0.022284  0.075295  0.551268
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20872    0.10283  2.0297  0.04239
##
## Lambda: 0.93328 LR test value: 31.281 p-value: 2.2325e-08
```

```
## Numerical Hessian standard error of lambda: 0.065028
##
## Log likelihood: 224.5228
## ML residual variance (sigma squared): 0.018969, (sigma: 0.13773)
## Number of observations: 403
## Number of parameters estimated: 3
## AIC: -443.05
```

$$Y_K = \alpha_0 + \lambda W Y_K + uu = \rho W u + \epsilon$$

Si $\rho = 0$, $\mathbf{u} = \epsilon$

$$Y_K = \alpha_0 + \lambda W Y_K + \epsilon$$

```
res_6 <- model_auto$fit$residuals
```

```
Moran.I(res_6, m_dist_inv)
```

```
## $observed
## [1] 0.02593183
##
## $expected
## [1] -0.002487562
##
## $sd
## [1] 0.004258952
##
## $p.value
## [1] 2.50866e-11
```