import data

```r
library(reshape2)
library(ggplot2)
```

```r
crime <- read.table('APPENC02(1).txt')
colnames(crime) <- c("id","county","state","area","population",
                     "perc.young","perc.old","physicians",
                     "hospital.beds","n.crimes","perc.hs",
                     "perc.bs","perc.poor","unemployment",
                     "per.income","tot.income","region")
crime['pop.density'] = crime$population/crime$area
crime['physician.per.1000'] = crime$physicians/(crime$population/1000)
crime['beds.per.1000'] = crime$hospital.beds/(crime$population/1000)
crime['crime.rate.per.1000'] = crime$n.crimes/(crime$population/1000)
head(crime)
```

```
##   id      county state area population perc.young perc.old physicians
## 1  1 Los_Angeles    CA 4060    8863164       32.1      9.7      23677
## 2  2        Cook    IL  946    5105067       29.2     12.4      15153
## 3  3      Harris    TX 1729    2818199       31.3      7.1       7553
## 4  4   San_Diego    CA 4205    2498016       33.5     10.9       5905
## 5  5      Orange    CA  790    2410556       32.6      9.2       6062
## 6  6       Kings    NY   71    2300664       28.3     12.4       4861
##   hospital.beds n.crimes perc.hs perc.bs perc.poor unemployment per.income
## 1         27700   688936    70.0    22.3      11.6          8.0      20786
## 2         21550   436936    73.4    22.8      11.1          7.2      21729
## 3         12449   253526    74.9    25.4      12.5          5.7      19517
## 4          6179   173821    81.9    25.3       8.1          6.1      19588
## 5          6369   144524    81.2    27.8       5.2          4.8      24400
## 6          8942   680966    63.7    16.6      19.5          9.5      16803
##   tot.income region pop.density physician.per.1000 beds.per.1000
## 1     184230      4   2183.0453           2.671394      3.125295
## 2     110928      2   5396.4767           2.968227      4.221296
## 3      55003      3   1629.9589           2.680080      4.417360
## 4      48931      4    594.0585           2.363876      2.473563
## 5      58818      4   3051.3367           2.514773      2.642129
## 6      38658      1  32403.7183           2.112868      3.886704
##   crime.rate.per.1000
## 1            77.73026
## 2            85.58869
## 3            89.96029
## 4            69.58362
## 5            59.95463
## 6           295.98672
```

```r
dim(crime)
```

```
## [1] 440  21
```

```r
lm.crime.full <- lm(crime.rate.per.1000 ~ area + perc.young + perc.old +
                    hospital.beds + perc.bs + perc.poor + unemployment +
                    per.income + I(region) + pop.density + physician.per.1000 +
                    beds.per.1000, data = crime)
summary(lm.crime.full)
```

```
##
## Call:
## lm(formula = crime.rate.per.1000 ~ area + perc.young + perc.old +
##     hospital.beds + perc.bs + perc.poor + unemployment + per.income +
##     I(region) + pop.density + physician.per.1000 + beds.per.1000,
##     data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.330 -11.885  -1.191  10.414  80.014
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -3.025e+01  1.589e+01  -1.903  0.05770 .
## area               -6.351e-04  6.477e-04  -0.980  0.32744
## perc.young          8.120e-01  3.382e-01   2.401  0.01680 *
## perc.old           -9.208e-02  3.080e-01  -0.299  0.76512
## hospital.beds       1.325e-03  4.652e-04   2.848  0.00462 **
## perc.bs            -1.232e-01  2.543e-01  -0.485  0.62827
## perc.poor           1.745e+00  3.312e-01   5.268 2.19e-07 ***
## unemployment       -1.659e-01  5.309e-01  -0.313  0.75479
## per.income          8.746e-04  4.759e-04   1.838  0.06676 .
## I(region)           9.559e+00  1.037e+00   9.219  < 2e-16 ***
## pop.density         4.577e-03  4.827e-04   9.481  < 2e-16 ***
## physician.per.1000 -1.413e+00  1.042e+00  -1.356  0.17581
## beds.per.1000       3.242e+00  7.981e-01   4.062 5.80e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.44 on 427 degrees of freedom
## Multiple R-squared:  0.5572, Adjusted R-squared:  0.5448
## F-statistic: 44.78 on 12 and 427 DF,  p-value: < 2.2e-16
```

```r
step(lm.crime.full,direction = "backward")$anova
```

```
## Start:  AIC=2577.49
## crime.rate.per.1000 ~ area + perc.young + perc.old + hospital.beds +
##     perc.bs + perc.poor + unemployment + per.income + I(region) +
##     pop.density + physician.per.1000 + beds.per.1000
##
##                      Df Sum of Sq    RSS    AIC
## - perc.old            1      30.4 145195 2575.6
## - unemployment        1      33.2 145198 2575.6
## - perc.bs             1      79.8 145245 2575.7
## - area                1     326.8 145492 2576.5
## - physician.per.1000  1     625.1 145790 2577.4
## <none>                            145165 2577.5
## - per.income          1    1148.4 146313 2579.0
## - perc.young          1    1959.2 147124 2581.4
## - hospital.beds       1    2756.8 147922 2583.8
## - beds.per.1000       1    5608.4 150773 2592.2
## - perc.poor           1    9435.8 154601 2603.2
## - I(region)           1   28890.8 174056 2655.4
## - pop.density         1   30559.9 175725 2659.6
##
```

```
## Step:  AIC=2575.59
## crime.rate.per.1000 ~ area + perc.young + hospital.beds + perc.bs +
##     perc.poor + unemployment + per.income + I(region) + pop.density +
##     physician.per.1000 + beds.per.1000
##
##                      Df Sum of Sq    RSS    AIC
## - unemployment        1      42.9 145238 2573.7
## - perc.bs             1      77.5 145273 2573.8
## - area                1     339.2 145535 2574.6
## - physician.per.1000  1     624.1 145819 2575.5
## <none>                            145195 2575.6
## - per.income          1    1164.8 146360 2577.1
## - hospital.beds       1    2748.3 147944 2581.8
## - perc.young          1    2920.8 148116 2582.3
## - beds.per.1000       1    5770.3 150966 2590.7
## - perc.poor           1    9887.0 155082 2602.6
## - I(region)           1   29185.7 174381 2654.2
## - pop.density         1   30612.4 175808 2657.8
##
## Step:  AIC=2573.72
## crime.rate.per.1000 ~ area + perc.young + hospital.beds + perc.bs +
##     perc.poor + per.income + I(region) + pop.density + physician.per.1000 +
##     beds.per.1000
##
##                      Df Sum of Sq    RSS    AIC
## - perc.bs             1      47.1 145285 2571.9
## - area                1     388.4 145627 2572.9
## - physician.per.1000  1     647.0 145885 2573.7
## <none>                            145238 2573.7
## - per.income          1    1124.4 146363 2575.1
## - hospital.beds       1    2797.2 148035 2580.1
## - perc.young          1    2927.6 148166 2580.5
## - beds.per.1000       1    6476.3 151715 2590.9
## - perc.poor           1   11855.9 157094 2606.2
## - pop.density         1   30578.8 175817 2655.8
## - I(region)           1   31250.1 176488 2657.5
##
## Step:  AIC=2571.86
## crime.rate.per.1000 ~ area + perc.young + hospital.beds + perc.poor +
##     per.income + I(region) + pop.density + physician.per.1000 +
##     beds.per.1000
##
##                      Df Sum of Sq    RSS    AIC
## - area                1       372 145658 2571.0
## <none>                            145285 2571.9
## - physician.per.1000  1       791 146076 2572.2
## - per.income          1      1473 146758 2574.3
## - hospital.beds       1      2938 148223 2578.7
## - perc.young          1      4124 149409 2582.2
## - beds.per.1000       1      6808 152094 2590.0
## - perc.poor           1     11848 157133 2604.3
## - pop.density         1     31142 176427 2655.3
## - I(region)           1     32793 178078 2659.4
##
```

```
## Step:  AIC=2570.99
## crime.rate.per.1000 ~ perc.young + hospital.beds + perc.poor +
##     per.income + I(region) + pop.density + physician.per.1000 +
##     beds.per.1000
##
##                       Df Sum of Sq    RSS    AIC
## <none>                              145658 2571.0
## - physician.per.1000  1       862 146520 2571.6
## - per.income          1      1606 147263 2573.8
## - hospital.beds       1      2655 148313 2576.9
## - perc.young          1      4381 150039 2582.0
## - beds.per.1000       1      7530 153188 2591.2
## - perc.poor           1     11602 157260 2602.7
## - pop.density         1     33011 178669 2658.9
## - I(region)           1     33326 178984 2659.6

##             Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                NA        NA       427   145164.9 2577.494
## 2     - perc.old  1  30.38379       428   145195.3 2575.587
## 3 - unemployment  1  42.89025       429   145238.2 2573.717
## 4      - perc.bs  1  47.14222       430   145285.4 2571.859
## 5         - area  1 372.27533       431   145657.6 2570.985
```

```r
lm.crime.fit1 <- lm(crime.rate.per.1000 ~ perc.young + hospital.beds + perc.poor + per.income +
                    I(region) + pop.density + beds.per.1000, data = crime)
summary(lm.crime.fit1)
```

```
##
## Call:
## lm(formula = crime.rate.per.1000 ~ perc.young + hospital.beds +
##     perc.poor + per.income + I(region) + pop.density + beds.per.1000,
##     data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.958 -11.812  -1.665  10.467  80.194
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.383e+01  9.812e+00  -2.428  0.01558 *
## perc.young     6.890e-01  2.125e-01   3.242  0.00128 **
## hospital.beds  1.253e-03  4.521e-04   2.772  0.00581 **
## perc.poor      1.674e+00  2.879e-01   5.813 1.19e-08 ***
## per.income     5.001e-04  3.085e-04   1.621  0.10565
## I(region)      9.007e+00  9.202e-01   9.789  < 2e-16 ***
## pop.density    4.591e-03  4.703e-04   9.763  < 2e-16 ***
## beds.per.1000  2.595e+00  5.090e-01   5.098 5.14e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.42 on 432 degrees of freedom
## Multiple R-squared:  0.5531, Adjusted R-squared:  0.5458
## F-statistic: 76.37 on 7 and 432 DF,  p-value: < 2.2e-16
```

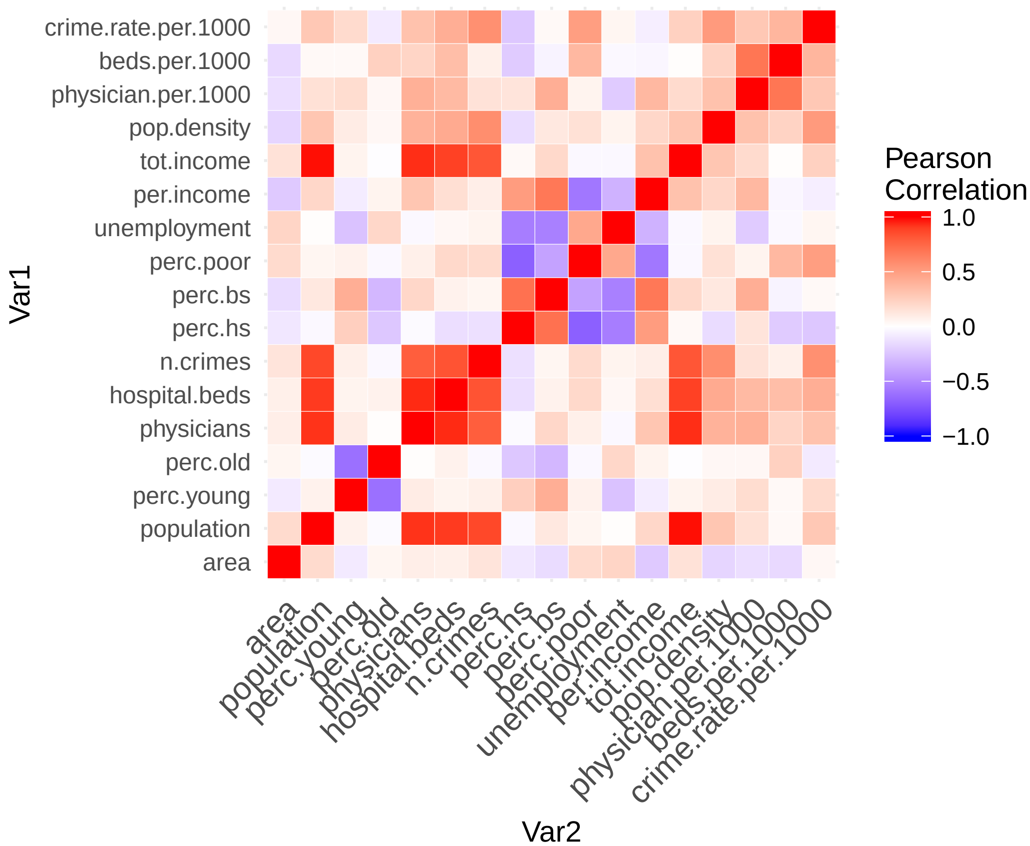
```r
step(lm.crime.fit1,direction = "backward")$anova
```

```
## Start:  AIC=2571.58
## crime.rate.per.1000 ~ perc.young + hospital.beds + perc.poor +
##     per.income + I(region) + pop.density + beds.per.1000
##
##                 Df Sum of Sq    RSS    AIC
## <none>                        146520 2571.6
## - per.income     1       892 147412 2572.2
## - hospital.beds  1      2606 149126 2577.3
## - perc.young     1      3564 150084 2580.2
## - beds.per.1000  1      8815 155335 2595.3
## - perc.poor      1     11463 157983 2602.7
## - pop.density    1     32330 178850 2657.3
## - I(region)      1     32499 179019 2657.7

##   Step Df Deviance Resid. Df Resid. Dev      AIC
## 1      NA       NA       432   146520.1 2571.583
```

Poisson regression

```r
df <- crime
```

```r
# correlation map
UNI <- 2823
set.seed(2823)
index <- sample(c(1:440))
train_df <- df[index[1:300],]
numeric_features <- c('area','population','perc.young','perc.old',
                      'physicians','hospital.beds','n.crimes','perc.hs',
                      'perc.bs','perc.poor','unemployment','per.income',
                      'tot.income','pop.density','physician.per.1000',
                      'beds.per.1000','crime.rate.per.1000')
cormat <- round(cor(train_df[numeric_features]),2)

melted_cormat <- melt(cormat)

options(repr.plot.width=12, repr.plot.height=5)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()
```



train a fisrt poisson model

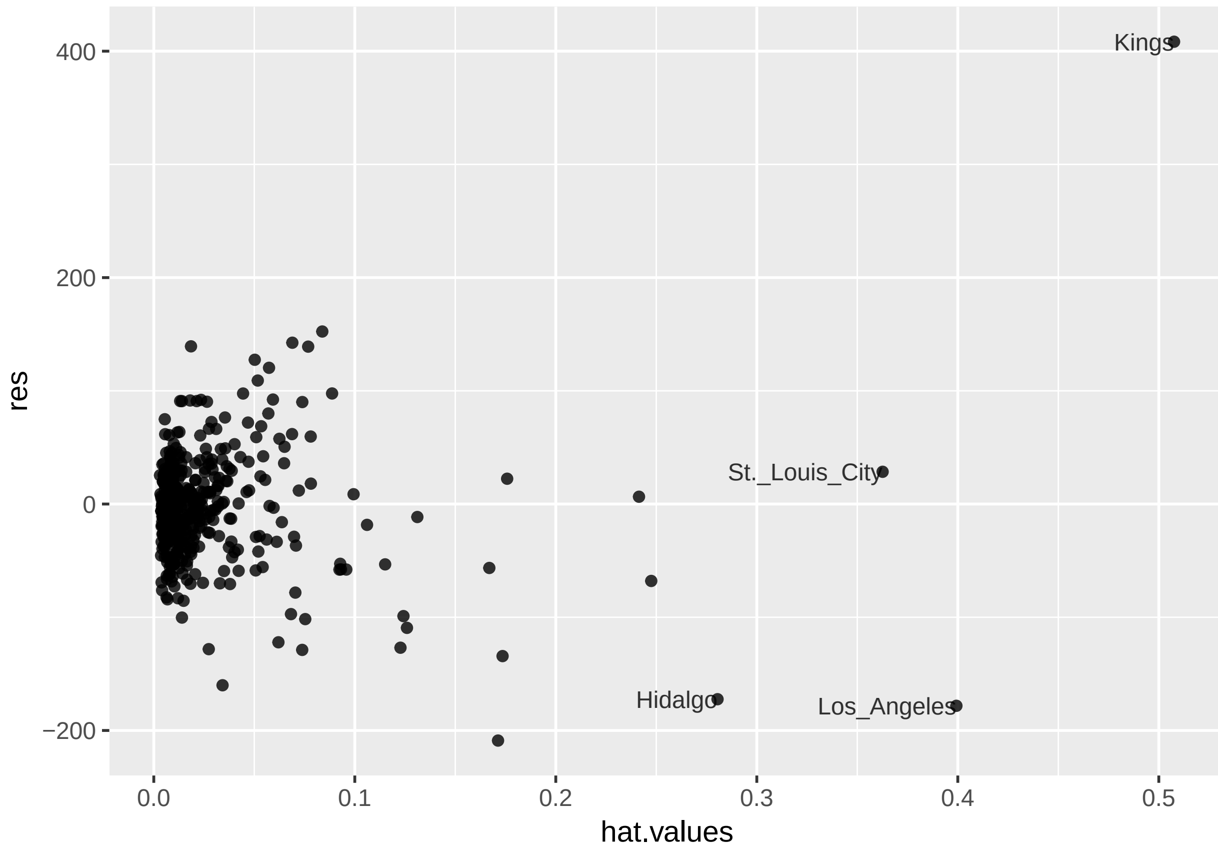
```r
#poission

m2 <- glm( n.crimes/(population/1000)~ perc.young + perc.poor + per.income +
          factor(region) + log(pop.density) + physician.per.1000 + beds.per.1000 +
          perc.bs + unemployment, family=quasipoisson, data=df, weights=(population/1000))
summary(m2)
```

```
##
## Call:
## glm(formula = n.crimes/(population/1000) ~ perc.young + perc.poor +
##     per.income + factor(region) + log(pop.density) + physician.per.1000 +
##     beds.per.1000 + perc.bs + unemployment, family = quasipoisson,
##     data = df, weights = (population/1000))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -208.91   -28.08    -7.14    18.51   408.35
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.473e+00  2.349e-01  10.524  < 2e-16 ***
## perc.young           2.878e-03  5.958e-03   0.483  0.62936
## perc.poor            3.125e-02  5.696e-03   5.485 7.07e-08 ***
## per.income          -2.264e-05  9.407e-06  -2.406  0.01653 *
```

```
## factor(region)2     4.286e-02  4.859e-02   0.882  0.37826
## factor(region)3     3.204e-01  4.857e-02   6.596 1.25e-10 ***
## factor(region)4     3.366e-01  5.080e-02   6.627 1.04e-10 ***
## log(pop.density)    2.114e-01  1.622e-02  13.032  < 2e-16 ***
## physician.per.1000 -6.088e-02  2.206e-02  -2.759  0.00604 **
## beds.per.1000       4.037e-02  1.541e-02   2.620  0.00910 **
## perc.bs             6.818e-03  5.031e-03   1.355  0.17609
## unemployment        3.246e-03  1.072e-02   0.303  0.76227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2574.241)
##
##     Null deviance: 2698854  on 439  degrees of freedom
## Residual deviance: 1092827  on 428  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

detect outlier

```r
#h=0.25
outliers <- which(hatvalues(m2)>0.25)
a1<-hatvalues(m2)
a2<-residuals(m2)
a3<-data.frame(hat.values=a1,res=a2)

options(repr.plot.width=5, repr.plot.height=4)
ggplot(data=a3,aes(hat.values,res)) +
  geom_point(alpha=0.8)+
  geom_text(data=a3[outliers,],aes(hat.values,res, label=df$county[outliers]),size=3,hjust=1,alpha=0.8)
```



delete outlier and retrain model

```
df2 = df[-outliers,]

#poission

m3 <- glm( n.crimes/(population/1000)~ perc.poor + per.income +
           factor(region) + log(pop.density) + physician.per.1000 + beds.per.1000,
           family=quasipoisson, data=df2, weights=(population/1000))
summary(m3)
```

```
##
## Call:
## glm(formula = n.crimes/(population/1000) ~ perc.poor + per.income +
##     factor(region) + log(pop.density) + physician.per.1000 +
##     beds.per.1000, family = quasipoisson, data = df2, weights = (population/1000))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -145.139   -29.725    -9.329    13.974   143.136
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.708e+00  1.240e-01  21.844  < 2e-16 ***
## perc.poor             2.741e-02  4.289e-03   6.391 4.33e-10 ***
## per.income           -7.874e-07  5.487e-06  -0.143  0.88598
```

```
## factor(region)2     2.638e-01  4.204e-02   6.273 8.68e-10 ***
## factor(region)3     5.420e-01  3.928e-02  13.800  < 2e-16 ***
## factor(region)4     5.330e-01  4.460e-02  11.950  < 2e-16 ***
## log(pop.density)    1.125e-01  1.416e-02   7.946 1.73e-14 ***
## physician.per.1000 -6.635e-03  1.546e-02  -0.429  0.66809
## beds.per.1000       3.384e-02  1.220e-02   2.774  0.00579 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1620.463)
##
##     Null deviance: 1677476  on 435  degrees of freedom
## Residual deviance:  711046  on 427  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

retrain model

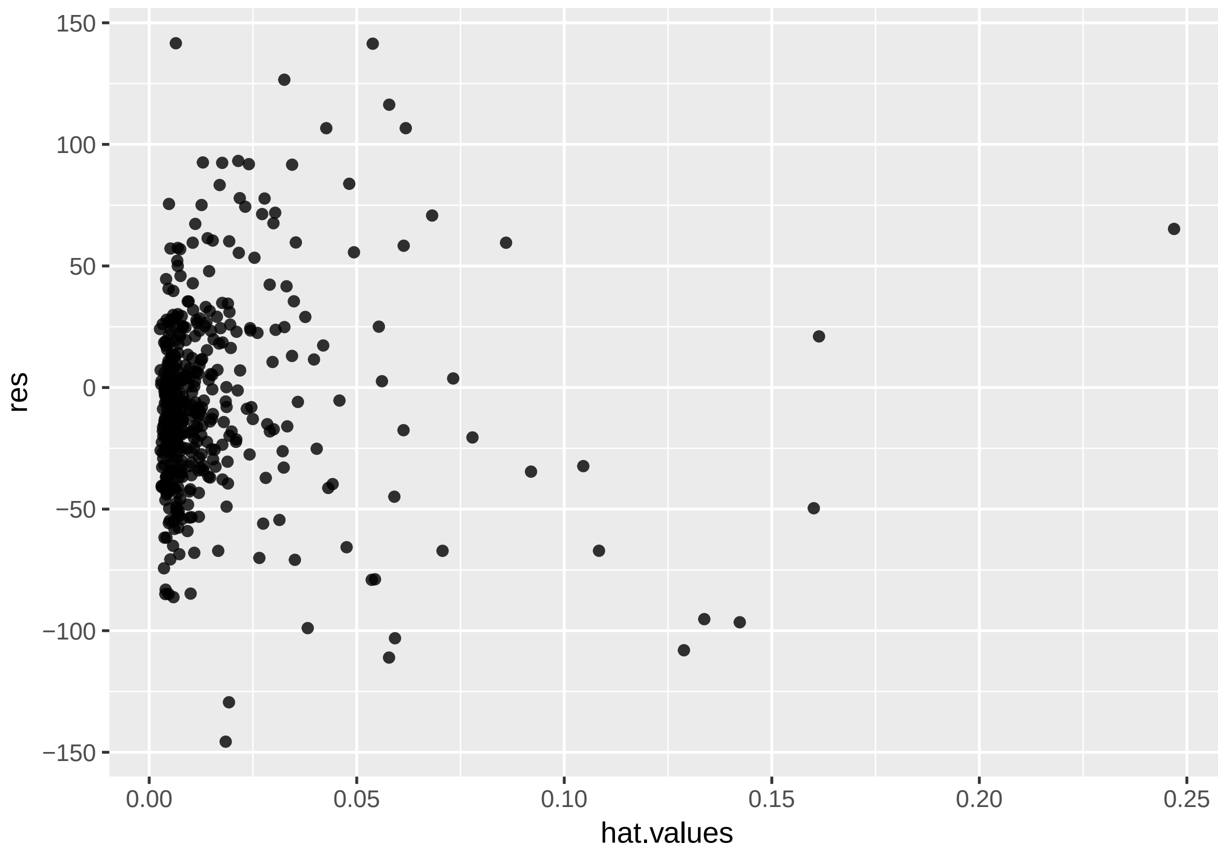
delete `per.income` and `physician.per.1000`

```r
#poission

m4 <- glm( n.crimes/(population/1000)~ perc.poor +
          factor(region) + log(pop.density) + beds.per.1000,
          family=quasipoisson, data=df2, weights=(population/1000))
summary(m4)
```

```
##
## Call:
## glm(formula = n.crimes/(population/1000) ~ perc.poor + factor(region) +
##     log(pop.density) + beds.per.1000, family = quasipoisson,
##     data = df2, weights = (population/1000))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -145.641   -29.605    -9.469    13.823   141.591
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.711299   0.083353  32.528  < 2e-16 ***
## perc.poor        0.028199   0.003245   8.690  < 2e-16 ***
## factor(region)2  0.267811   0.041238   6.494 2.31e-10 ***
## factor(region)3  0.542804   0.039066  13.895  < 2e-16 ***
## factor(region)4  0.528021   0.043359  12.178  < 2e-16 ***
## log(pop.density) 0.108331   0.011009   9.840  < 2e-16 ***
## beds.per.1000    0.029890   0.008822   3.388 0.000769 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1613.752)
##
##     Null deviance: 1677476  on 435  degrees of freedom
## Residual deviance:  711526  on 429  degrees of freedom
```

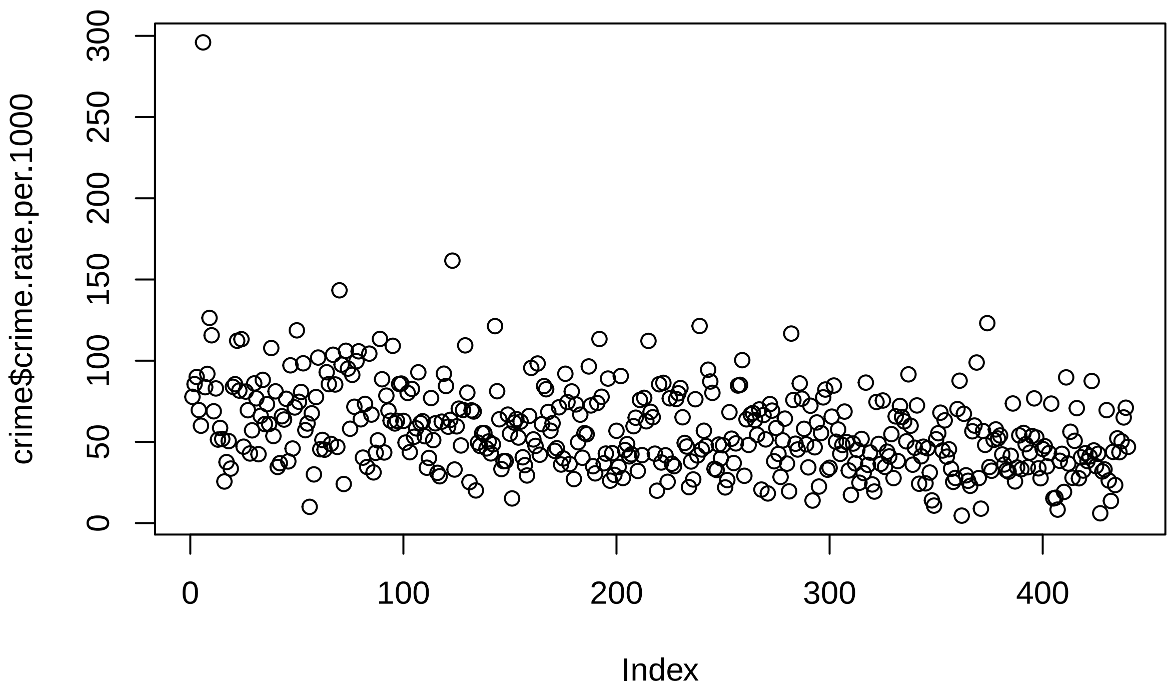
```
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
#h=0.25
outliers <- which(hatvalues(m4)>0.25)
a1<-hatvalues(m4)
a2<-residuals(m4)
a3<-data.frame(hat.values=a1,res=a2)

options(repr.plot.width=5, repr.plot.height=4)
ggplot(data=a3,aes(hat.values,res)) +
  geom_point(alpha=0.8)+
  geom_text(data=a3[outliers,],aes(hat.values,res, label=df$county[outliers]),size=3,hjust=1,alpha=0.8)
```



EDA

```
plot(crime$crime.rate.per.1000)
```




```r
logi.data <- crime
logi.data$crime.rate.level <- logi.data$crime.rate.per.1000
logi.data$crime.rate.level[which(crime$crime.rate.per.1000 <= 30)] <- 1
logi.data$crime.rate.level[which(crime$crime.rate.per.1000 > 30 &
                                   crime$crime.rate.per.1000 <= 60)] <- 2
logi.data$crime.rate.level[which(crime$crime.rate.per.1000 > 60 &
                                   crime$crime.rate.per.1000 <= 90)] <- 3
logi.data$crime.rate.level[which(crime$crime.rate.per.1000 > 90 &
                                   crime$crime.rate.per.1000 <= 120)] <- 4
logi.data$crime.rate.level[which(crime$crime.rate.per.1000 > 120)] <- 5
```

```r
library(MASS)
polr.fit1 <- polr(factor(crime.rate.level) ~ perc.young + hospital.beds + perc.poor + per.income +
                    I(region) + pop.density + beds.per.1000, data =logi.data)
summary(polr.fit1)
```

```
##
## Re-fitting to get Hessian

## Warning in sqrt(diag(vc)): NaNs produced

## Call:
## polr(formula = factor(crime.rate.level) ~ perc.young + hospital.beds +
##     perc.poor + per.income + I(region) + pop.density + beds.per.1000,
##     data = logi.data)
##
## Coefficients:
##                    Value Std. Error t value
## perc.young    1.042e-01  6.555e-03  15.889
## hospital.beds 1.208e-04  9.881e-05   1.222
## perc.poor     1.817e-01  2.307e-02   7.873
## per.income    9.042e-05        NaN     NaN
## I(region)     9.066e-01  2.113e-02  42.915
## pop.density   1.924e-04  1.130e-04   1.703
```

```
## beds.per.1000 2.982e-01  5.624e-02   5.301
##
## Intercepts:
##     Value     Std. Error t value
## 1|2    7.0874    0.0017  4077.3751
## 2|3   10.3726    0.1784    58.1441
## 3|4   13.2592    0.2743    48.3441
## 4|5   16.0088    0.2761    57.9879
##
## Residual Deviance: 834.2314
## AIC: 856.2314
```

```
1-pchisq(deviance(polr.fit1),df.residual(polr.fit1))
```

```
## [1] 0
```

```
polr.fit2 <- polr(factor(crime.rate.level) ~ perc.young + hospital.beds + perc.poor +
                     pop.density + beds.per.1000, data =logi.data)
summary(polr.fit2)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = factor(crime.rate.level) ~ perc.young + hospital.beds +
##     perc.poor + pop.density + beds.per.1000, data = logi.data)
##
## Coefficients:
##                   Value Std. Error t value
## perc.young    0.0959293  8.308e-03  11.547
## hospital.beds 0.0001506  9.881e-05   1.524
## perc.poor     0.1827593  2.323e-02   7.866
## pop.density   0.0001893  1.161e-04   1.631
## beds.per.1000 0.1664188  5.457e-02   3.050
##
## Intercepts:
##     Value    Std. Error t value
## 1|2   2.8595   0.0037   779.1939
## 2|3   5.6569   0.1748    32.3550
## 3|4   8.3175   0.2758    30.1615
## 4|5  10.9124   0.5406    20.1859
##
## Residual Deviance: 916.3924
## AIC: 934.3924
```

```
1-pchisq(deviance(polr.fit2),df.residual(polr.fit2))
```

```
## [1] 0
```

```
drop1(polr.fit2,test = "Chi")
```

```
## Single term deletions
##
## Model:
## factor(crime.rate.level) ~ perc.young + hospital.beds + perc.poor +
##     pop.density + beds.per.1000
##               Df     AIC    LRT  Pr(>Chi)
```

```
## <none>            934.39
## perc.young     1  949.59 17.197  3.37e-05 ***
## hospital.beds  1  941.96  9.568 0.0019799 **
## perc.poor      1 1000.05 67.661 < 2.2e-16 ***
## pop.density    1  943.86 11.466 0.0007088 ***
## beds.per.1000  1  942.03  9.641 0.0019029 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
library(VGAM)
```

```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```r
logit.fit1 <- vglm(crime.rate.level ~ perc.young + hospital.beds + perc.poor + per.income +
                    I(region) + pop.density + beds.per.1000, data =logi.data,
             family=cumulative(parallel=TRUE))
summary(logit.fit1)
```

```
##
## Call:
## vglm(formula = crime.rate.level ~ perc.young + hospital.beds +
##     perc.poor + per.income + I(region) + pop.density + beds.per.1000,
##     family = cumulative(parallel = TRUE), data = logi.data)
##
##
## Pearson residuals:
##                     Min       1Q   Median       3Q   Max
## logit(P[Y<=1])  -1.121 -0.39239 -0.19142 -0.07598 5.690
## logit(P[Y<=2])  -2.856 -0.56660  0.19457  0.54789 2.761
## logit(P[Y<=3])  -9.708  0.05721  0.11748  0.25326 3.409
## logit(P[Y<=4]) -12.583  0.01884  0.03196  0.06719 1.190
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  7.087e+00  1.134e+00   6.251 4.09e-10 ***
## (Intercept):2  1.037e+01  1.192e+00   8.702  < 2e-16 ***
## (Intercept):3  1.326e+01  1.273e+00  10.418  < 2e-16 ***
## (Intercept):4  1.601e+01  1.397e+00  11.457  < 2e-16 ***
## perc.young    -1.042e-01  2.359e-02  -4.415 1.01e-05 ***
## hospital.beds -1.208e-04  4.825e-05  -2.504 0.012295 *
## perc.poor     -1.817e-01  3.228e-02  -5.627 1.83e-08 ***
## per.income    -9.041e-05  3.410e-05  -2.652 0.008012 **
## I(region)     -9.066e-01  1.108e-01  -8.181 2.81e-16 ***
## pop.density   -1.924e-04  5.830e-05  -3.300 0.000968 ***
## beds.per.1000 -2.982e-01  5.886e-02  -5.066 4.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4])
##
## Residual deviance: 834.2314 on 1749 degrees of freedom
```

```
##
## Log-likelihood: -417.1157 on 1749 degrees of freedom
##
## Number of iterations: 8
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##    perc.young hospital.beds     perc.poor    per.income     I(region)
##     0.9010829     0.9998792     0.8338865     0.9999096     0.4038827
##   pop.density beds.per.1000
##     0.9998077     0.7421837
```

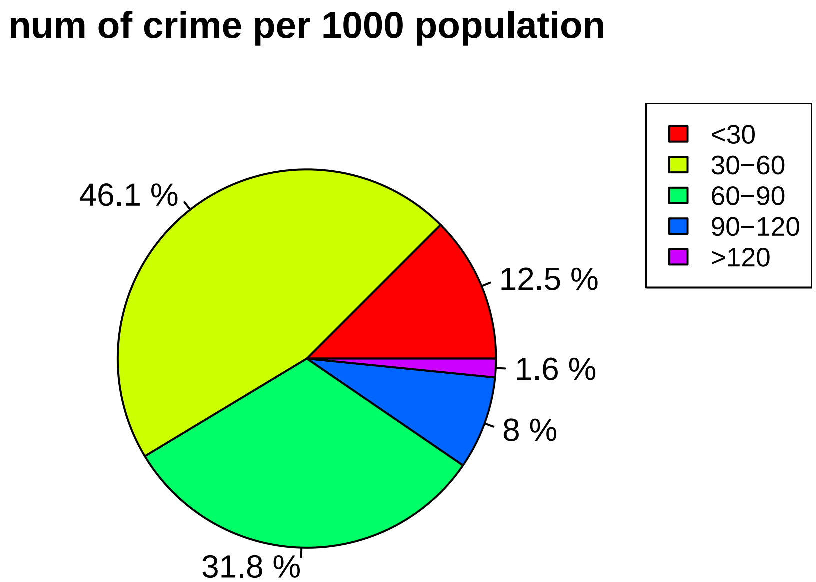
```
# logit.fit2 <- vglm(crime.rate.level ~ perc.young + hospital.beds + perc.poor + per.income +
                    # I(region) + pop.density + beds.per.1000, data =logi.data, family=cumulative)
# summary(fit2)
# pchisq(deviance(fit1)-deviance(fit), df=df.residual(fit1)-df.residual(fit),
# lower.tail=FALSE)


# step(logit.fit2,direction = "backward")$anova
```

Our main goal is to find the cause of high crime rate. Since crime rate is a quantitative variable, we simply divide this variable into 5 groups, which are `very low`(<30), `low`(30-60), `medium`(60-90), `high`(90-120) and `very high`(>120).

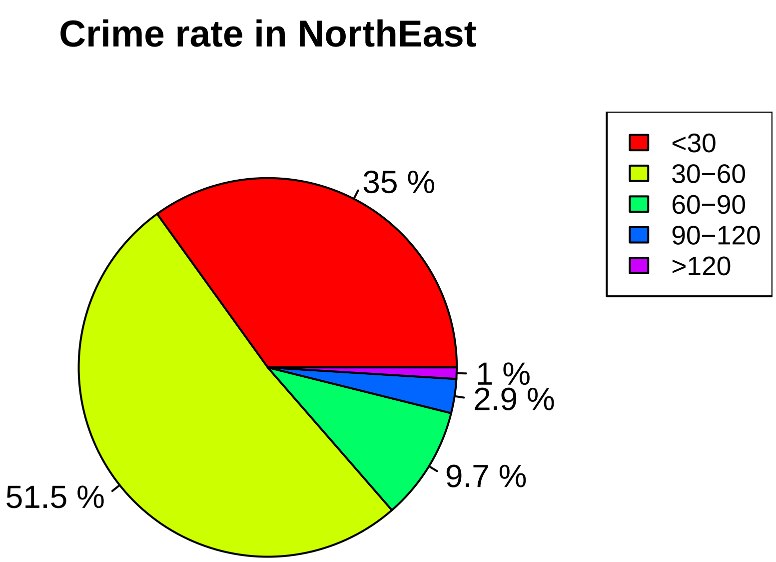
```
table(logi.data$crime.rate.level)
```

```
##
##   1   2   3   4   5
##  55 203 140  35   7
```

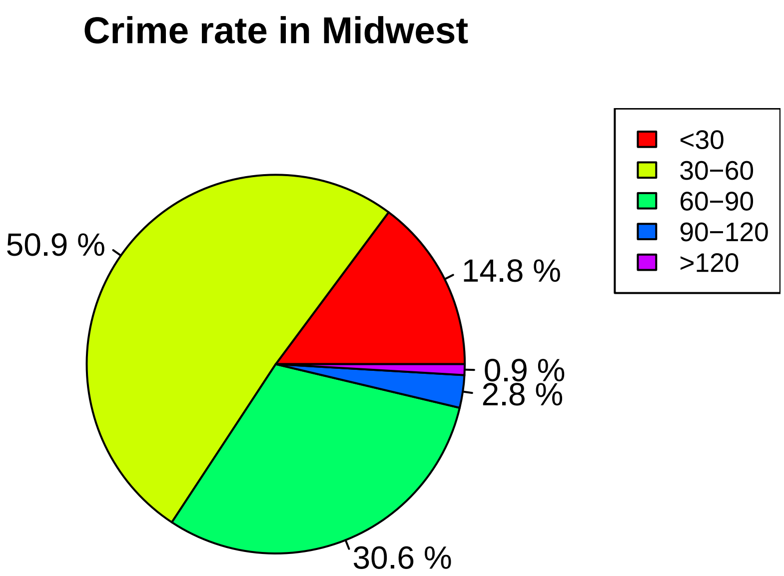
```
labels <- c("<30", "30-60", "60-90", "90-120",">120")
x = table(logi.data$crime.rate.level)
piepercent <- round(100*x/sum(x), 1)
pie(x,labels=paste(piepercent,"%"),main = "num of crime per 1000 population",col =rainbow(length(x)))
legend("topright", labels, cex = 0.8,
   fill = rainbow(length(x)))
```



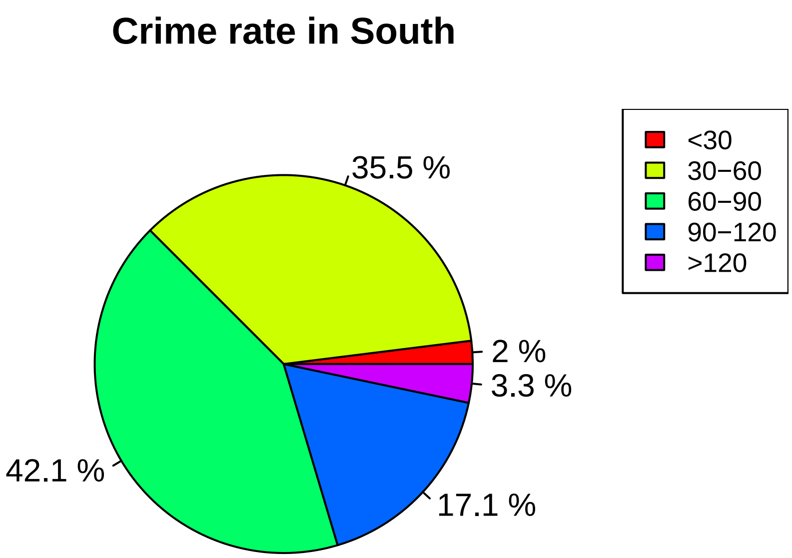

```r
ind.region1 <- which(logi.data$region == 1)
ind.region2 <- which(logi.data$region == 2)
ind.region3 <- which(logi.data$region == 3)
ind.region4 <- which(logi.data$region == 4)

x = table(logi.data$crime.rate.level[ind.region1])
piepercent <- round(100*x/sum(x), 1)
pie(x,labels=paste(piepercent,"%"),main = "Crime rate in NorthEast",col =rainbow(length(x)))
legend("topright", labels, cex = 0.8,
   fill = rainbow(length(x)))
```




```r
x = table(logi.data$crime.rate.level[ind.region2])
piepercent <- round(100*x/sum(x), 1)
pie(x,labels=paste(piepercent,"%"),main = "Crime rate in Midwest",col =rainbow(length(x)))
legend("topright", labels, cex = 0.8,
   fill = rainbow(length(x)))
```

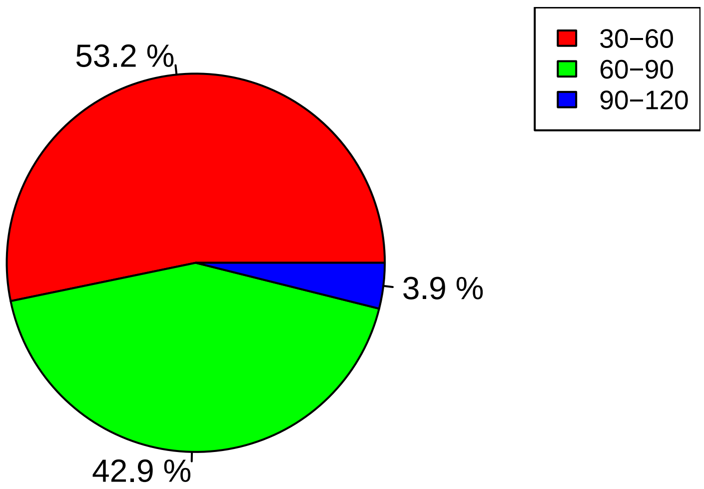



```r
x = table(logi.data$crime.rate.level[ind.region3])
piepercent <- round(100*x/sum(x), 1)
pie(x,labels=paste(piepercent,"%"),main = "Crime rate in South",col =rainbow(length(x)))
legend("topright", labels, cex = 0.8,
   fill = rainbow(length(x)))
```




```r
x = table(logi.data$crime.rate.level[ind.region4])
piepercent <- round(100*x/sum(x), 1)
pie(x,labels=paste(piepercent,"%"),main = "Crime rate in West",col =rainbow(length(x)))
legend("topright", labels[2:4], cex = 0.8,
   fill = rainbow(length(x)))
```

Crime rate in West



Then we fit a Cumulative logit model to find the factor that influence the crime rate

```r
names(logi.data)
```

```
##  [1] "id"                    "county"                "state"
##  [4] "area"                  "population"            "perc.young"
##  [7] "perc.old"              "physicians"            "hospital.beds"
## [10] "n.crimes"              "perc.hs"               "perc.bs"
## [13] "perc.poor"             "unemployment"          "per.income"
## [16] "tot.income"            "region"                "pop.density"
## [19] "physician.per.1000"    "beds.per.1000"         "crime.rate.per.1000"
## [22] "crime.rate.level"
```

```r
library(VGAM)
logit.fit1 <- vglm(crime.rate.level ~ perc.young + perc.old + perc.poor +
                      perc.hs  + per.income + physician.per.1000 + beds.per.1000 +
                    I(region) + pop.density , data =logi.data,
             family=cumulative(parallel=TRUE))
summary(logit.fit1)
```

```
##
## Call:
## vglm(formula = crime.rate.level ~ perc.young + perc.old + perc.poor +
##     perc.hs + per.income + physician.per.1000 + beds.per.1000 +
##     I(region) + pop.density, family = cumulative(parallel = TRUE),
##     data = logi.data)
##
##
## Pearson residuals:
##                    Min       1Q   Median       3Q   Max
## logit(P[Y<=1])  -1.146 -0.38151 -0.18565 -0.07911 6.340
## logit(P[Y<=2])  -3.270 -0.57761  0.18806  0.55117 3.098
## logit(P[Y<=3])  -7.818  0.05809  0.11653  0.26080 1.940
## logit(P[Y<=4]) -11.872  0.01865  0.03351  0.06506 1.474
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept):1       8.284e+00  2.276e+00   3.640 0.000273 ***
## (Intercept):2       1.157e+01  2.314e+00   5.000 5.73e-07 ***
## (Intercept):3       1.442e+01  2.353e+00   6.127 8.93e-10 ***
## (Intercept):4       1.720e+01  2.427e+00   7.086 1.38e-12 ***
## perc.young         -1.231e-01  3.253e-02  -3.784 0.000154 ***
## perc.old           -2.954e-03  3.365e-02  -0.088 0.930054
## perc.poor          -1.884e-01  3.895e-02  -4.836 1.32e-06 ***
## perc.hs             4.675e-03  2.253e-02   0.207 0.835637
## per.income         -1.375e-04  3.878e-05  -3.545 0.000392 ***
## physician.per.1000  1.752e-01  1.083e-01   1.618 0.105591
## beds.per.1000      -4.133e-01  8.421e-02  -4.908 9.21e-07 ***
## I(region)          -9.591e-01  1.171e-01  -8.189 2.62e-16 ***
## pop.density        -2.409e-04  6.251e-05  -3.854 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4])
##
## Residual deviance: 836.9297 on 1747 degrees of freedom
##
## Log-likelihood: -418.4649 on 1747 degrees of freedom
##
## Number of iterations: 6
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## 'pop.density'
##
## Exponentiated coefficients:
##         perc.young           perc.old          perc.poor
##          0.8841676          0.9970508          0.8282876
##            perc.hs         per.income physician.per.1000
##          1.0046858          0.9998625          1.1915029
##      beds.per.1000          I(region)        pop.density
##          0.6614838          0.3832314          0.9997591
```

```
names(logi.data)
```

```
##  [1] "id"                 "county"             "state"
##  [4] "area"               "population"         "perc.young"
##  [7] "perc.old"           "physicians"         "hospital.beds"
## [10] "n.crimes"           "perc.hs"            "perc.bs"
## [13] "perc.poor"          "unemployment"       "per.income"
## [16] "tot.income"         "region"             "pop.density"
## [19] "physician.per.1000" "beds.per.1000"      "crime.rate.per.1000"
## [22] "crime.rate.level"
```

```
library(MASS)
polr.fit1 <- polr(factor(crime.rate.level) ~ perc.young + perc.old + perc.poor +
                      perc.hs + perc.bs + per.income + physician.per.1000 + beds.per.1000 +
                    I(region) + pop.density , data =logi.data)
summary(polr.fit1)
```

```
##
## Re-fitting to get Hessian
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
## Call:
## polr(formula = factor(crime.rate.level) ~ perc.young + perc.old +
##     perc.poor + perc.hs + perc.bs + per.income + physician.per.1000 +
##     beds.per.1000 + I(region) + pop.density, data = logi.data)
##
## Coefficients:
##                         Value Std. Error t value
## perc.young          0.1308890  0.0290694  4.5026
## perc.old            0.0029908  0.0282110  0.1060
## perc.poor           0.1948914  0.0270043  7.2170
## perc.hs             0.0021503  0.0133114  0.1615
## perc.bs            -0.0137440  0.0193741 -0.7094
## per.income          0.0001527        NaN     NaN
## physician.per.1000 -0.1630729  0.0271350 -6.0097
## beds.per.1000       0.4060326  0.0412510  9.8430
## I(region)           0.9641693  0.0204153 47.2279
## pop.density         0.0002389  0.0001029  2.3220
##
## Intercepts:
##     Value      Std. Error t value
## 1|2     9.0950     0.0004 22048.4426
## 2|3    12.3854     0.1809    68.4482
## 3|4    15.2290     0.2742    55.5354
## 4|5    18.0136     0.2748    65.5508
##
## Residual Deviance: 836.7338
## AIC: 864.7338
```

AIC = 864.7338 `perc.old` `perc.hs` `perc.bs` and `physician.per.1000`

After fit the logistic model, I found the most of the predictor variables are significant. But the `perc.old` `perc.hs` `perc.bs` and `physician.per.1000` are not so important. So I decide to drop those 4 variables. In order to provide a solid evidence for dropping variables. I used AIC criterion. For the current full model, the AIC is 864.7338. Then I used backstep method to select variables.

```r
polr.fit2 <- polr(factor(crime.rate.level) ~ perc.young  + perc.poor +
                      per.income  + beds.per.1000 +
                    I(region) + pop.density , data =logi.data)
summary(polr.fit2)
```

```
##
## Re-fitting to get Hessian
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
## Call:
## polr(formula = factor(crime.rate.level) ~ perc.young + perc.poor +
##     per.income + beds.per.1000 + I(region) + pop.density, data = logi.data)
##
## Coefficients:
##                   Value Std. Error t value
## perc.young    0.1044856  6.617e-03  15.790
## perc.poor     0.1909416  2.327e-02   8.204
```

```
## per.income    0.0001064        NaN     NaN
## beds.per.1000 0.3214852  5.380e-02   5.975
## I(region)     0.9110400  2.120e-02  42.981
## pop.density   0.0002343  9.344e-05   2.507
##
## Intercepts:
##     Value     Std. Error t value
## 1|2    7.4567    0.0017  4287.9009
## 2|3   10.7168    0.1758    60.9650
## 3|4   13.5586    0.2673    50.7164
## 4|5   16.3800    0.2686    60.9722
##
## Residual Deviance: 839.7049
## AIC: 859.7049
```

```
logit.fit2 <- vglm(crime.rate.level ~ perc.young  + perc.poor +
                      per.income  + beds.per.1000 +
                    I(region) + pop.density , data =logi.data,
             family=cumulative(parallel=TRUE))
summary(logit.fit2)
```

```
##
## Call:
## vglm(formula = crime.rate.level ~ perc.young + perc.poor + per.income +
##     beds.per.1000 + I(region) + pop.density, family = cumulative(parallel = TRUE),
##     data = logi.data)
##
##
## Pearson residuals:
##                    Min       1Q   Median       3Q   Max
## logit(P[Y<=1])  -1.129 -0.38829 -0.19232 -0.07890 5.764
## logit(P[Y<=2])  -2.799 -0.59184  0.19488  0.55810 2.962
## logit(P[Y<=3])  -8.952  0.05866  0.11946  0.26048 2.015
## logit(P[Y<=4]) -12.119  0.01904  0.03261  0.06461 1.522
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  7.457e+00  1.127e+00   6.615 3.73e-11 ***
## (Intercept):2  1.072e+01  1.187e+00   9.032  < 2e-16 ***
## (Intercept):3  1.356e+01  1.267e+00  10.699  < 2e-16 ***
## (Intercept):4  1.638e+01  1.407e+00  11.640  < 2e-16 ***
## perc.young    -1.045e-01  2.355e-02  -4.436 9.14e-06 ***
## perc.poor     -1.909e-01  3.202e-02  -5.962 2.49e-09 ***
## per.income    -1.063e-04  3.358e-05  -3.167  0.00154 **
## beds.per.1000 -3.215e-01  5.825e-02  -5.519 3.41e-08 ***
## I(region)     -9.111e-01  1.106e-01  -8.237  < 2e-16 ***
## pop.density   -2.343e-04  6.010e-05  -3.898 9.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4])
##
```

```
## Residual deviance: 839.7049 on 1750 degrees of freedom
##
## Log-likelihood: -419.8525 on 1750 degrees of freedom
##
## Number of iterations: 6
##
## No Hauck-Donner effect found in any of the estimates
##
## Exponentiated coefficients:
##    perc.young     perc.poor    per.income beds.per.1000     I(region)
##     0.9007858     0.8261816     0.9998937     0.7250680     0.4020996
##   pop.density
##     0.9997658
```

```
logit.fit2@coefficients
```

```
## (Intercept):1 (Intercept):2 (Intercept):3 (Intercept):4    perc.young
##  7.4567869624 10.7169178197 13.5586713207 16.3801456018 -0.1044878324
##     perc.poor    per.income beds.per.1000     I(region)   pop.density
## -0.1909406643 -0.0001063533 -0.3214898559 -0.9110555465 -0.0002342573
```

```
exp(logit.fit2@coefficients)
```

```
## (Intercept):1 (Intercept):2 (Intercept):3 (Intercept):4    perc.young
##  1.731575e+03  4.511264e+04  7.734926e+05  1.299591e+07  9.007858e-01
##     perc.poor    per.income beds.per.1000     I(region)   pop.density
##  8.261816e-01  9.998937e-01  7.250680e-01  4.020996e-01  9.997658e-01
```

When we drop `perc.old` `perc.hs` `perc.bs` and `physician.per.1000`, the AIC now has been reduced to 859.7049, which is the least among all models. Therefore we can drop those four variables and get our final model.

Till now we have selected the most important variables `perc.young` `perc.poor` `per.income` `beds.per.1000` `pop.density` and `region`. The model now is

$$\log\left(\frac{P(Y\le j)}{1-P(Y\le j)}\right)=\beta_j-0.1045*perc.young-0.1909*perc.poor-1.063e-04*per.income-0.3215*beds.per-0.9111*I(regio$$

```
# stepAIC(logit.fit2,direction="backward",trace=FALSE)
# install.packages("usdm")
library(usdm)
```

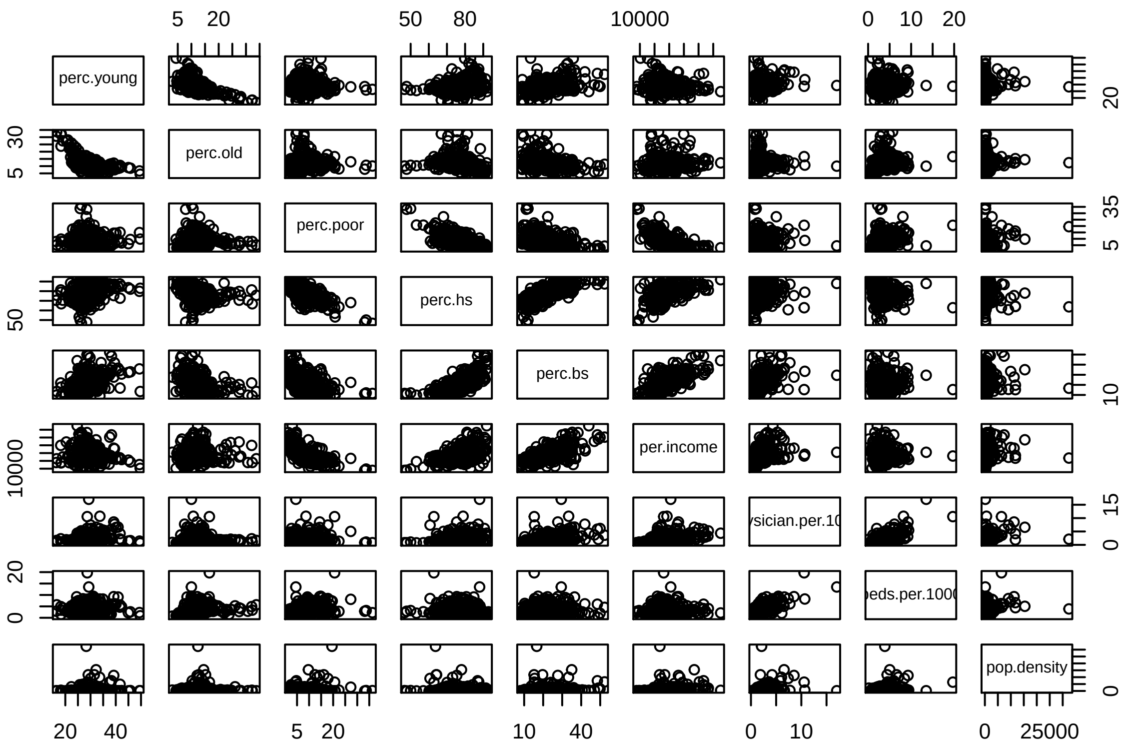
```
## Loading required package: sp

## Loading required package: raster

##
## Attaching package: 'raster'

## The following objects are masked from 'package:MASS':
##
##     area, select
```

```
vif.df = data.frame(perc.young = logi.data$perc.young, perc.old = logi.data$perc.old,
                    perc.poor = logi.data$perc.poor, perc.hs = logi.data$perc.hs,
                    perc.bs = logi.data$perc.bs, per.income = logi.data$per.income,
                    physician.per.1000 = logi.data$physician.per.1000,
                    beds.per.1000 = logi.data$beds.per.1000,
                    pop.density = logi.data$pop.density)
```

```r
vif(vif.df)
```

```
##            Variables      VIF
## 1         perc.young 2.545384
## 2           perc.old 1.941458
## 3          perc.poor 3.620321
## 4            perc.hs 3.991634
## 5            perc.bs 6.129869
## 6         per.income 4.521226
## 7 physician.per.1000 3.215007
## 8      beds.per.1000 2.817894
## 9        pop.density 1.298084
```

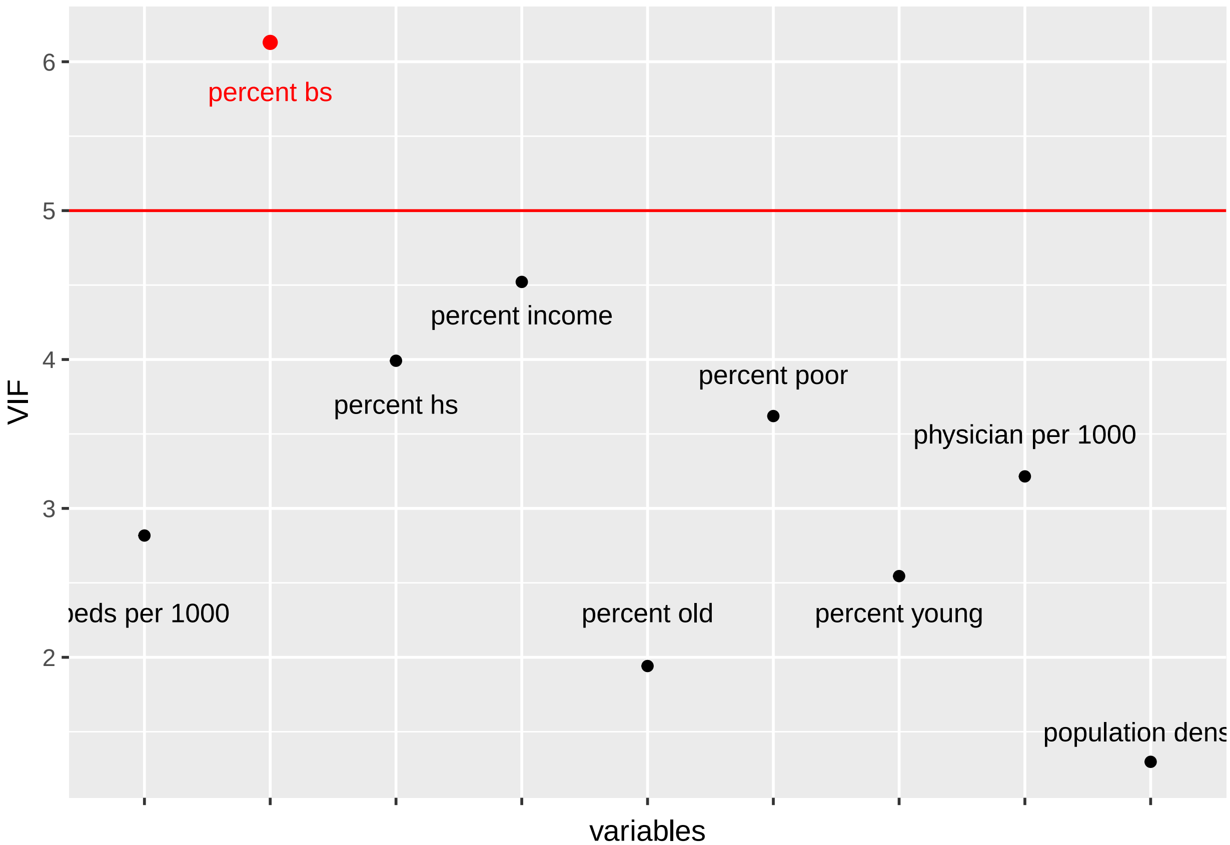
```r
# ggplot() + hist(vif(vif.df)$VIF)
pairs(vif.df)
```




```r
vif.num <- as.vector(vif(vif.df)$VIF)
vif.labels <- c('percent young','percent old','percent poor','percent hs','percent bs','percent income'
                'physician per 1000','beds per 1000','population density')
names(vif.num) <- vif.labels

ggplot() +
  geom_point(aes(x=vif.labels,y=vif.num) ) +
  theme(axis.text.x = element_blank()) +
  geom_text(aes(x = 'percent young',y=2.3, label='percent young'),size=3.5) +
  geom_text(aes(x = 'percent old',y=2.3, label='percent old'),size=3.5) +
  geom_text(aes(x = 'percent poor',y=3.9, label='percent poor'),size=3.5) +
  geom_text(aes(x = 'percent hs',y=3.7, label='percent hs'),size=3.5) +
  geom_text(aes(x = 'percent bs',y=5.8, label='percent bs'),size=3.5,col='red') +
  geom_text(aes(x = 'percent income',y=4.3, label='percent income'),size=3.5) +
```

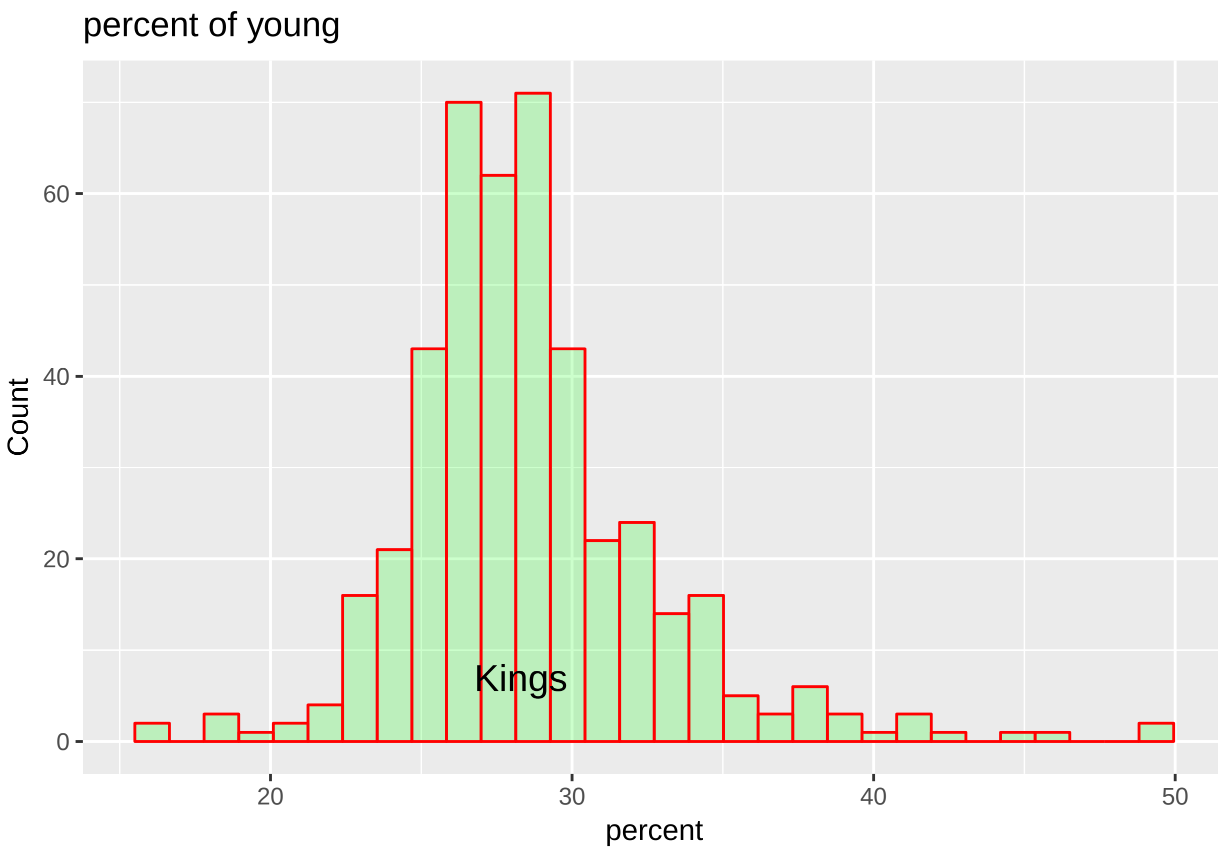
```
  geom_text(aes(x = 'physician per 1000',y=3.5, label='physician per 1000'),size=3.5) +
  geom_text(aes(x = 'beds per 1000',y=2.3, label='beds per 1000'),size=3.5) +
  geom_text(aes(x = 'population density',y=1.5, label='population density'),size=3.5) +
  geom_abline(intercept = 5, slope = 0,col='red') +
  xlab('variables') + ylab('VIF') +
  geom_point(aes(x = 'percent bs',y=6.129869, label='percent bs'),size=2,col='red')
```

```
## Warning: Ignoring unknown aesthetics: label
```



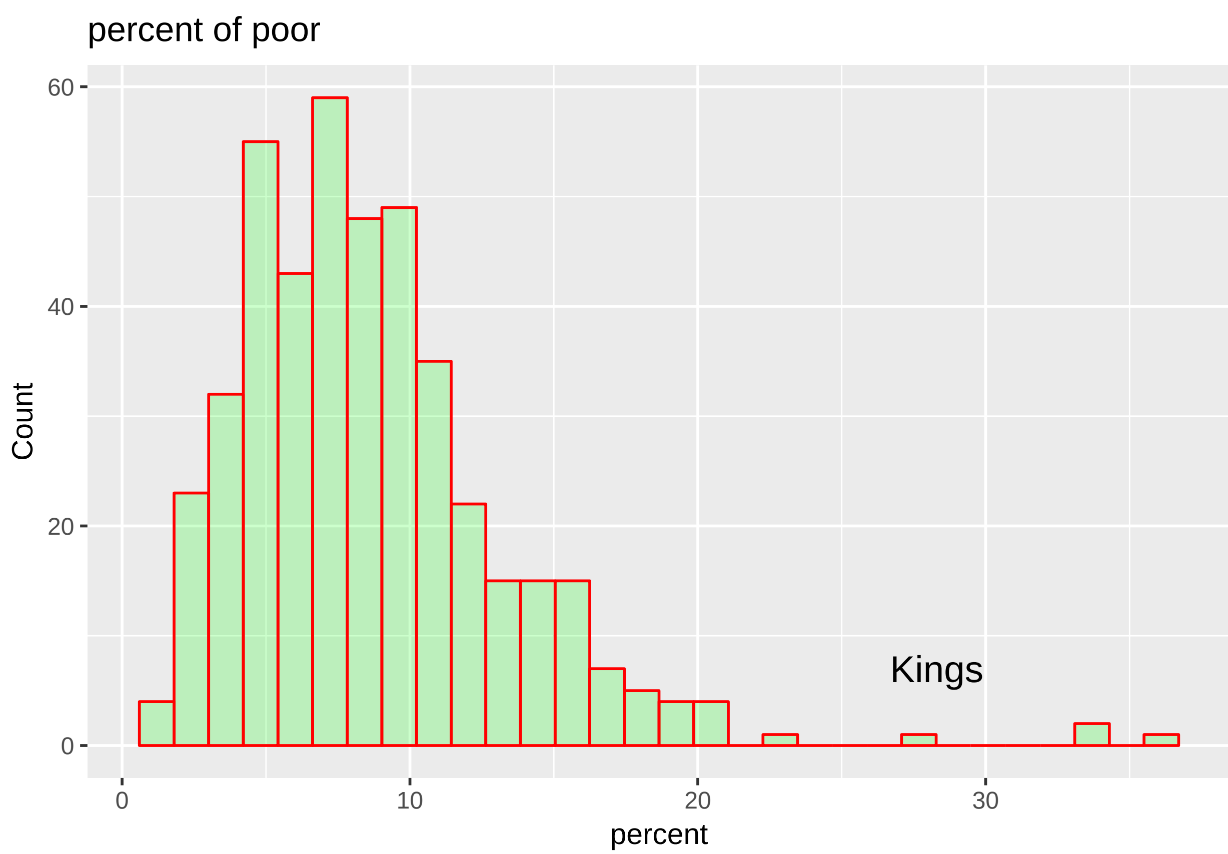

```
ggplot() +
  geom_histogram( aes(crime$perc.young),
                  col="red",
                  fill="green",
                  alpha = .2) +
  labs(title="percent of young") +
  labs(x="percent", y="Count") +
  geom_text(aes(x = crime$perc.young[6], y=7, label='Kings'),size=5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



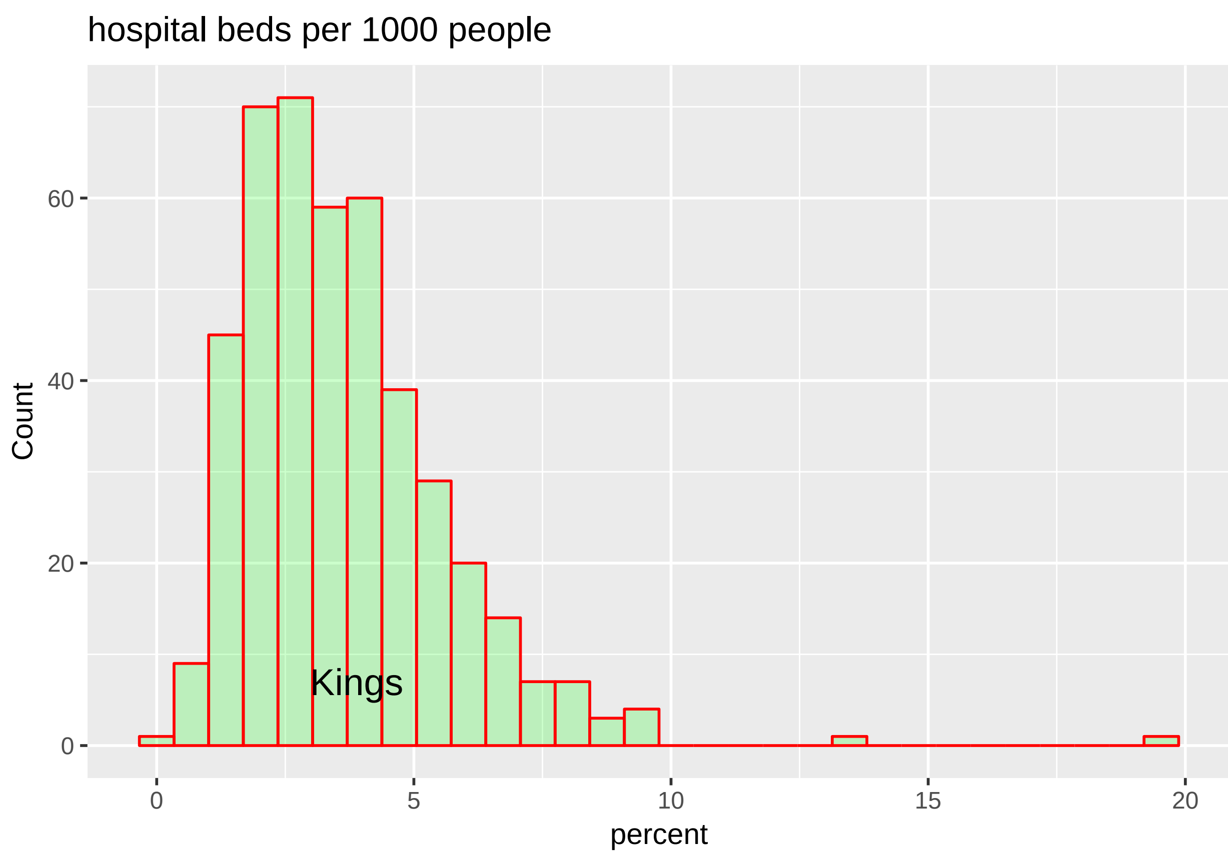

```
ggplot() +
  geom_histogram( aes(crime$perc.poor),
                  col="red",
                  fill="green",
                  alpha = .2) +
  labs(title="percent of poor") +
  labs(x="percent", y="Count") +
  geom_text(aes(x = crime$perc.young[6], y=7, label='Kings'),size=5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



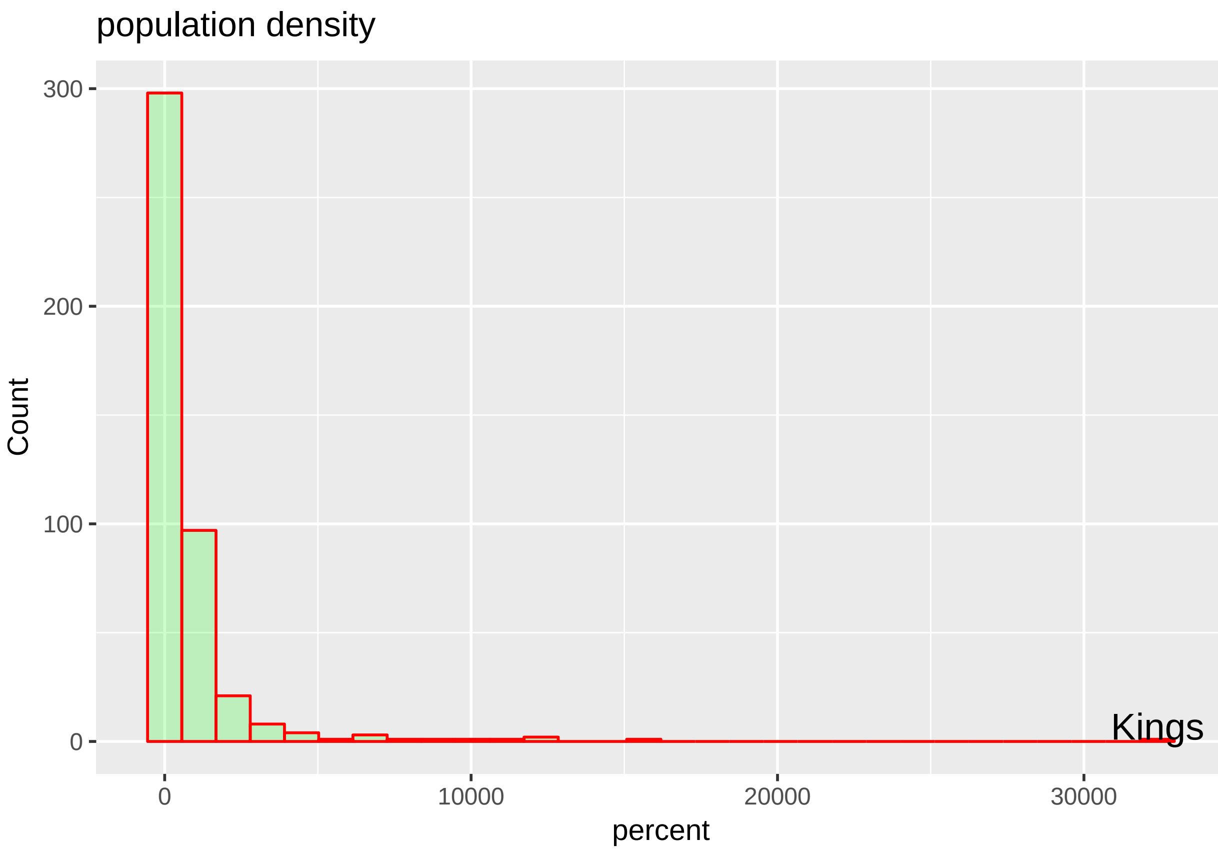

```r
ggplot() +
  geom_histogram( aes(crime$beds.per.1000),
                  col="red",
                  fill="green",
                  alpha = .2) +
  labs(title="hospital beds per 1000 people") +
  labs(x="percent", y="Count") +
  geom_text(aes(x = crime$beds.per.1000[6], y=7, label='Kings'),size=5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
ggplot() +
  geom_histogram( aes(crime$pop.density),
                  col="red",
                  fill="green",
                  alpha = .2) +
  labs(title="population density") +
  labs(x="percent", y="Count") +
  geom_text(aes(x = crime$pop.density[6], y=7, label='Kings'),size=5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


population density
Count
300
200
100
0
Kings
0
10000
20000
30000
percent