Final Project

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

```
library(readxl)

laws <- read_csv(file.path("data","waste_laws.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   Bill = col_character(),
##   Location_Person = col_character(),
##   Description = col_character(),
##   County = col_logical()
## )
```

```
tons <- read_csv(file.path("data","allYears.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   County = col_character(),
##   `1st Qtr.` = col_double(),
##   `2nd Qtr.` = col_double(),
##   `3rd Qtr.` = col_double(),
##   `4th Qtr.` = col_double(),
##   total = col_double(),
##   year = col_double()
## )
```

```
head(tons)
```

```
## # A tibble: 6 x 7
##   County       `1st Qtr.` `2nd Qtr.` `3rd Qtr.` `4th Qtr.`   total  year
##   <chr>             <dbl>      <dbl>      <dbl>      <dbl>   <dbl> <dbl>
## 1 Alameda          724440     805801     798844     733191 3062276  1990
## 2 Amador            11438      13853      15138      12522   52951  1990
## 3 Butte             34460      38909      35016      30720  139105  1990
## 4 Calaveras          7720       9596      10581       8914   36811  1990
## 5 Colusa             3550       4215       4337       4157   16259  1990
## 6 Contra Costa     142015     157067     138661     122657  560400  1990
```

```
sums <- tons %>% group_by(County) %>%
                  summarize(q1 = sum(`1st Qtr.`),
                            q2 = sum(`2nd Qtr.`),
                            q3 = sum(`3rd Qtr.`),
                            q4 = sum(`4th Qtr.`),
                            total = sum(total))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
fileName <- file.path("data","population.xlsx")
pop <- readxl::read_excel(fileName)
pop <- pop[pop$Year >= 1990,]
pop.total <- pop %>% group_by(Year, County) %>%
                    summarize(population = sum(Population))
```

```
## `summarise()` regrouping output by 'Year' (override with `.groups` argument)
```

```
test <- full_join(tons, pop.total, by= c("year"="Year", "County"="County"))

(num.laws <- laws %>% count(Year))
```

```
## # A tibble: 30 x 2
##     Year     n
##    <dbl> <int>
##  1  1990    27
##  2  1991    21
##  3  1992    29
##  4  1993    28
##  5  1994    20
##  6  1995    32
##  7  1996    17
##  8  1997     4
##  9  1998     5
## 10  1999    32
## # ... with 20 more rows
```

```
year.ton.totals <- tons %>% group_by(year) %>%
  summarize(tot = sum(total))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(year.ton.totals)
```

```
## # A tibble: 6 x 2
##    year      tot
##   <dbl>    <dbl>
## 1  1990 40107923
## 2  1991 36505782
## 3  1992 36057916
## 4  1993 34601584
## 5  1994 34391532
## 6  1995 33606562
```

```
year.pop.totals <- pop.total %>% group_by(Year) %>%
  summarize(tot = sum(population))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
popNumLaws <- data.frame(Year=year.pop.totals$Year,
                         Tons=year.ton.totals$tot[1:length(year.ton.totals$tot)-1],
                         Laws=num.laws$n[1:length(num.laws$n)-1],
                         Pop=year.pop.totals$tot)

popNumLaws <- popNumLaws %>% mutate(TonPer = Tons/Pop)
```

```
### tons includes year, Q1-4 tons, and total tons for each county
### 1990-2019
tons <- read_csv(file.path("data","allYears.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   County = col_character(),
##   `1st Qtr.` = col_double(),
##   `2nd Qtr.` = col_double(),
##   `3rd Qtr.` = col_double(),
##   `4th Qtr.` = col_double(),
##   total = col_double(),
##   year = col_double()
## )
```

```
### statewide.tons includes year, total tonnage
statewide.tons <- tons %>% group_by(year) %>%
  summarize(total_tons = sum(total))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
### counties.tons includes year, County name, total tonnage
counties.tons <- tons %>% select(year, County, total_tons=total)
```

```
# Populations for each county for 1990-2018
populations <- read_csv(file.path("data","population.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   County = col_character(),
##   population = col_double()
## )
```

```
# Populations statewide for 1990-2018
statewide.pop <- populations %>% group_by(Year) %>%
  summarize(total_pop = sum(population))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
laws <- read_csv(file.path("data","waste_laws_categories.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   Bill = col_character(),
##   Rep = col_character(),
##   Description = col_character(),
##   `Household?` = col_character(),
##   `Industry?` = col_character(),
##   `IDK?` = col_character()
## )
```

```
#laws$Year <- as.factor(laws$Year)
# num.laws is count of laws made each year
num.laws <- laws %>% count(Year)
# add row with num laws in previous year
num.laws$lag <- lag(num.laws$n)
# add row with cumulative laws before that year
num.laws$cumul <- cumsum(num.laws$n)
# get number of industry laws
#num.laws$household_n <-
temp <- laws %>% group_by(Year) %>% count(`Household?`, .drop=FALSE) %>% filter(`Household?` ==
'X')
temp <- temp %>% select(Year, household_n = n)
num.laws <- full_join(num.laws, temp, by='Year')
num.laws$hh_lag <- lag(num.laws$household_n)
#
# get number of household laws
#num.laws$industry_n <-
col <- laws %>% group_by(Year) %>% filter(`Industry?` == 'X') %>% count(`Industry?`) %>% ungroup
()
num.laws <- num.laws %>% add_column(industry_n = col$n)
num.laws$ind_lag <- lag(num.laws$industry_n)
num.laws$ind_cumul <- cumsum(num.laws$industry_n)
num.laws <- num.laws %>% mutate_all(funs(replace_na(., 0)))
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
num.laws$hh_cumul <- cumsum(num.laws$household_n)

num.laws
```

```
## # A tibble: 30 x 10
##     Year     n   lag cumul household_n hh_lag industry_n ind_lag ind_cumul
##    <dbl> <dbl> <dbl> <dbl>       <dbl>  <dbl>      <dbl>   <dbl>     <dbl>
##  1  1990    27     0    27           5      0         13       0        13
##  2  1991    21    27    48           4      5         16      13        29
##  3  1992    29    21    77           4      4         21      16        50
##  4  1993    28    29   105           2      4         23      21        73
##  5  1994    20    28   125           3      2         16      23        89
##  6  1995    32    20   157           6      3         28      16       117
##  7  1996    17    32   174           2      6         15      28       132
##  8  1997     4    17   178           0      2          4      15       136
##  9  1998     5     4   183           0      0          5       4       141
## 10  1999    32     5   215           1      0         24       5       165
## # ... with 20 more rows, and 1 more variable: hh_cumul <dbl>
```

```
dat.industry <- read_csv(file.path("data","download.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   GeoFips = col_character(),
##   GeoName = col_character(),
##   Description = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
gdps <- dat.industry[dat.industry$LineCode == 1,]
gdps <- gdps %>% separate(GeoName, c("County","State"), sep=",")
fixed.gdps <- gdps %>% pivot_longer(cols=starts_with("20"), names_to="Year", values_to="GDP")
fixed.gdps$Year <- as.integer(fixed.gdps$Year)
county.gdp <- fixed.gdps %>% select(c('County','Year','GDP'))
```

```
statewide.gdp <- read_csv(file.path("data","gdp_statewide_byindustry.csv"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   `Series ID` = col_character(),
##   `Region Name` = col_character(),
##   `Region Code` = col_character(),
##   Description = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
statewide.gdp <- statewide.gdp %>% select(-c('Series ID','Region Name','Region Code'))
#(state.gdp <- statewide.gdp %>% pivot_longer(cols=everything(), names_to = 'Year'))
#state.gdp$Year <- as.integer(state.gdp$Year)
state.gdp <- as.data.frame(t(as.matrix(statewide.gdp)))
names(state.gdp) <- as.matrix(state.gdp[1, ])
state.gdp <- state.gdp[-1, ]
state.gdp[] <- lapply(state.gdp, function(x) type.convert(as.character(x)))
state.gdp <- cbind('Year'=rownames(state.gdp), data.frame(state.gdp, row.names=NULL))
state.gdp$Year <- as.integer(state.gdp$Year)
```

```
all_data <- inner_join(statewide.tons, statewide.pop, by=c("year"="Year"))
all_data <- inner_join(all_data, state.gdp, by=c("year"="Year"))
all_data <- inner_join(all_data, num.laws, by=c("year"="Year"))
all_data
```

```
## # A tibble: 22 x 25
##     year total_tons total_pop Total_GDP Resources_Mining Private_Goods
##    <dbl>      <dbl>     <dbl>     <dbl>            <dbl>         <dbl>
##  1  1997   33694292  32207869  1071327.           24875.       216644.
##  2  1998   35555133  32657877  1147944.           21252.       226783.
##  3  1999   35508312  33140771  1247734.           22300.       248947.
##  4  2000   36954946  33739265  1366166.           25165.       285606.
##  5  2001   38128084  34274507  1387552.           24135.       260632.
##  6  2002   37829174  34743110  1439342.           24478.       246662.
##  7  2003   40275549  35181056  1530560.           28098.       271918.
##  8  2004   41158946  35588110  1632429.           34376.       299273.
##  9  2005   42158298  35886171  1752649.           37951        338925.
## 10  2006   41632572  36132925  1874851.           41993.       362382.
## # ... with 12 more rows, and 19 more variables: Ag_For_Fish_Hunt <dbl>,
## #   Art_Ent_Food <dbl>, Construction <dbl>, Ed_Health_Soci <dbl>,
## #   Fin_Ensu_Estate <dbl>, Information <dbl>, Manufacturing <dbl>,
## #   Private_Mining <dbl>, Nondurable_Manu <dbl>, Trans_Ware <dbl>, n <dbl>,
## #   lag <dbl>, cumul <dbl>, household_n <dbl>, hh_lag <dbl>, industry_n <dbl>,
## #   ind_lag <dbl>, ind_cumul <dbl>, hh_cumul <dbl>
```

```
c_data <- inner_join(county.gdp, populations, by=c('County'='County', 'Year'='Year'))
c_data <- inner_join(c_data, counties.tons, by=c('Year'='year','County'='County'))
c_data <- inner_join(c_data, num.laws, by=c('Year'='Year'))
c_data
```

```
## # A tibble: 925 x 14
##    County  Year    GDP population total_tons     n   lag cumul household_n
##    <chr>  <dbl>  <dbl>      <dbl>      <dbl> <dbl> <dbl> <dbl>       <dbl>
##  1 Alame~  2001 8.09e7    1457185    2152028    17    13   245           5
##  2 Alame~  2002 8.23e7    1467063    2035572    25    17   270           1
##  3 Alame~  2003 8.64e7    1467892    2000026    13    25   283           1
##  4 Alame~  2004 8.84e7    1466407    2055950    13    13   296           2
##  5 Alame~  2005 9.07e7    1462736    2064312    15    13   311           0
##  6 Alame~  2006 9.47e7    1462371    2218252    34    15   345           6
##  7 Alame~  2007 9.54e7    1470622    2090272    10    34   355           2
##  8 Alame~  2008 9.53e7    1484085    1790756    13    10   368           1
##  9 Alame~  2009 9.02e7    1497799    1558254     6    13   374           1
## 10 Alame~  2010 9.30e7    1509240    1463801    10     6   384           0
## # ... with 915 more rows, and 5 more variables: hh_lag <dbl>, industry_n <dbl>,
## #   ind_lag <dbl>, ind_cumul <dbl>, hh_cumul <dbl>
```

```
str(all_data)
```

```
## tibble [22 x 25] (S3: tbl_df/tbl/data.frame)
##  $ year            : num [1:22] 1997 1998 1999 2000 2001 ...
##  $ total_tons      : num [1:22] 33694292 35555133 35508312 36954946 38128084 ...
##  $ total_pop       : num [1:22] 32207869 32657877 33140771 33739265 34274507 ...
##  $ Total_GDP       : num [1:22] 1071327 1147944 1247734 1366167 1387552 ...
##  $ Resources_Mining: num [1:22] 24875 21253 22300 25165 24135 ...
##  $ Private_Goods   : num [1:22] 216644 226783 248947 285606 260632 ...
##  $ Ag_For_Fish_Hunt: num [1:22] 18233 16787 17459 18018 17707 ...
##  $ Art_Ent_Food    : num [1:22] 40733 44375 48558 53642 55427 ...
##  $ Construction    : num [1:22] 35706 42298 47722 54562 58705 ...
##  $ Ed_Health_Soci  : num [1:22] 61632 64467 68541 73227 78912 ...
##  $ Fin_Ensu_Estate : num [1:22] 210000 225654 243258 259482 281761 ...
##  $ Information     : num [1:22] 64379 72684 88650 84930 92900 ...
##  $ Manufacturing   : num [1:22] 156063 163232 178925 205879 177792 ...
##  $ Private_Mining  : num [1:22] 6642 4465 4841 7146 6428 ...
##  $ Nondurable_Manu : num [1:22] 50077 53181 56613 62628 64824 ...
##  $ Trans_Ware      : num [1:22] 28965 30629 31208 33725 33588 ...
##  $ n               : num [1:22] 4 5 32 13 17 25 13 13 15 34 ...
##  $ lag             : num [1:22] 17 4 5 32 13 17 25 13 13 15 ...
##  $ cumul           : num [1:22] 178 183 215 228 245 270 283 296 311 345 ...
##  $ household_n     : num [1:22] 0 0 1 0 5 1 1 2 0 6 ...
##  $ hh_lag          : num [1:22] 2 0 0 1 0 5 1 1 2 0 ...
##  $ industry_n      : num [1:22] 4 5 24 11 12 19 11 13 15 30 ...
##  $ ind_lag         : num [1:22] 15 4 5 24 11 12 19 11 13 15 ...
##  $ ind_cumul       : num [1:22] 136 141 165 176 188 207 218 231 246 276 ...
##  $ hh_cumul        : num [1:22] 26 26 27 27 32 33 34 36 36 42 ...
```

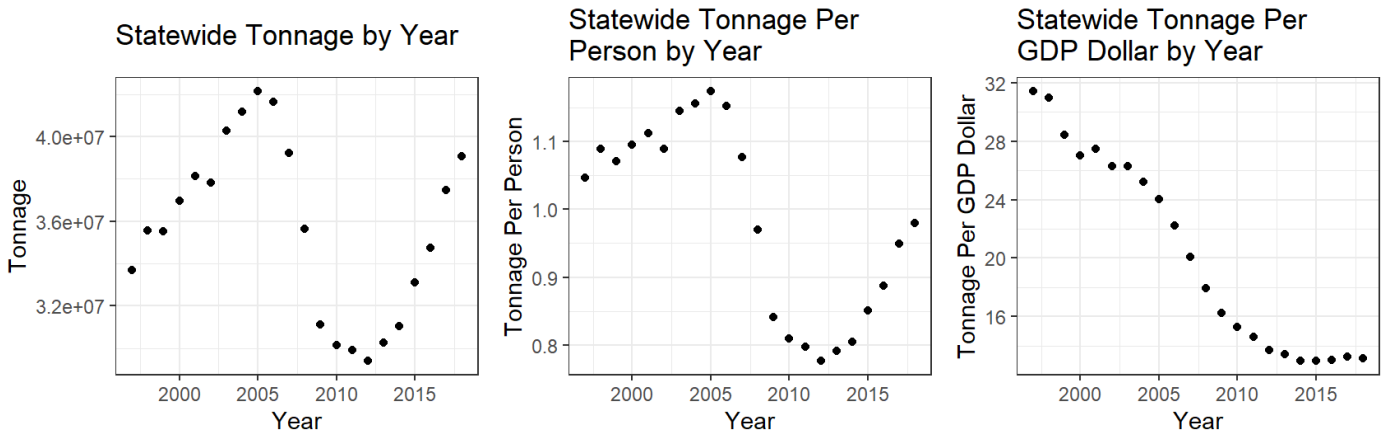
```
model1 <- lm(total_tons ~ .-year -Fin_Ensu_Estate -Information -Resources_Mining -ind_lag -Ed_He
alth_Soci -ind_cumul -hh_cumul -Nondurable_Manu -Total_GDP -total_pop, data=all_data)
summary(model1)
```

```
##
## Call:
## lm(formula = total_tons ~ . - year - Fin_Ensu_Estate - Information -
##     Resources_Mining - ind_lag - Ed_Health_Soci - ind_cumul -
##     hh_cumul - Nondurable_Manu - Total_GDP - total_pop, data = all_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -413374  -94468  -20708  129408  305803
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.484e+07  7.962e+05  56.310 1.10e-11 ***
## Private_Goods    -1.145e+07  2.815e+06  -4.068 0.003595 **
## Ag_For_Fish_Hunt  1.145e+07  2.815e+06   4.068 0.003595 **
## Art_Ent_Food      5.859e+02  6.181e+01   9.480 1.26e-05 ***
## Construction      1.145e+07  2.815e+06   4.068 0.003595 **
## Manufacturing     1.145e+07  2.815e+06   4.068 0.003595 **
## Private_Mining    1.145e+07  2.815e+06   4.068 0.003595 **
## Trans_Ware       -3.416e+02  6.175e+01  -5.532 0.000553 ***
## n                 2.574e+05  4.732e+04   5.440 0.000616 ***
## lag               6.858e+04  1.222e+04   5.612 0.000503 ***
## cumul            -1.150e+05  1.008e+04 -11.414 3.14e-06 ***
## household_n      -2.462e+05  6.210e+04  -3.964 0.004155 **
## hh_lag           -3.726e+05  6.267e+04  -5.945 0.000344 ***
## industry_n       -2.978e+05  5.951e+04  -5.005 0.001047 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 269900 on 8 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9957
## F-statistic: 373.5 on 13 and 8 DF,  p-value: 1.192e-09
```

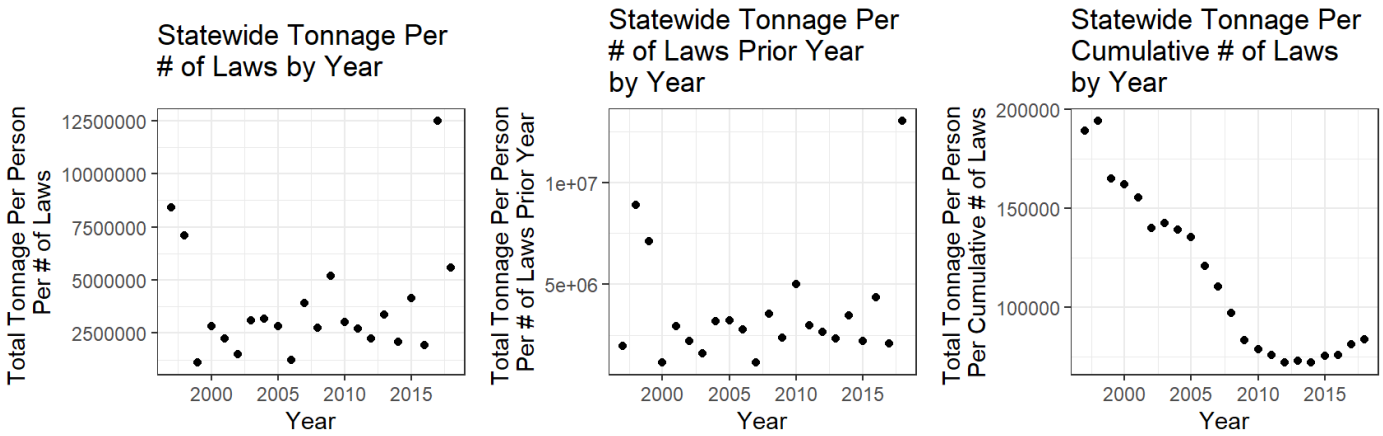
```
source("https://cipolli.com/students/code/plotResiduals.R")
#plotResiduals(model1)
```

```
#plotResiduals(model1)
```

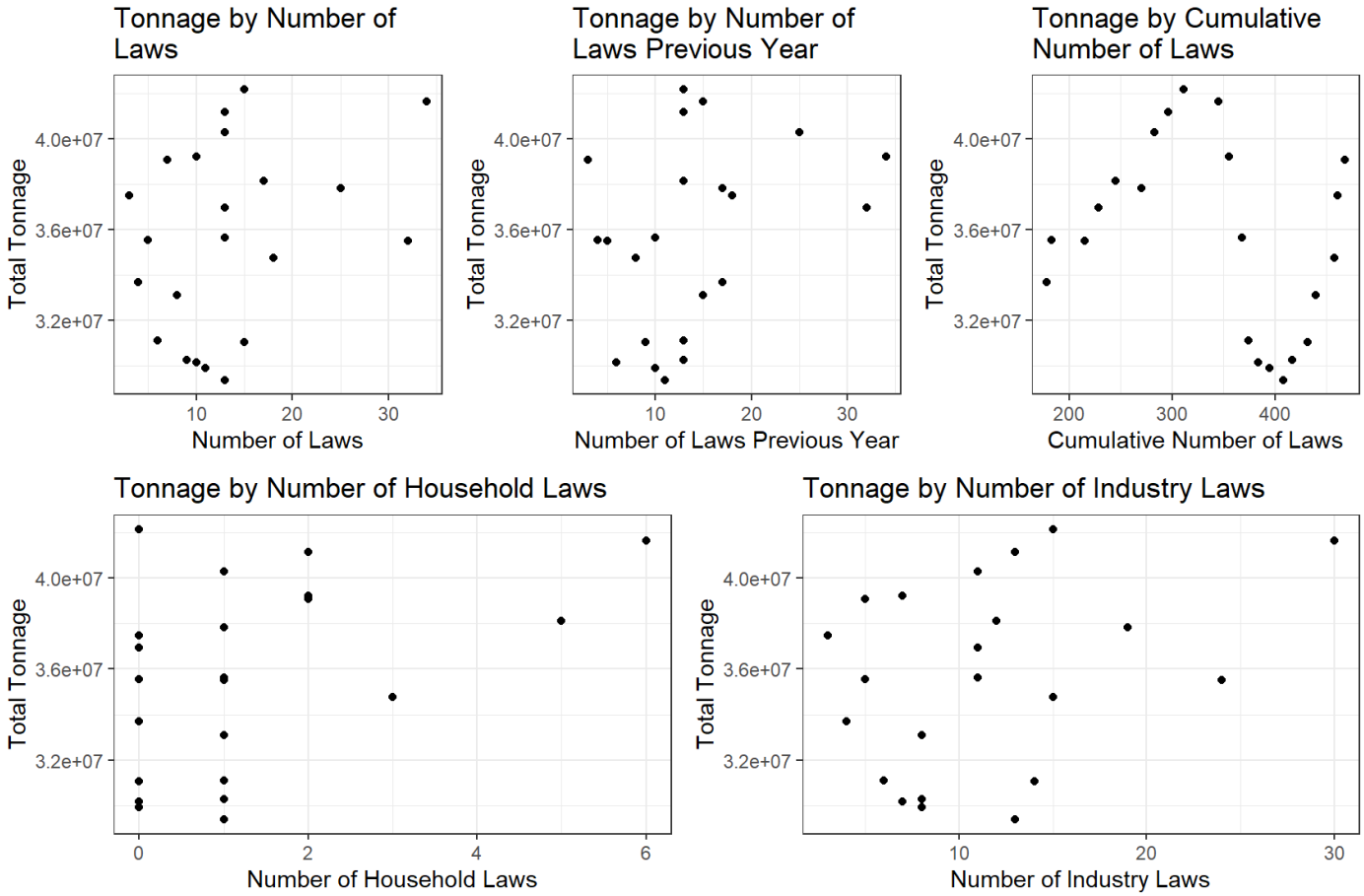
```
library(patchwork)
tonsvsyears <- ggplot(all_data, aes(year, total_tons)) +
  geom_point() +
  ggtitle("Statewide Tonnage by Year") +
  xlab("Year") +
  ylab("Tonnage") +
  theme_bw()

tonsppvsyears <- ggplot(all_data, aes(year, total_tons/total_pop)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \nPerson by Year") +
  xlab("Year") +
  ylab("Tonnage Per Person") +
  theme_bw()

tonspgdpvsyears <- ggplot(all_data, aes(year, total_tons/Total_GDP)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \nGDP Dollar by Year") +
  xlab("Year") +
  ylab("Tonnage Per GDP Dollar") +
  theme_bw()

tonsvsyears + tonsppvsyears + tonspgdpvsyears
```



```
tonspcumulvsyears <- ggplot(all_data, aes(year, total_tons/cumul)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \nCumulative # of Laws \nby Year") +
  xlab("Year") +
  ylab("Total Tonnage Per Person \nPer Cumulative # of Laws") +
  theme_bw()

tonspnlawsvsyears <- ggplot(all_data, aes(year, total_tons/n)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \n# of Laws by Year") +
  xlab("Year") +
  ylab("Total Tonnage Per Person \nPer # of Laws") +
  theme_bw()

tonsplagvsyears <- ggplot(all_data, aes(year, total_tons/lag)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \n# of Laws Prior Year \nby Year") +
  xlab("Year") +
  ylab("Total Tonnage Per Person \nPer # of Laws Prior Year") +
  theme_bw()

tonspindnvsyears <- ggplot(all_data, aes(year, total_tons/industry_n)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \n# of Industry Laws \nby Year") +
  xlab("Year") +
  ylab("Total Tonnage Per Person \nPer # of Industry Laws") +
  theme_bw()

tonsphhnvsyears <- ggplot(all_data, aes(year, total_tons/household_n)) +
  geom_point() +
  ggtitle("Statewide Tonnage Per \n# of Household Laws \nby Year") +
  xlab("Year") +
  ylab("Total Tonnage Per Person \nPer # of Household Laws") +
  theme_bw()

#(tonspcumulvsyears + tonspnlawsvsyears + tonsplagvsyears)/(tonspindnvsyears + tonsphhnvsyears)
tonspnlawsvsyears + tonsplagvsyears + tonspcumulvsyears
```

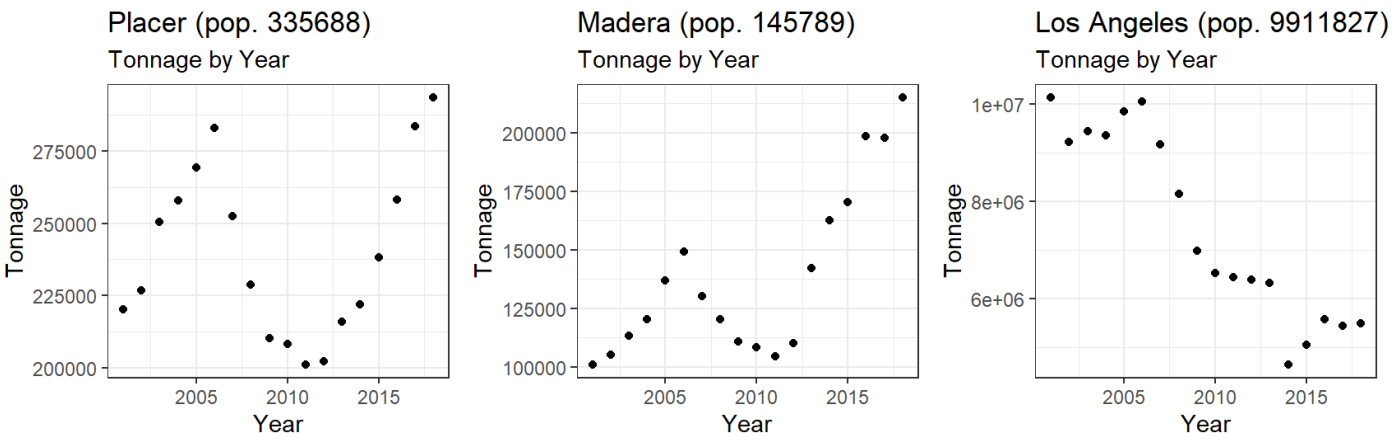


```
tonsvslaws <- ggplot(all_data, aes(n, total_tons)) +
  geom_point() +
  ggtitle("Tonnage by Number of \nLaws") +
  xlab("Number of Laws") +
  ylab("Total Tonnage") +
  theme_bw()

tonsvslaglaws <- ggplot(all_data, aes(lag, total_tons)) +
  geom_point() +
  ggtitle("Tonnage by Number of \nLaws Previous Year") +
  xlab("Number of Laws Previous Year") +
  ylab("Total Tonnage") +
  theme_bw()

tonsvscumullaws <- ggplot(all_data, aes(cumul, total_tons)) +
  geom_point() +
  ggtitle("Tonnage by Cumulative \nNumber of Laws") +
  xlab("Cumulative Number of Laws") +
  ylab("Total Tonnage") +
  theme_bw()

tonsvshhlaws <- ggplot(all_data, aes(household_n, total_tons)) +
  geom_point() +
  ggtitle("Tonnage by Number of Household Laws") +
  xlab("Number of Household Laws") +
  ylab("Total Tonnage") +
  theme_bw()

tonsvsindlaws <- ggplot(all_data, aes(industry_n, total_tons)) +
  geom_point() +
  ggtitle("Tonnage by Number of Industry Laws") +
  xlab("Number of Industry Laws") +
  ylab("Total Tonnage") +
  theme_bw()

(tonsvslaws + tonsvslaglaws + tonsvscumullaws)/(tonsvshhlaws + tonsvsindlaws)
```

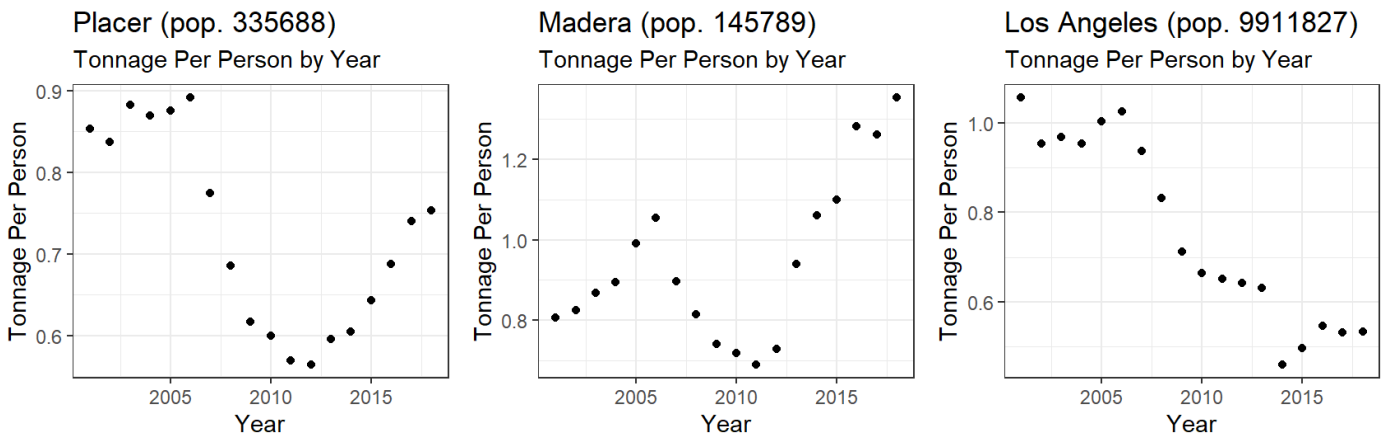



```
placer <- c_data[c_data$County=='Placer',]
madera <- c_data[c_data$County=='Madera',]
la <- c_data[c_data$County=='Los Angeles',]
```

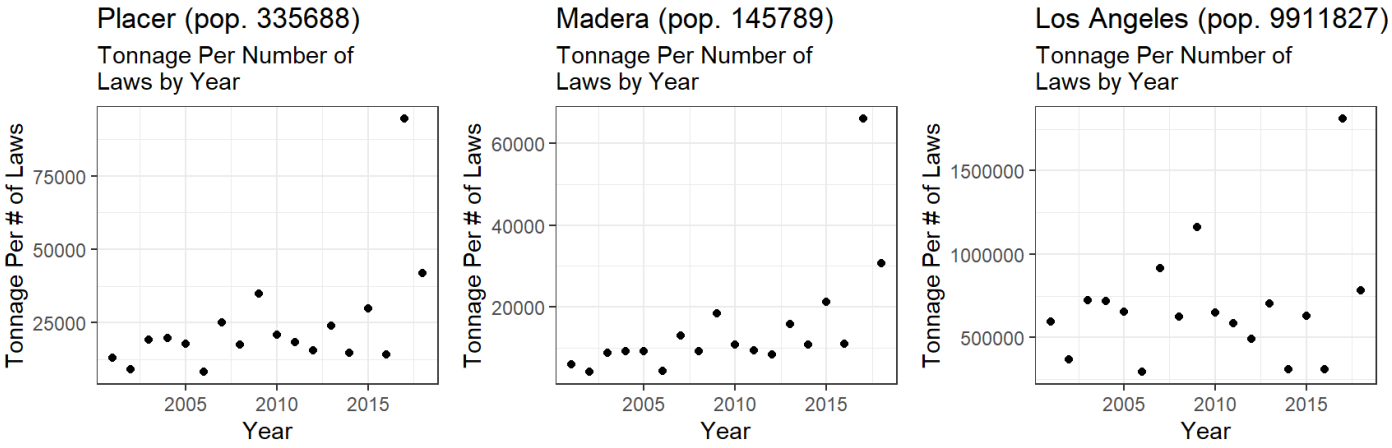
```
ggplacer <- ggplot(placer, aes(Year, total_tons)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="Tonnage by Yea
r") +
  xlab("Year") +
  ylab("Tonnage") +
  theme_bw()

ggmad <- ggplot(madera, aes(Year, total_tons)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="Tonnage by Yea
r") +
  xlab("Year") +
  ylab("Tonnage") +
  theme_bw()

ggla <- ggplot(la, aes(Year, total_tons)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="Tonnage by Ye
ar") +
  xlab("Year") +
  ylab("Tonnage") +
  theme_bw()

(ggplacer + ggmad + ggla)
```

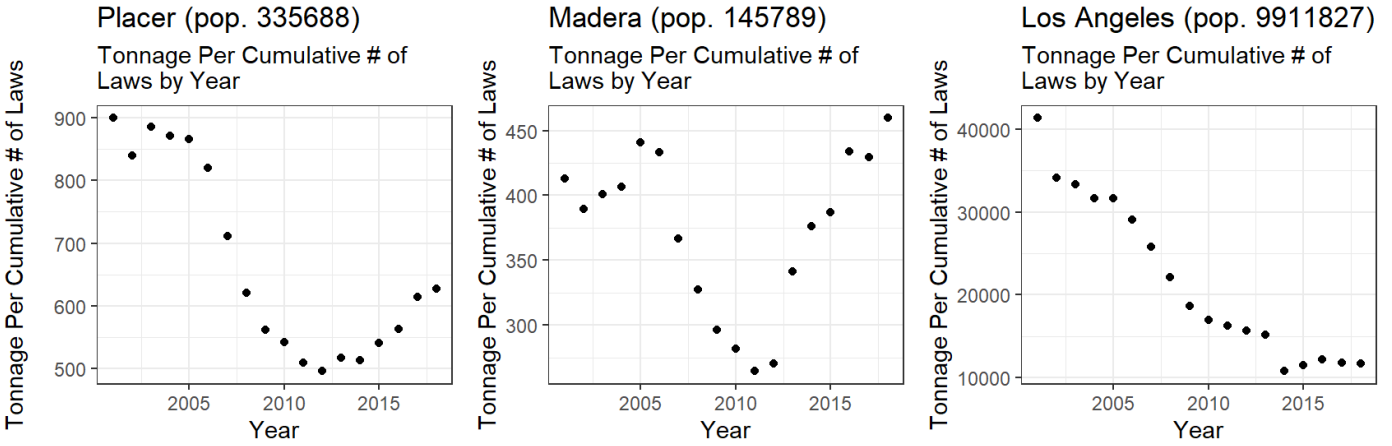


```
ggperplacer <- ggplot(placer, aes(Year, total_tons/population)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="Tonnage Per Pe
rson by Year") +
  xlab("Year") +
  ylab("Tonnage Per Person") +
  theme_bw()

ggpermad <- ggplot(madera, aes(Year, total_tons/population)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="Tonnage Per Pe
rson by Year") +
  xlab("Year") +
  ylab("Tonnage Per Person") +
  theme_bw()

ggperla <- ggplot(la, aes(Year, total_tons/population)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="Tonnage Per P
erson by Year") +
  xlab("Year") +
  ylab("Tonnage Per Person") +
  theme_bw()

(ggperplacer + ggpermad + ggperla)
```

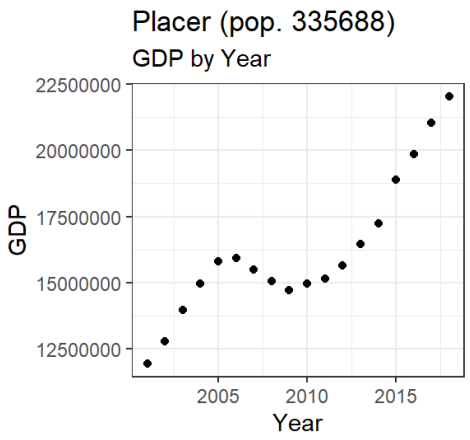


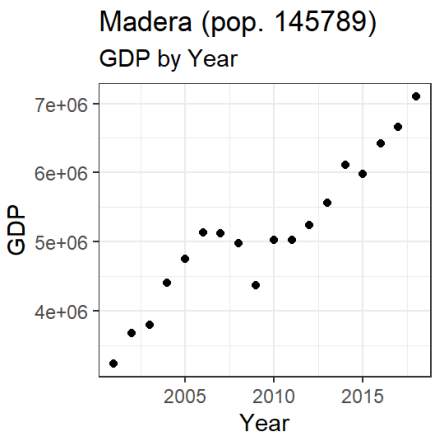
```
ggplacern <- ggplot(placer, aes(Year, total_tons/n)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="Tonnage Per Nu
mber of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per # of Laws") +
  theme_bw()

ggplacercumul <- ggplot(placer, aes(Year, total_tons/cumul)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="Tonnage Per Cu
mulative # of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per Cumulative # of Laws") +
  theme_bw()

ggmadn <- ggplot(madera, aes(Year, total_tons/n)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="Tonnage Per Nu
mber of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per # of Laws") +
  theme_bw()

ggmadcumul <- ggplot(madera, aes(Year, total_tons/cumul)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="Tonnage Per Cu
mulative # of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per Cumulative # of Laws") +
  theme_bw()

gglan <- ggplot(la, aes(Year, total_tons/n)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="Tonnage Per N
umber of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per # of Laws") +
  theme_bw()

gglacumul <- ggplot(la, aes(Year, total_tons/cumul)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="Tonnage Per C
umulative # of \nLaws by Year") +
  xlab("Year") +
  ylab("Tonnage Per Cumulative # of Laws") +
  theme_bw()

(ggplacern + ggmadn + gglan)/ (ggplacercumul + ggmadcumul + gglacumul)
```

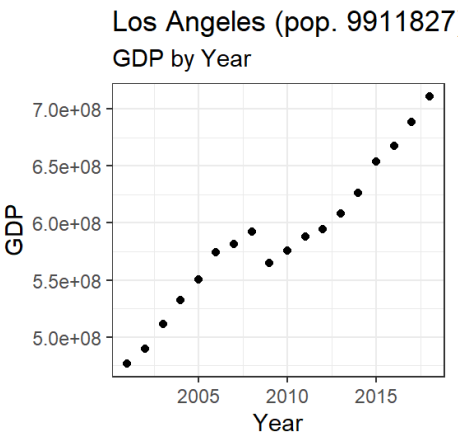




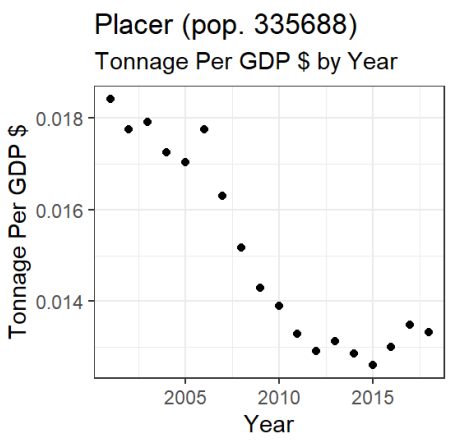

```
gggdpplacer <- ggplot(placer, aes(Year, GDP)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="GDP by Year")
 +
  xlab("Year") +
  ylab("GDP") +
  theme_bw()

gggdpmad <- ggplot(madera, aes(Year, GDP)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="GDP by Year")
 +
  xlab("Year") +
  ylab("GDP") +
  theme_bw()

gggdpla <- ggplot(la, aes(Year, GDP)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="GDP by Year")
+
  xlab("Year") +
  ylab("GDP") +
  theme_bw()

(gggdpplacer + gggdpmad + gggdpla)
```

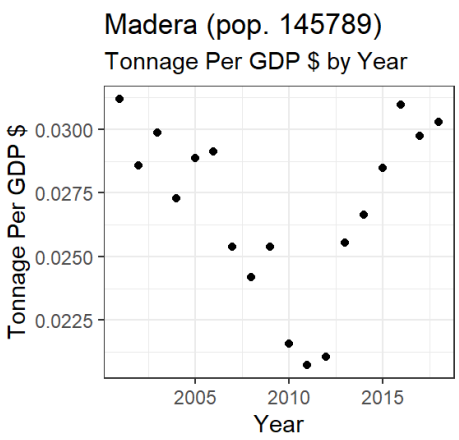


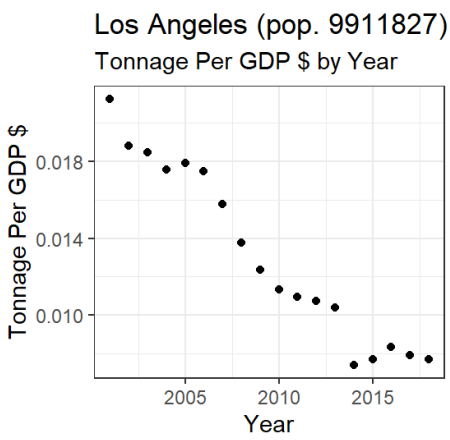





```
gggdpplacerton <- ggplot(placer, aes(Year, total_tons/GDP)) +
  geom_point() +
  ggtitle(paste0("Placer (pop. ", round(mean(placer$population)), ")"), subtitle="Tonnage Per GD
P $ by Year") +
  xlab("Year") +
  ylab("Tonnage Per GDP $") +
  theme_bw()

gggdpmadton <- ggplot(madera, aes(Year, total_tons/GDP)) +
  geom_point() +
  ggtitle(paste0("Madera (pop. ", round(mean(madera$population)), ")"), subtitle="Tonnage Per GD
P $ by Year") +
  xlab("Year") +
  ylab("Tonnage Per GDP $") +
  theme_bw()

gggdplaton <- ggplot(la, aes(Year, total_tons/GDP)) +
  geom_point() +
  ggtitle(paste0("Los Angeles (pop. ", round(mean(la$population)), ")"), subtitle="Tonnage Per G
DP $ by Year") +
  xlab("Year") +
  ylab("Tonnage Per GDP $") +
  theme_bw()

(gggdpplacerton + gggdpmadton + gggdplaton)
```







```
placer <- placer %>% select(-County)
madera <- madera %>% select(-County)
la <- la %>% select(-County)
mod.placer <- lm(total_tons ~ .-Year, data=placer)
mod.madera <- lm(total_tons ~ .-Year, data=madera)
mod.la <- lm(total_tons ~ .-Year, data=la)

summary(mod.placer)
```

```
##
## Call:
## lm(formula = total_tons ~ . - Year, data = placer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6253.3 -1935.7     1.6  1849.4  5253.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.956e+04  8.866e+04  -0.333   0.7501
## GDP          2.280e-02  1.458e-03  15.634 4.34e-06 ***
## population   5.070e-01  8.362e-01   0.606   0.5665
## n            1.174e+03  1.657e+03   0.708   0.5052
## lag          3.964e+02  1.649e+03   0.240   0.8180
## cumul       -2.244e+03  2.012e+03  -1.115   0.3075
## household_n -4.589e+03  3.247e+03  -1.413   0.2072
## hh_lag      -2.305e+03  1.975e+03  -1.167   0.2875
## industry_n   7.042e+02  1.785e+03   0.395   0.7068
## ind_lag      8.417e+02  1.914e+03   0.440   0.6756
## ind_cumul    5.963e+02  2.173e+03   0.274   0.7929
## hh_cumul     8.191e+03  3.105e+03   2.638   0.0387 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5224 on 6 degrees of freedom
## Multiple R-squared:  0.9889, Adjusted R-squared:  0.9687
## F-statistic: 48.76 on 11 and 6 DF,  p-value: 5.826e-05
```

```
summary(mod.madera)
```

```
##
## Call:
## lm(formula = total_tons ~ . - Year, data = madera)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9888  -5678    525   4051   9922
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.233e+06  3.714e+05   3.318  0.01604 *
## GDP          5.164e-02  1.041e-02   4.962  0.00255 **
## population  -1.115e+01  3.920e+00  -2.845  0.02938 *
## n            1.589e+03  2.628e+03   0.604  0.56772
## lag          3.804e+01  2.486e+03   0.015  0.98829
## cumul       -9.671e+03  3.797e+03  -2.547  0.04368 *
## household_n -1.339e+04  8.719e+03  -1.536  0.17543
## hh_lag      -8.009e+03  5.127e+03  -1.562  0.16929
## industry_n  -1.195e+03  2.690e+03  -0.444  0.67244
## ind_lag      9.402e+02  2.946e+03   0.319  0.76047
## ind_cumul    1.064e+04  4.042e+03   2.633  0.03891 *
## hh_cumul     1.618e+04  9.424e+03   1.717  0.13680
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10030 on 6 degrees of freedom
## Multiple R-squared:  0.9729, Adjusted R-squared:  0.9232
## F-statistic: 19.58 on 11 and 6 DF,  p-value: 0.0008109
```

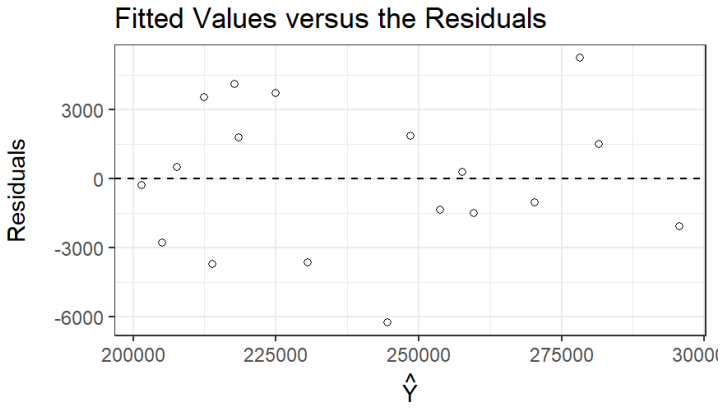
```
summary(mod.la)
```

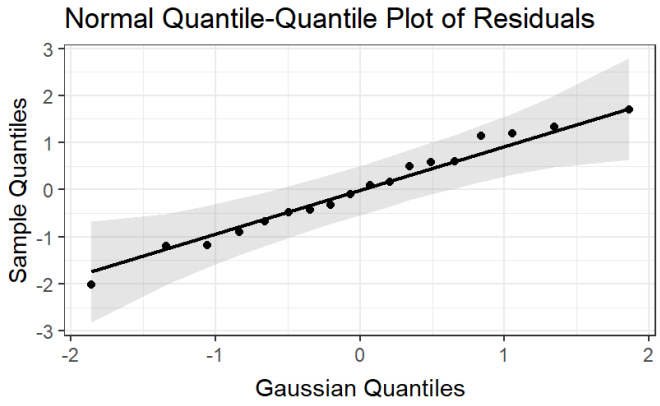
```
##
## Call:
## lm(formula = total_tons ~ . - Year, data = la)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -517810 -128589   50307  153975  422680
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.695e+07  4.072e+07   1.399   0.2115
## GDP          4.235e-02  1.259e-02   3.364   0.0151 *
## population  -6.797e+00  4.786e+00  -1.420   0.2054
## n           -2.153e+05  1.393e+05  -1.546   0.1731
## lag         -1.617e+05  1.005e+05  -1.609   0.1587
## cumul        2.306e+05  1.871e+05   1.233   0.2637
## household_n  1.864e+05  2.904e+05   0.642   0.5447
## hh_lag       9.444e+04  1.759e+05   0.537   0.6106
## industry_n   2.706e+05  1.263e+05   2.142   0.0759 .
## ind_lag      2.072e+05  1.120e+05   1.849   0.1139
## ind_cumul   -3.180e+05  1.907e+05  -1.667   0.1465
## hh_cumul     2.202e+04  3.418e+05   0.064   0.9507
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444600 on 6 degrees of freedom
## Multiple R-squared:  0.9814, Adjusted R-squared:  0.9472
## F-statistic: 28.73 on 11 and 6 DF,  p-value: 0.0002714
```

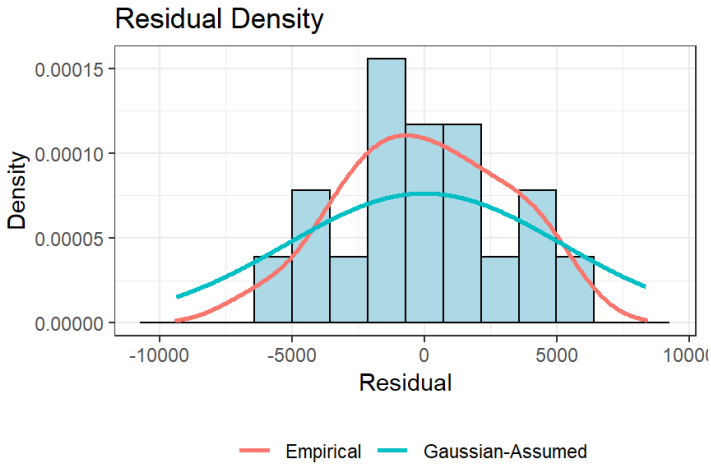
```
plotResiduals(mod.placer)
```

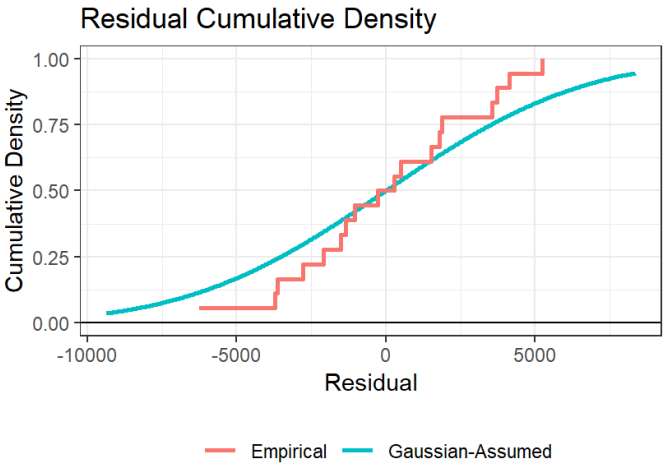
```
##
## Attaching package: 'qqplotr'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```


Fitted Values versus the Residuals
Residuals
3000
0
-3000
-6000
200000
225000
250000
275000
Ŷ


Normal Quantile-Quantile Plot of Residuals
Sample Quantiles
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
Gaussian Quantiles


Residual Density
Density
0.00015
0.00010
0.00005
0.00000
-10000
-5000
0
5000
Residual


Residual Cumulative Density
Cumulative Density
1.00
0.75
0.50
0.25
0.00
-10000
-5000
0
5000
Residual

Empirical
Gaussian-Assumed

Empirical
Gaussian-Assumed