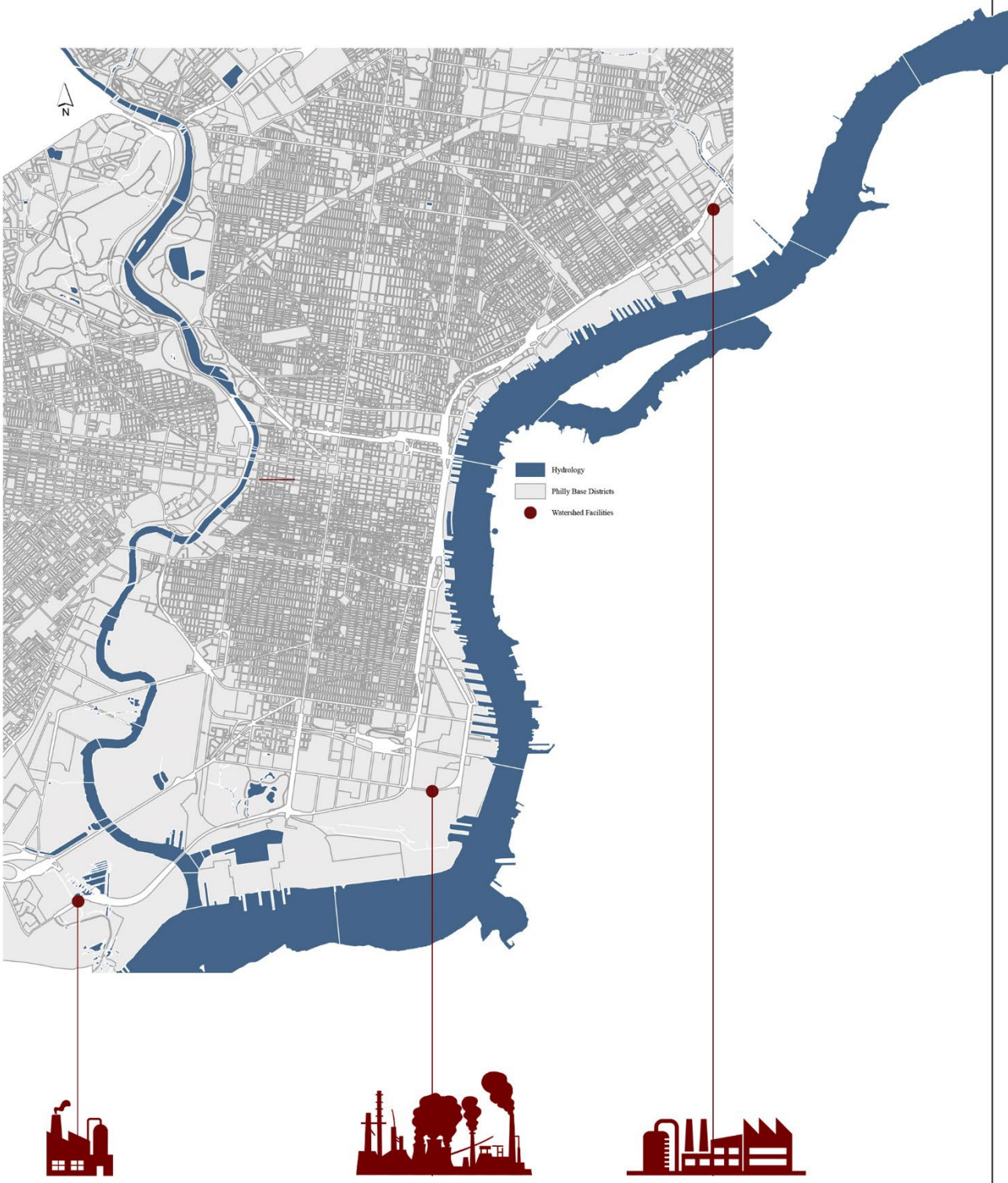
ASSIGNMENT 2

A Study of Philadelphia's Water Infrastructure Needs



Stephanie Onuaja & Chris Michael
CPLN 505, Spring' 22

MEMORANDUM

TO: Mayor Jim Kenney of Philadelphia, Pennsylvania
FROM: Stephanie Onuaja and Christina Michael
DATE: 3rd March 2022
RE: Analyzing Watershed Infrastructure Needs in Philadelphia County

This memorandum is a comprehensive analysis of the capital investments required to meet Philadelphia County's wastewater and stormwater needs. The $1.2 billion figure includes capital needs for the three publicly owned wastewater infrastructure and treatment facilities in the county. We have developed statistical models to evaluate the interrelationship of potential indicators and variables to the high infrastructure need. Multiple associations, correlation, and regression tests were run based on data obtained from the Clean Watersheds Needs Survey 2012 Report to Congress, to understand the future demand for water infrastructure in Philadelphia. Three key takeaways have been identified from the models that were developed and what they mean for the County and State as a whole.

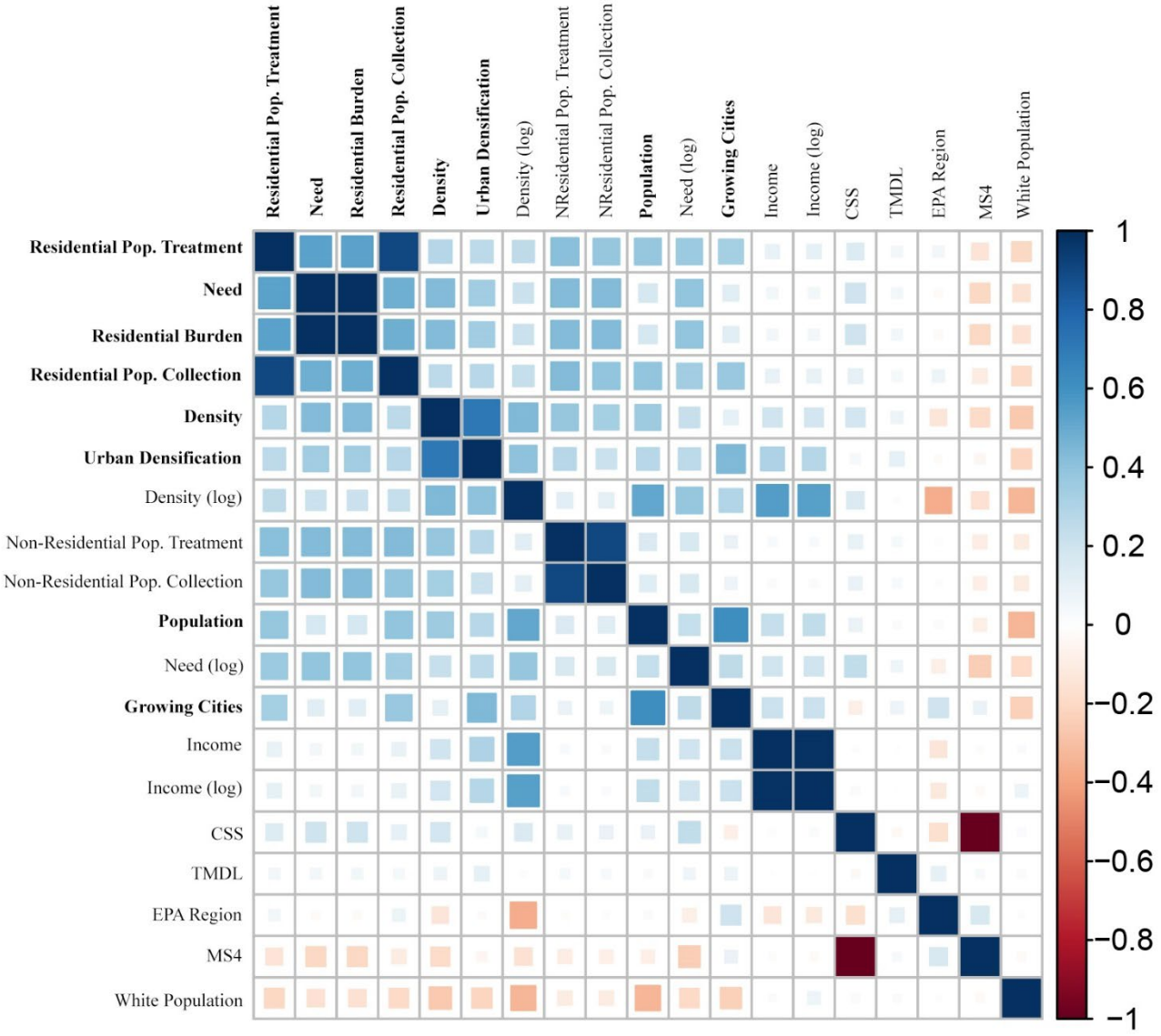
Philadelphia County has a high need for capital investments in watershed infrastructure compared to other counties in Pennsylvania

When comparing the financial need of watershed facilities across the state, Philadelphia County's facilities rank amongst the five highest within PA. It is crucial to categorize Philadelphia's needs in the context of Pennsylvania to prioritize funding and implementation to attenuate these needs. The current Combined Sewer System serves approximately 60% of the county is tasked with channeling both stormwater and wastewater to a wastewater treatment facility. Due to the high presence of impervious surfaces in Philadelphia, the systems tend to be overloaded in the rainy seasons causing runoff and contamination of the main water bodies in Philadelphia. Knowing these environmental challenges, the city should invest in multifaceted solutions that deal with both the causes and effects.

Philadelphia's high water infrastructure need can be attributed to its high population density

The correlation matrix below shows the strength of the linear relationship between variables when tested against the total infrastructural need. The matrix indicates that growth in residential population density is followed by a corresponding increase in the need for water infrastructure. This urban densification has to be supplemented by adequate infrastructure provisions that can cater to this need. Other variables that closely affect water infrastructure needs are population growth over time and the number of people receiving access to wastewater collection and treatment. Despite being an effective indicator of demographic change, the weaker correlation between need and median household income can be

attributed to the fact that need fluctuates with household size and is not directly tied to income levels. Median income also does not accurately capture disadvantaged households within a community of diverse economic standing.



Philadelphia's facilities are among the highest in residential burden for water infrastructure in Pennsylvania

The residential burden better reflects the reality of what customers need to pay to cover their basic water infrastructure needs. Multiple regressions were run to estimate the effect that residential burden has on the need for water infrastructure. The matrix indicates a high correlation between residential burden and infrastructure needs and helps explain the affordability challenge posed by water sector costs. Families of similar sizes pay about the same for public services and utilities regardless of household income, creating

a bigger burden for low-income households. The interdependency of need and residential burden provides for an assessment that could inform decision-making and future assessment of the water infrastructure needs of Philadelphia.

Concluding thoughts

From our preliminary statistical models, there are a few strategies that Philadelphia County and governing bodies can take to mitigate the financial and infrastructural burden on the city's current water infrastructure systems. The capability of the facilities to service the growing population has to be determined to plan for the city's future infrastructural capacity demands.

- Further studies and analysis should be conducted to understand how Philadelphia's facilities have performed over time and to establish the necessity for regular maintenance, and repairs.
- Due to the tendency of the Combined Sewer System (CSS) to have high loads on the treatment facilities, the capacities of the facilities must be examined and supplemented to prevent overloading.
- Investments in Green Stormwater Infrastructure should be maintained and expanded to reduce the impacts of impervious surfaces and to serve as a means to alleviate the pressure on the current wastewater systems.
- Financial relief programs and revised fiscal funding opportunities can be implemented to reduce the residential burden on low-income households.

CPLN 505 Assignment 2

By Stephanie Onuaja & Chris Michael

April 1st, 2022

A Study of Philadelphia's Water Infrastructure Needs

This report is a comprehensive analysis of the capital investments required to meet Philadelphia County's wastewater and stormwater needs. The $1.2 billion figure includes capital needs for the three publicly owned wastewater infrastructure and treatment facilities in the county. We have developed statistical models to evaluate the interrelationship of potential indicators and variables to the high infrastructure need. Multiple associations, correlation, and regression tests were run based on data obtained from the Clean Watersheds Needs Survey 2012 Report to Congress, to understand the future demand for water infrastructure in Philadelphia. The accompanying memorandum includes three key takeaways have been identified from the models that were developed and what they mean for the County and State as a whole.

Part I. Comparing Groups & Creating Metrics to Contexualize Reported Need

```
dat<- mutate(dat, pc_res_coll=PROJ_RES_REC_COLLCTN-PRES_RES_REC_COLLCTN)
```

1.1 Projected change in residential population receiving collection

```
dat <- mutate(dat, ppc_res_coll=(pc_res_coll/PRES_RES_REC_COLLCTN)*100)
```

1.2 Projected percent change in residential population receiving collection

```
dat <- mutate(dat, pc_res_treat=PROJ_RES_REC_TRMT-PRES_RES_REC_TRMT)
```

1.3 Projected change in residential population receiving treatment

```
dat <- mutate(dat, ppc_res_treat=(pc_res_treat/PRES_RES_REC_TRMT)*100)
```

1.4 Projected percent change in residential population receiving treatment

```
dat <- mutate(dat, pc_nonres_coll=PROJ_N_RES_REC_COLLCTN-PRES_N_RES_REC_COLLCTN)
```

1.5 Projected change in non-residential population receiving collection

```
dat <- mutate(dat, ppc_nonres_coll=(pc_nonres_coll/PRES_N_RES_REC_COLLCTN)*100)
```

1.6 Projected percent change in non-residential population receiving collection

```
dat <- mutate(dat, pc_nonres_treat=PROJ_N_RES_REC_TRMT-PRES_N_RES_REC_TRTM)
```

1.7 Projected change in non-residential population receiving treatment

```
dat <- mutate(dat, ppc_nonres_treat=(pc_nonres_treat/PRES_N_RES_REC_TRTM)*100)
```

1.8 Projected percent change in non-residential population receiving treatment

```
dat <- mutate(dat, ALAND2=ALAND/2.59e+6)
dat <- mutate(dat, res_pop_den=POP10/ALAND2)
```

1.9 Residential population density

```
dat <- mutate(dat, pc_res_pop_den=(POP10-POP00)/ALAND2)
dat <- mutate(dat, ppc_res_pop_den=(pc_res_pop_den/(POP00/ALAND2))*100)
```

1.10 Change & percent change in population density

```
dat <- mutate(dat, pc_medinc=MEDINC09-MEDINC99)
dat <- mutate(dat, ppc_medinc=(pc_medinc/MEDINC99)*100)
```

1.11 Percent change in median income

```
dat <- mutate(dat, PROJ_REC_COLL=PROJ_RES_REC_COLLCTN+PROJ_N_RES_REC_COLLCTN)
```

1.12 Projected res + non res receiving collection

```
dat <- mutate(dat, PROJ_REC_TRMT=PROJ_RES_REC_TRMT+PROJ_N_RES_REC_TRMT)
```

1.13 Projected res + non res receiving treatment (same as collection, exclude)

```
dat2 <-
  mutate(dat2, TOT_NEED_LIM = case_when(TOTAL_OFFICIAL_NEED>=0 &
                                          TOTAL_OFFICIAL_NEED<763000 ~ "1",
                                        TOTAL_OFFICIAL_NEED>=763000 &
                                          TOTAL_OFFICIAL_NEED<2499000 ~ "2",
                                        TOTAL_OFFICIAL_NEED>=2499000 &
                                          TOTAL_OFFICIAL_NEED<7906000 ~ "3",
                                        TOTAL_OFFICIAL_NEED>=7906000 ~ "4"))
```

1.14 Setting limits for need

```
dat2 <-
  mutate(dat2, COLL_LIMIT = case_when(PROJ_REC_COLL>=0 &
                                        PROJ_REC_COLL<1110 ~ "1",
                                      PROJ_REC_COLL>=1110 &
                                        PROJ_REC_COLL<3500 ~ "2",
                                      PROJ_REC_COLL>=3500 &
                                        PROJ_REC_COLL<13708 ~ "3",
                                      PROJ_REC_COLL>=13708 ~ "4"))
```

1.15 Setting limits for projected population receiving collection

```
dat2 <-
  mutate(dat2, DEN_LIMIT = case_when(res_pop_den>=0 &
                                       res_pop_den<=37.30 ~ "1",
                                     res_pop_den>=37.30 &
                                       res_pop_den<104.39 ~ "2",
                                     res_pop_den>=104.39 &
                                       res_pop_den<378.32 ~ "3",
                                     res_pop_den>=378.32 ~ "4"))
```

1.16 Setting limits for pop density

```
dat2 <-
  mutate(dat2, INC_LIMIT = case_when(MEANINC09>=0 & MEANINC09<=59105 ~ "1",
                                     MEANINC09>=59105 & MEANINC09<55332 ~ "2",
                                     MEANINC09>=55332 & MEANINC09<178541 ~ "3",
                                     MEANINC09>=178541 ~ "4"))
```

1.17 Setting limits for med income

```
EPA1 <- subset.data.frame(dat2, EPA_REGION=="1")
EPA2 <- subset.data.frame(dat2, EPA_REGION=="2")
EPA3 <- subset.data.frame(dat2, EPA_REGION=="3")
EPA4 <- subset.data.frame(dat2, EPA_REGION=="4")
EPA5 <- subset.data.frame(dat2, EPA_REGION=="5")
EPA6 <- subset.data.frame(dat2, EPA_REGION=="6")
EPA7 <- subset.data.frame(dat2, EPA_REGION=="7")
EPA8 <- subset.data.frame(dat2, EPA_REGION=="8")
EPA9 <- subset.data.frame(dat2, EPA_REGION=="9")
EPA10 <- subset.data.frame(dat2, EPA_REGION=="10")
```

1.18 Creating subsets for different EPA regions

Part II. Providing Summary Statistics of the Data

```
dat2 <- dat2[order(-dat2$TOTAL_OFFICIAL_NEED),]
most_need <-slice_head(dat2, n=10)
least_need <-slice_tail(dat2, n=10 )
least_need <- least_need[order(least_need$TOTAL_OFFICIAL_NEED),]
PA_need <-filter(dat2, STATE=="PA")
Philly_need <-filter(dat2, COUNTYNAME=="Philadelphia")
```

2.1 Top and bottom 10 facilities (not based on EPA region)

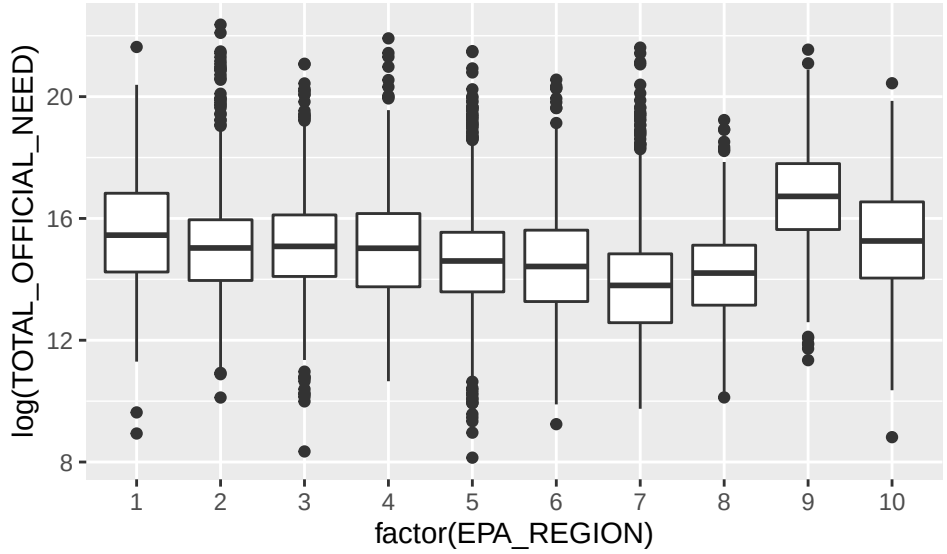
```
EPAsum <- dat2 %>%
  group_by(EPA_REGION,) %>%
  summarise(TOTAL_OFFICIAL_NEED = mean(TOTAL_OFFICIAL_NEED),
            MEDINC = median(MEDINC09),
            POPDEN = mean(res_pop_den)) %>% as.data.frame(EPAsum)

summary(EPAsum)
```

2.2 Summary statistics of EPA region by need, income and pop density

2.3 Box plots, Density Plots, Histograms

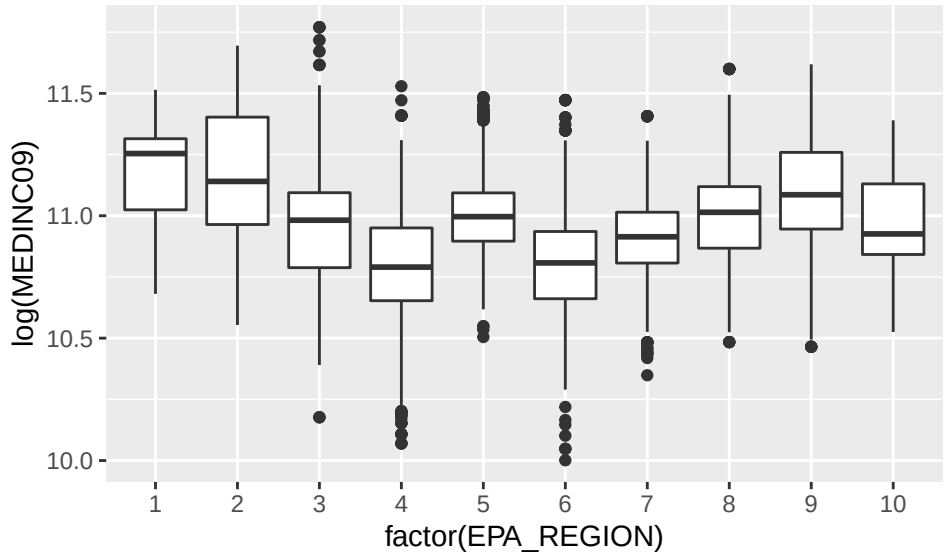
2.3.1 Box plots- need, income and pop density by EPA region A. Box plot of total official need by EPA region

```
ggplot(dat2, aes(x=factor(EPA_REGION), y=log(TOTAL_OFFICIAL_NEED)))+
  geom_boxplot()
```



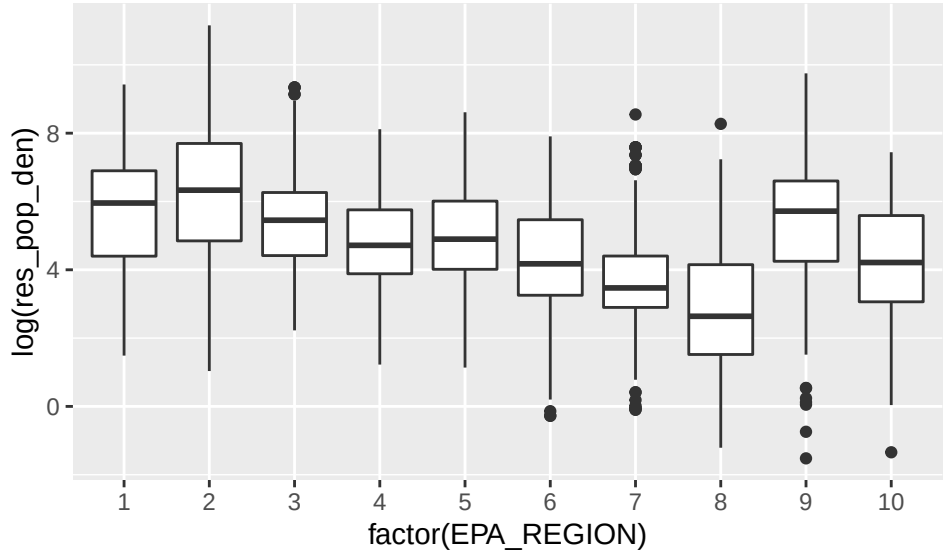
B. Box plot of median income (2009) by EPA region

```
ggplot(dat2, aes(x=factor(EPA_REGION), y=log(MEDINC09)))+
  geom_boxplot()
```



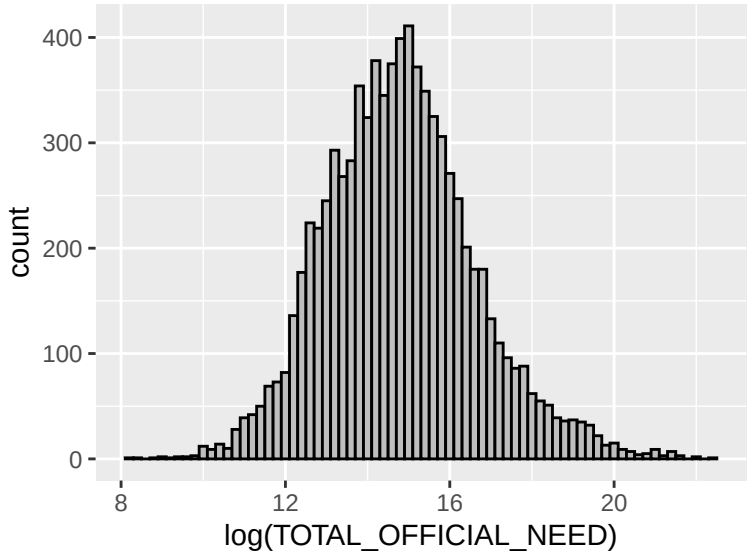
C. Box plot of residential population density (2010) by EPA region

```
ggplot(dat2, aes(x=factor(EPA_REGION), y=log(res_pop_den)))+
  geom_boxplot()
```



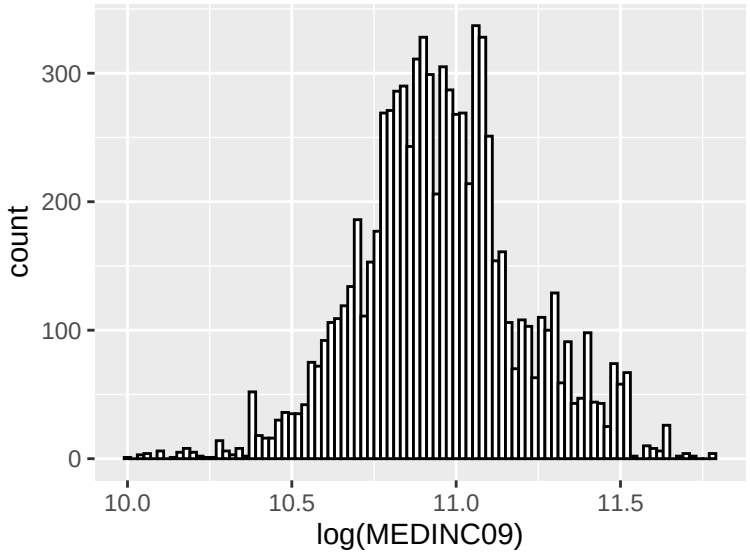
2.3.2 Histograms A. Histogram of total official need

```
ggplot(dat2, aes(x=log(TOTAL_OFFICIAL_NEED))) +
  geom_histogram(binwidth=0.2, color="black", fill="grey")
```



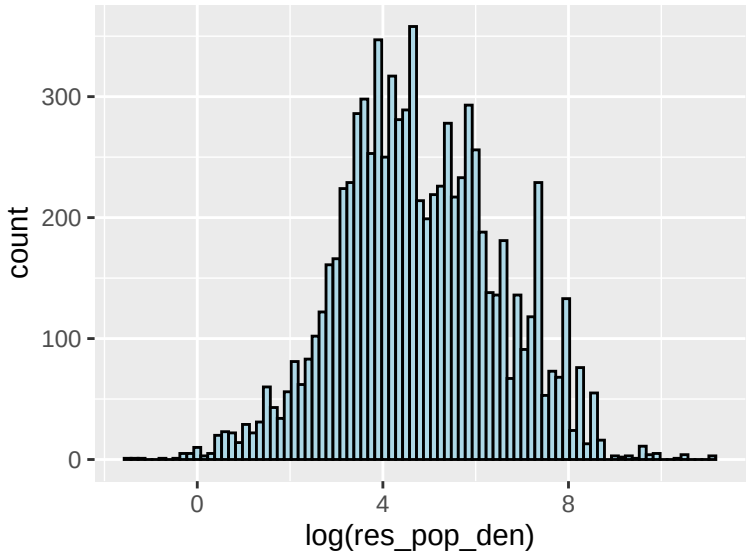
B. Histogram of median income (2009)

```
ggplot(dat2, aes(x=log(MEDINC09))) +
  geom_histogram(binwidth=0.02, color="black", fill="white")
```



C. Histogram of residential population density (2010)

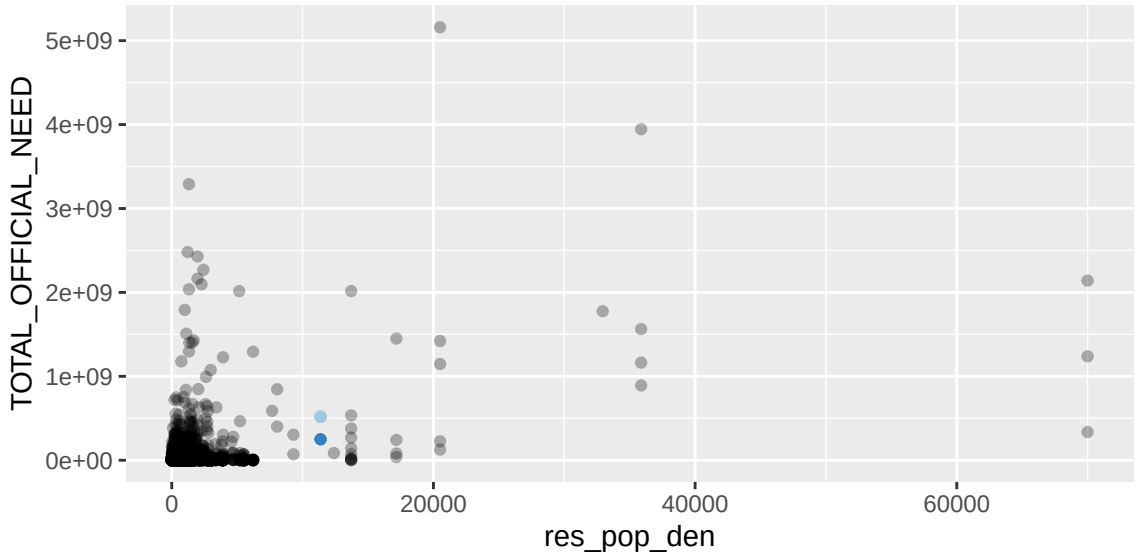
```
ggplot(dat2, aes(x=log(res_pop_den))) +
  geom_histogram(binwidth=0.15, color="black", fill="lightblue")
```



2.3.3 Density plots A. Total official need of facilities in the USA by residential pop den

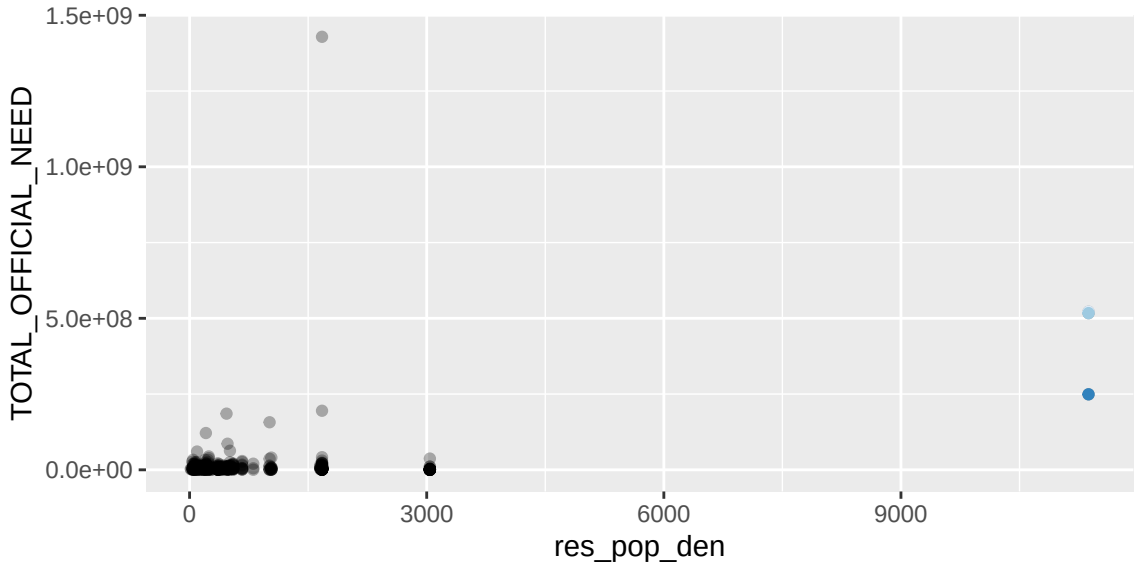
```
ggplot(dat2, aes(x=res_pop_den, y=TOTAL_OFFICIAL_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_need,
```

```
            mapping =  aes(x=res_pop_den, y=TOTAL_OFFICIAL_NEED),
            col=brewer.pal (n=3,name = "Blues"))
```



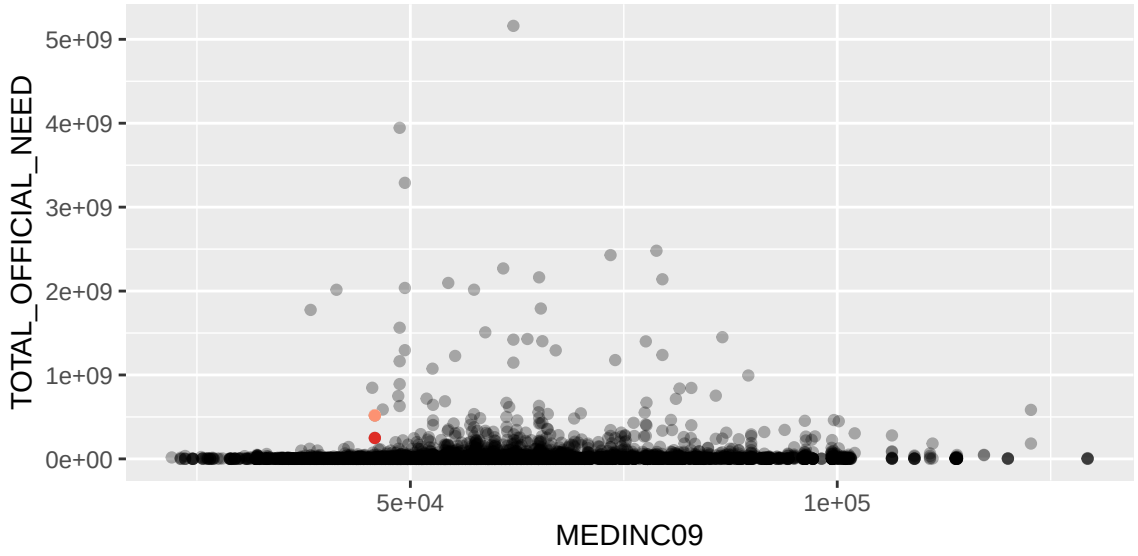
B. Total official need of facilities in PA by residential pop den

```
ggplot(PA_need, aes(x=res_pop_den, y=TOTAL_OFFICIAL_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_need,
             mapping =  aes(x=res_pop_den, y=TOTAL_OFFICIAL_NEED),
             col=brewer.pal (n=3,name = "Blues"))
```



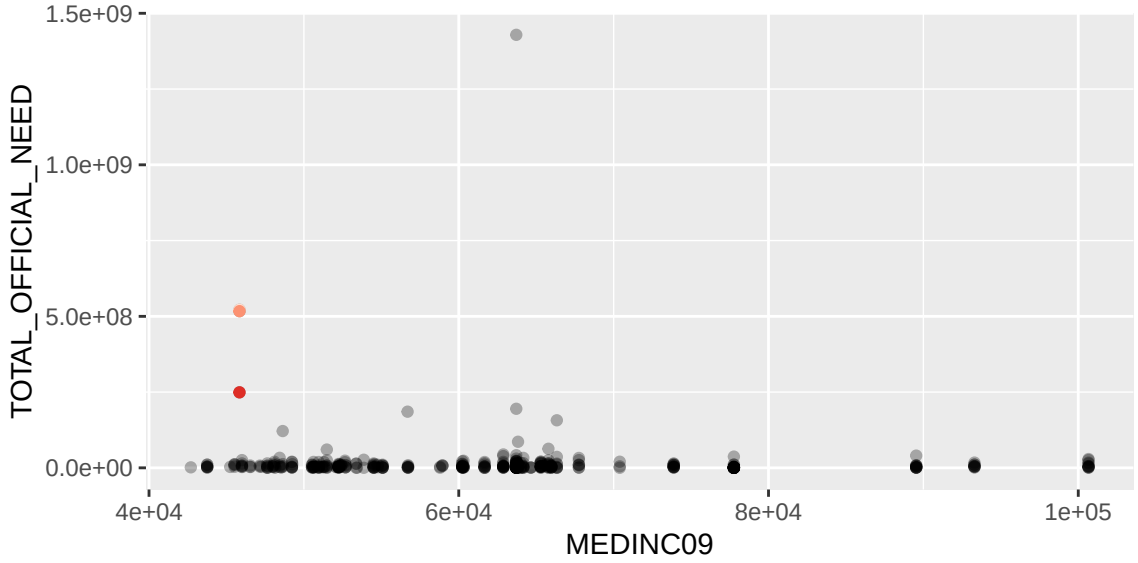
C. Total official need of facilities in the USA by median income

```
ggplot(dat2, aes(x=MEDINC09, y=TOTAL_OFFICIAL_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_need,
             mapping =  aes(x=MEDINC09, y=TOTAL_OFFICIAL_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```



D. Total official need of facilities in PA by medan income

```
ggplot(PA_need, aes(x=MEDINC09, y=TOTAL_OFFICIAL_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_need,
             mapping =  aes(x=MEDINC09, y=TOTAL_OFFICIAL_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```



Part III. Tests of Association

```
t.test (TOTAL_OFFICIAL_NEED ~ CSS, data = dat2, paired = FALSE)
```

3.1 Association Test of Total Needs in CSS vs. MS4s

```
##
##  Welch Two Sample t-test
##
## data:  TOTAL_OFFICIAL_NEED by CSS
## t = -6.6985, df = 696.31, p-value = 4.344e-11
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -129163169  -70608544
## sample estimates:
## mean in group 0 mean in group 1
##        13656852       113542709
```

```
t.test (TOTAL_OFFICIAL_NEED ~ MS4, data = dat2, paired = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  TOTAL_OFFICIAL_NEED by MS4
## t = 6.6985, df = 696.31, p-value = 4.344e-11
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##   70608544 129163169
## sample estimates:
## mean in group 0 mean in group 1
##       113542709        13656852
```

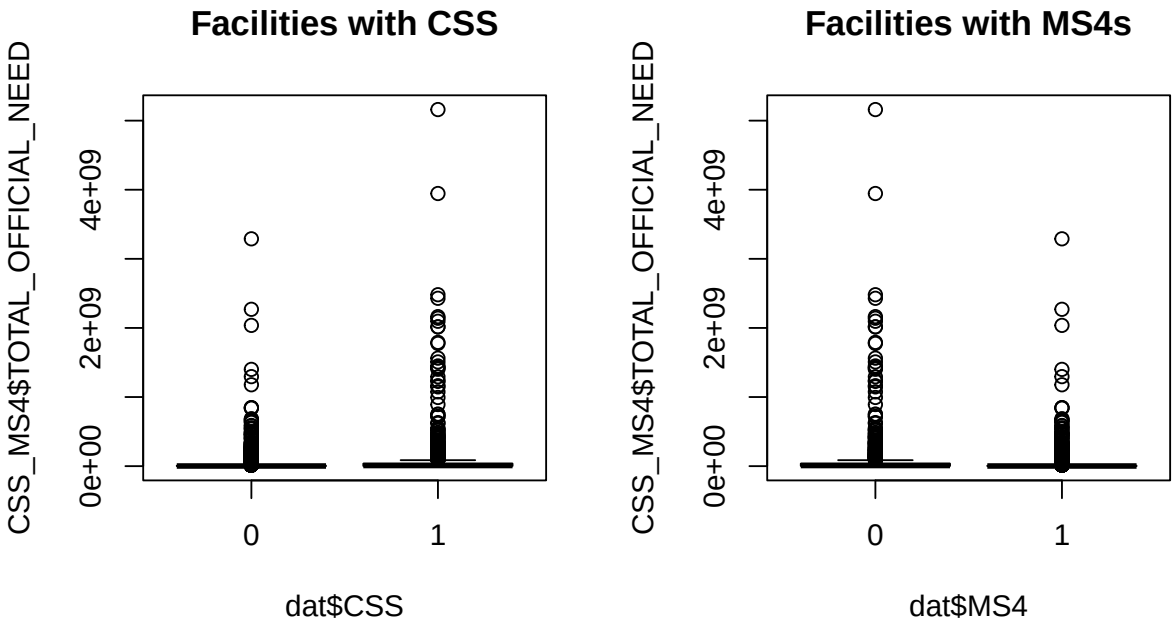
```
dat2[c("TOTAL_OFFICIAL_NEED", "CSS", "MS4")]
```

```
CSS_MS4 <- c("TOTAL_OFFICIAL_NEED", "CSS", "MS4")
CSS_MS4 <- dat[CSS_MS4]
CSS <- subset(CSS_MS4, CSS_MS4$CSS=="1")
MS4 <- subset(CSS_MS4, CSS_MS4$MS4=="1")

t.test (CSS$TOTAL_OFFICIAL_NEED, MS4$TOTAL_OFFICIAL_NEED)
```

```
##
##  Welch Two Sample t-test
##
## data:  CSS$TOTAL_OFFICIAL_NEED and MS4$TOTAL_OFFICIAL_NEED
## t = 6.712, df = 699.19, p-value = 3.972e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   70514032 128823442
## sample estimates:
## mean of x mean of y
## 113053301  13384564
```

```
par(mfrow=c(1, 2))
boxplot ( CSS_MS4$TOTAL_OFFICIAL_NEED ~ dat$CSS, main = "Facilities with CSS")
boxplot ( CSS_MS4$TOTAL_OFFICIAL_NEED ~ dat$MS4, main = "Facilities with MS4s")
```

3.2 Box plots of facilities with CSS vs. MS



3.3 TMDL A. Facilities and TMDL (not by region)

```
t.test (TOTAL_OFFICIAL_NEED ~ TMDL_INDICATOR, data = dat2, paired = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  TOTAL_OFFICIAL_NEED by TMDL_INDICATOR
## t = -2.2094, df = 304.91, p-value = 0.02789
## alternative hypothesis: true difference in means between group N and group Y is not equal to 0
## 95 percent confidence interval:
##  -72917647  -4217568
## sample estimates:
## mean in group N mean in group Y
##        20624477        59192084
```

```
summary(dat2, TOTAL_OFFICIAL_NEED)
kable(CrossTable(dat2$TMDL_INDICATOR, dat2$EPA_REGION, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS"))
```

3.4 Cross tables

```
ass_vars <- dat2 %>%
  group_by(COUNTYNAME,) %>%
  summarise(EPA_REGION = (EPA_REGION),
            TOT_NEED_LIM = (TOT_NEED_LIM),
            COLL_LIMIT = (COLL_LIMIT),
            TMDL_INDICATOR = (TMDL_INDICATOR),
            INC_LIMIT = (INC_LIMIT),
            DEN_LIMIT = (DEN_LIMIT))


CrossTable(ass_vars$TOT_NEED_LIM, ass_vars$EPA_REGION, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS")

CrossTable(ass_vars$TOT_NEED_LIM, ass_vars$COLL_LIMIT, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS")

CrossTable(ass_vars$TOT_NEED_LIM, ass_vars$TMDL_INDICATOR, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS")

CrossTable(ass_vars$TOT_NEED_LIM, ass_vars$INC_LIMIT, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS")

CrossTable(ass_vars$TOT_NEED_LIM, ass_vars$DEN_LIMIT, fisher = FALSE, chisq = TRUE,
           expected = TRUE, sresid = FALSE, format = "SPSS")
```

3.5 Association variables table

Part IV. Correlation

```
dat2 <- dat2 %>%
  mutate(TMDL = ifelse(dat2$TMDL_INDICATOR == "Y", 1,0))

corr_vars <- dat2 %>%
  group_by(COUNTYNAME,) %>%
  summarise(EPA_REGION = (EPA_REGION),
            TOTAL_OFFICIAL_NEED = (TOTAL_OFFICIAL_NEED),
            PROJ_REC_COLL = (PROJ_REC_COLL),
            TMDL = (TMDL),
```

```
            MEDINC09 = (MEDINC09),
            res_pop_den = (res_pop_den))

cor(corr_vars$TOTAL_OFFICIAL_NEED, corr_vars$EPA_REGION,
    use="complete.obs", method="pearson")
```

4.1 Need, EPA, income, density, TMDL, collection

```
## [1] -0.0285843
```

```
cor(corr_vars$TOTAL_OFFICIAL_NEED, corr_vars$MEDINC09,
    use="complete.obs", method="pearson")
```

```
## [1] 0.05467852
```

```
cor(corr_vars$TOTAL_OFFICIAL_NEED, corr_vars$res_pop_den,
    use="complete.obs", method="pearson")
```

```
## [1] 0.4302197
```

```
cor(corr_vars$TOTAL_OFFICIAL_NEED, corr_vars$TMDL,
    use="complete.obs", method="pearson")
```

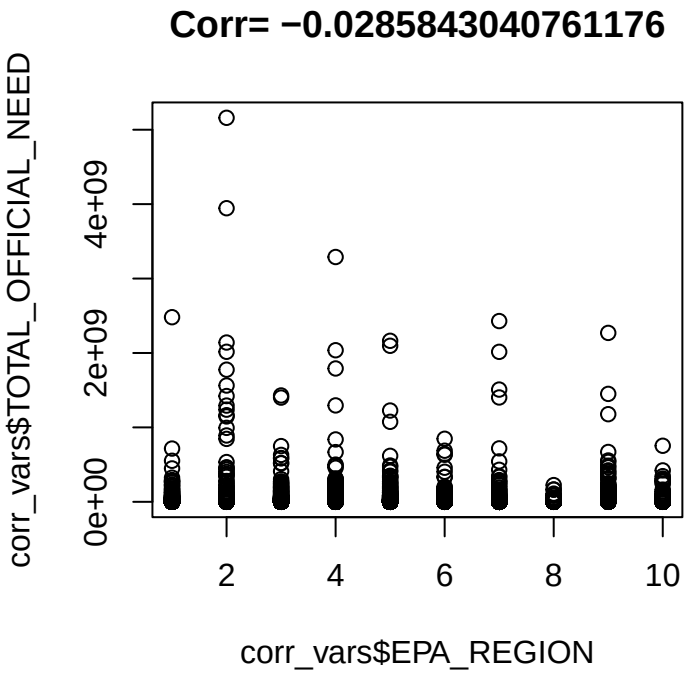
```
## [1] 0.0533406
```

```
cor(corr_vars$TOTAL_OFFICIAL_NEED, corr_vars$PROJ_REC_COLL,
    use="complete.obs", method="pearson")
```

```
## [1] 0.5151487
```

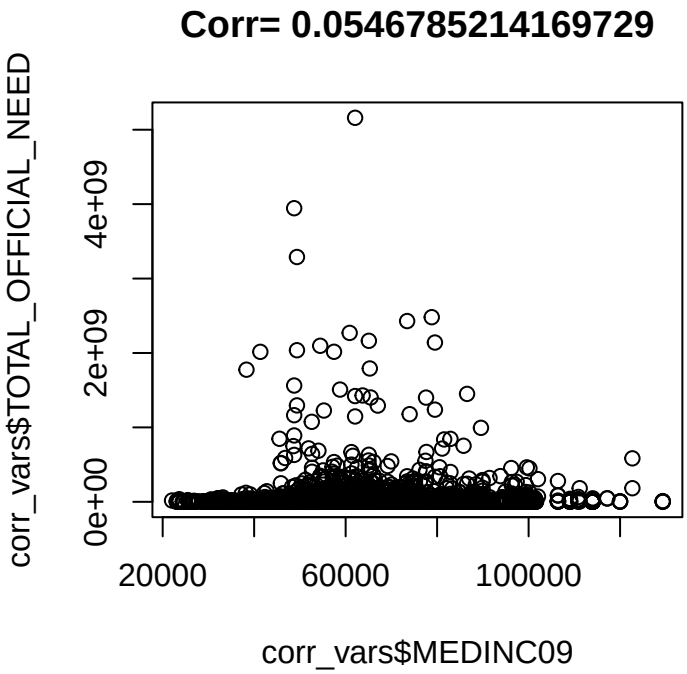
4.2 Scatterplots (5 against need) A. Total official need by EPA region

```
plot(corr_vars$EPA_REGION, corr_vars$TOTAL_OFFICIAL_NEED,
     main=paste("Corr=", cor(corr_vars$EPA_REGION, corr_vars$TOTAL_OFFICIAL_NEED,
                             use="complete.obs", method="pearson")))
```



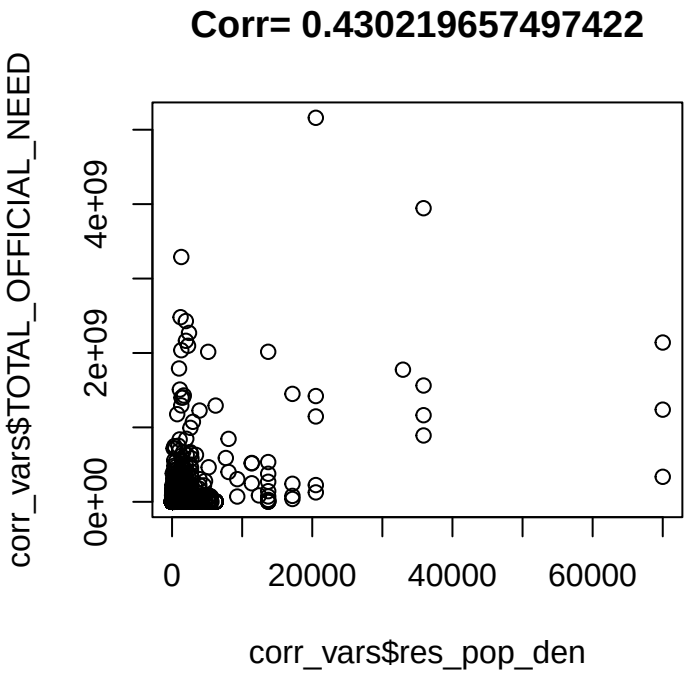
B. Total official need by median income (2009)

```
plot(corr_vars$MEDINC09, corr_vars$TOTAL_OFFICIAL_NEED,
     main=paste("Corr=", cor(corr_vars$MEDINC09, corr_vars$TOTAL_OFFICIAL_NEED,
                             use="complete.obs", method="pearson")))
```



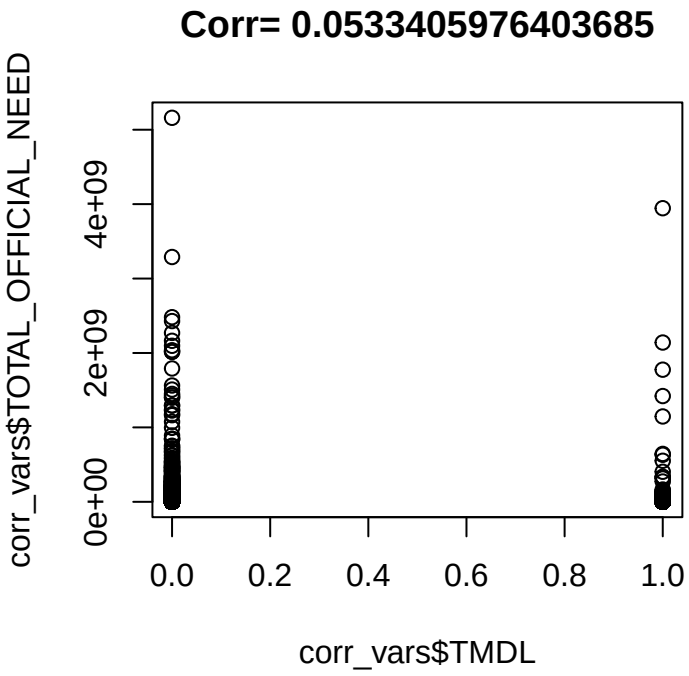
C. Total official need by residential population density

```
plot(corr_vars$res_pop_den, corr_vars$TOTAL_OFFICIAL_NEED,
     main=paste("Corr=", cor(corr_vars$res_pop_den, corr_vars$TOTAL_OFFICIAL_NEED,
                             use="complete.obs", method="pearson")))
```



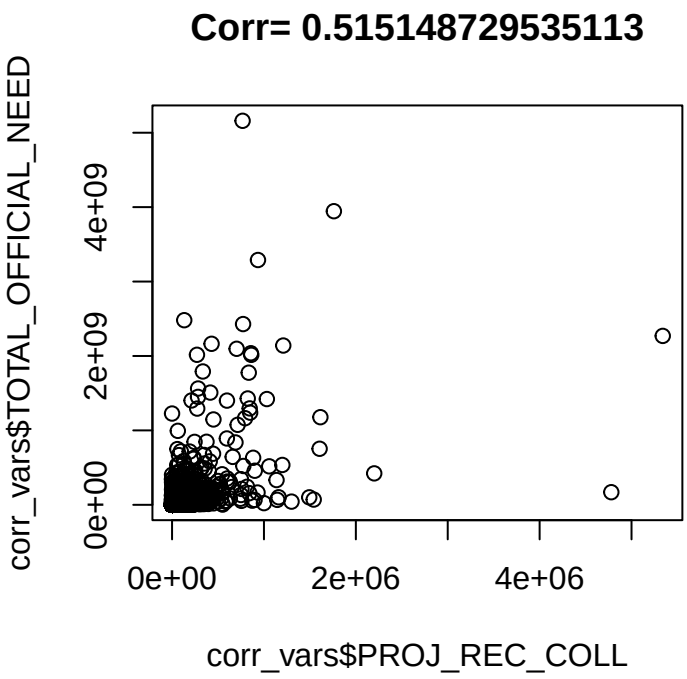
D. Total official need by TMDL indicator

```
plot(corr_vars$TMDL, corr_vars$TOTAL_OFFICIAL_NEED,
     main=paste("Corr=", cor(corr_vars$TMDL, corr_vars$TOTAL_OFFICIAL_NEED,
                             use="complete.obs", method="pearson")))
```



E. Total official need by projected residences + non residences receiving collection

```
plot(corr_vars$PROJ_REC_COLL, corr_vars$TOTAL_OFFICIAL_NEED,
     main=paste("Corr=", cor(corr_vars$PROJ_REC_COLL, corr_vars$TOTAL_OFFICIAL_NEED,
                             use="complete.obs", method="pearson")))
```



Part V. Regressions

```
regress<-(dat2)
```

4.1 Data Preparation and Preliminary Tests

```
regress <- mutate(regress, LOG_NEED=(log(TOTAL_OFFICIAL_NEED)))
```

4.1.1 Log of total official need

```
regress <- mutate(regress, res_burden=(TOTAL_OFFICIAL_NEED/max(PROJ_RES_REC_COLLCTN,
                                                               PROJ_RES_REC_TRMT)))
```

4.1.2 Residential burden

```
regress <- mutate(regress, log_res_burden=(log(res_burden)))
```

4.1.3 Log of Residential burden

```
regress <- mutate(regress, log_inc=log(MEDINC09))
```

4.1.4 Log of median income

```
regress <- mutate(regress, log_pop=log(POP10))
```

4.1.5 Log of population variables

```
regress <- mutate(regress, grow_city=POP10-POP80)
summary(regress)
regress <-
  mutate(regress, grow_city_lim = case_when(grow_city<72449.7 ~ "0",
                                            grow_city>=72449.7 ~ "1"))
```

4.1.6 Growing cities

```
regress <- mutate(regress, dens_city=(POP10/ALAND2)-(POP80/ALAND2))
summary(regress)
regress <-
  mutate(regress, dens_city_lim = case_when(dens_city<83.553 ~ "0",
                                            dens_city>=83.553 ~ "1"))
```

4.1.7 Densifying cities

```
regress <- mutate(regress, log_res_pop_den=log(res_pop_den))
```

4.1.8 Log of residential population density

```
regress <- mutate(regress, res_pop_den_sq=(res_pop_den)^2)
```

4.1.9 Square of residential population density

```
PA_regress <-filter(regress, STATE=="PA")
```

4.1.10 Subset PA from the regression dataset

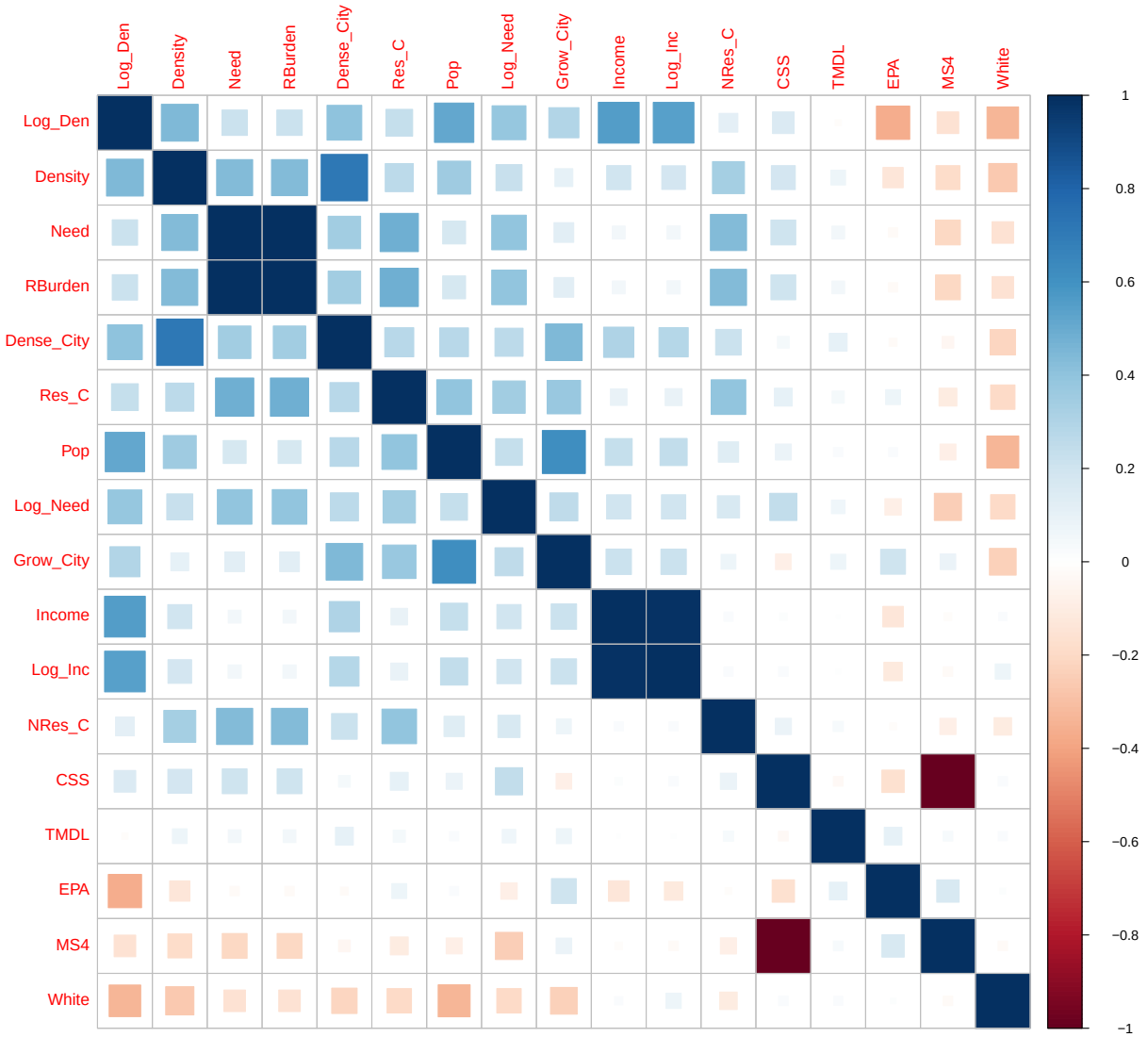
```
Philly_regress <-filter(regress, COUNTYNAME=="Philadelphia")
```

4.1.11 Subset Philly from the regression dataset

```
corr_vars_plot <- regress %>%
  summarise(EPA = (EPA_REGION),
            Need = (TOTAL_OFFICIAL_NEED),
            Log_Need = (LOG_NEED),
            Res_C = (PROJ_RES_REC_COLLCTN),
            NRes_C = (PROJ_N_RES_REC_COLLCTN),
            Pop = (POP00),
            White = (PCTWHITE10),
            TMDL = (TMDL),
            CSS = (CSS),
            MS4 = (MS4),
            RBurden= (res_burden),
            Grow_City = (grow_city),
            Dense_City = (dens_city),
            Income = (MEDINC09),
            Log_Inc =(log_inc),
            Density = (res_pop_den),
            Log_Den = (log_res_pop_den))

M= cor(corr_vars_plot)
```

4.2 Correlation plots to find variables Correlation Plot

```
corrplot(M, method="square", order = 'FPC')
```



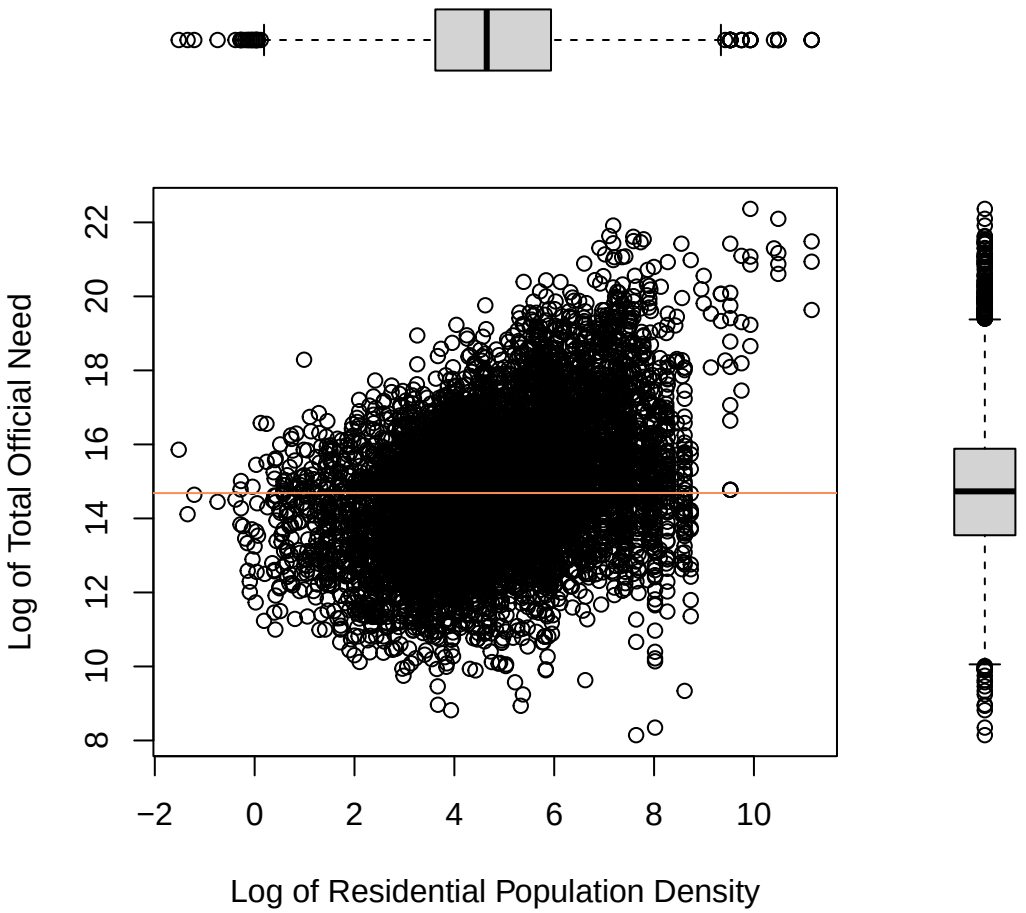
4.3 Bivariate regressions A. Regression 1 Graphs: log need and population density

```
plot.new()

cor.test(regress$log_res_pop_den, regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  regress$log_res_pop_den and regress$LOG_NEED
## t = 37.449, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.3623722 0.3992403
## sample estimates:
##       cor
## 0.3809577
```

```
par(fig=c(0,0.8,0,0.8))

plot(regress$log_res_pop_den, regress$LOG_NEED,
     xlab="Log of Residential Population Density",
     ylab="Log of Total Official Need")

abline(lm(LOG_NEED ~ res_pop_den, data = regress),
       col=brewer.pal (n=3,name = "Spectral"))

par(fig=c(0,0.8,0.55,1), new=TRUE)

boxplot(regress$log_res_pop_den, horizontal=TRUE, axes=FALSE)
par(fig=c(0.65,1,0,0.8),new=TRUE)
boxplot(regress$LOG_NEED,axes=FALSE)
mtext("Scatterplot & Bivariate Regression Plus Univariate Boxplots of Need vs. Density", side=3, outer="
```



Regression 1

```
reg1<-lm (TOTAL_OFFICIAL_NEED ~ res_pop_den, data = regress)
summary(reg1)
```

```
##
## Call:
## lm(formula = TOTAL_OFFICIAL_NEED ~ res_pop_den, data = regress)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.737e+09 -1.103e+07 -6.769e+06 -3.681e+06  4.548e+09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6498601.3  1394485.9    4.66 3.21e-06 ***
```

```
## res_pop_den   29520.1      681.5   43.32  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122500000 on 8261 degrees of freedom
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.185
## F-statistic:  1876 on 1 and 8261 DF,  p-value: < 2.2e-16
```

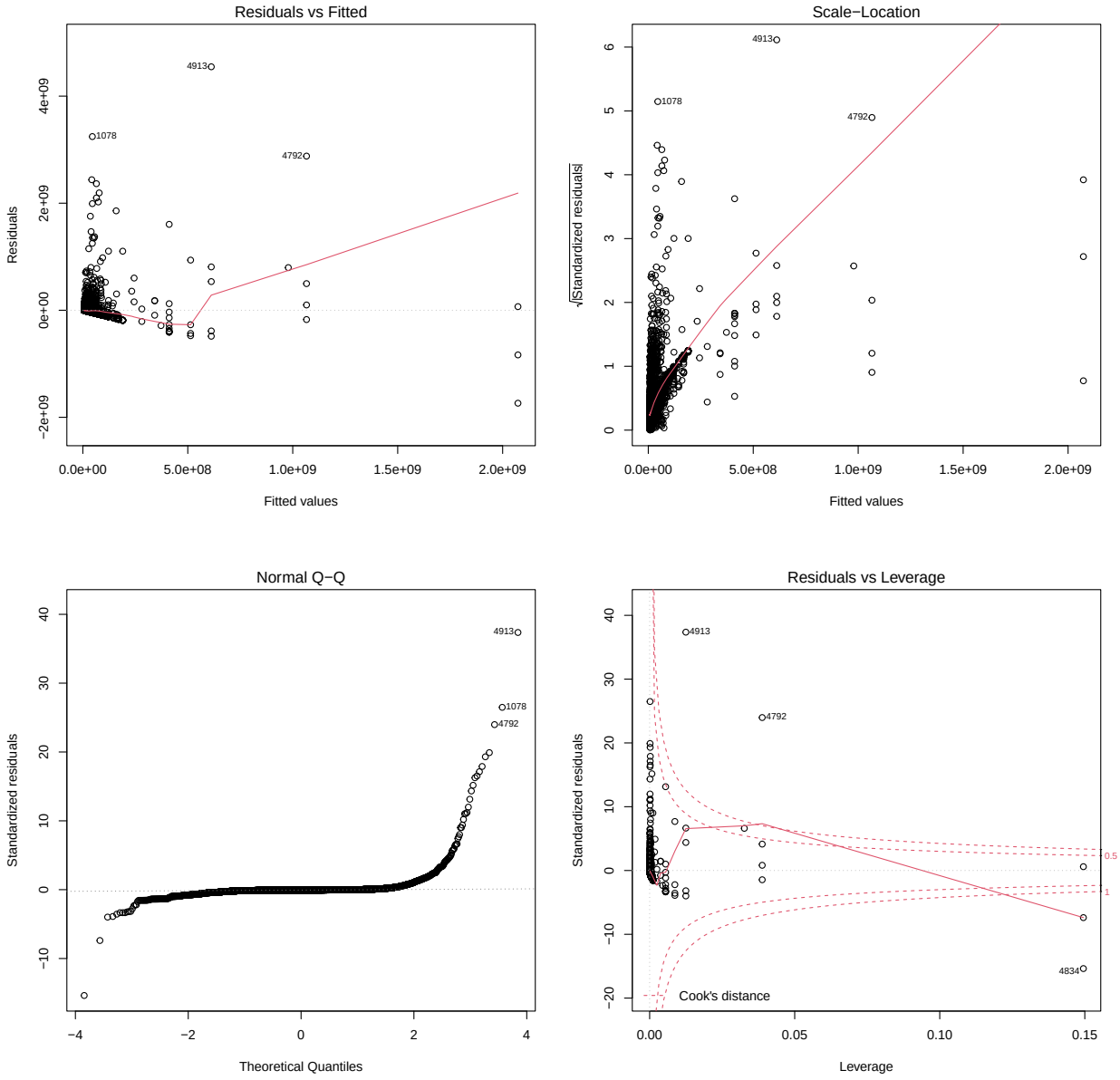
```
confint(reg1, level=0.95)
```

```
##                   2.5 %     97.5 %
## (Intercept) 3765058.66 9232144.03
## res_pop_den   28184.23   30856.06
```

```
anova(reg1)
```

```
## Analysis of Variance Table
##
## Response: TOTAL_OFFICIAL_NEED
##               Df     Sum Sq    Mean Sq F value    Pr(>F)
## res_pop_den    1 2.8154e+19 2.8154e+19  1876.3 < 2.2e-16 ***
## Residuals   8261 1.2396e+20 1.5005e+16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(reg1)
```




```
reg1.resid <- residuals(reg1)
reg1.pred <- predict(reg1)
reg1.zresid <- scale(reg1.resid, scale=T)
reg1.zpred <- scale(reg1.pred, scale=T)
```

B. Regression 2: Need and Income

```
reg2<-lm (TOTAL_OFFICIAL_NEED ~ MEDINC09, data = regress)
summary(reg2)
```

```
##
## Call:
## lm(formula = TOTAL_OFFICIAL_NEED ~ MEDINC09, data = regress)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
##  -56171951  -21428721  -16341482  -10463055 5136540363
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.008e+06  6.022e+06  -1.164    0.245
## MEDINC09     4.926e+02  9.898e+01   4.977 6.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 135500000 on 8261 degrees of freedom
## Multiple R-squared:  0.00299,    Adjusted R-squared:  0.002869
## F-statistic: 24.77 on 1 and 8261 DF,  p-value: 6.582e-07
```

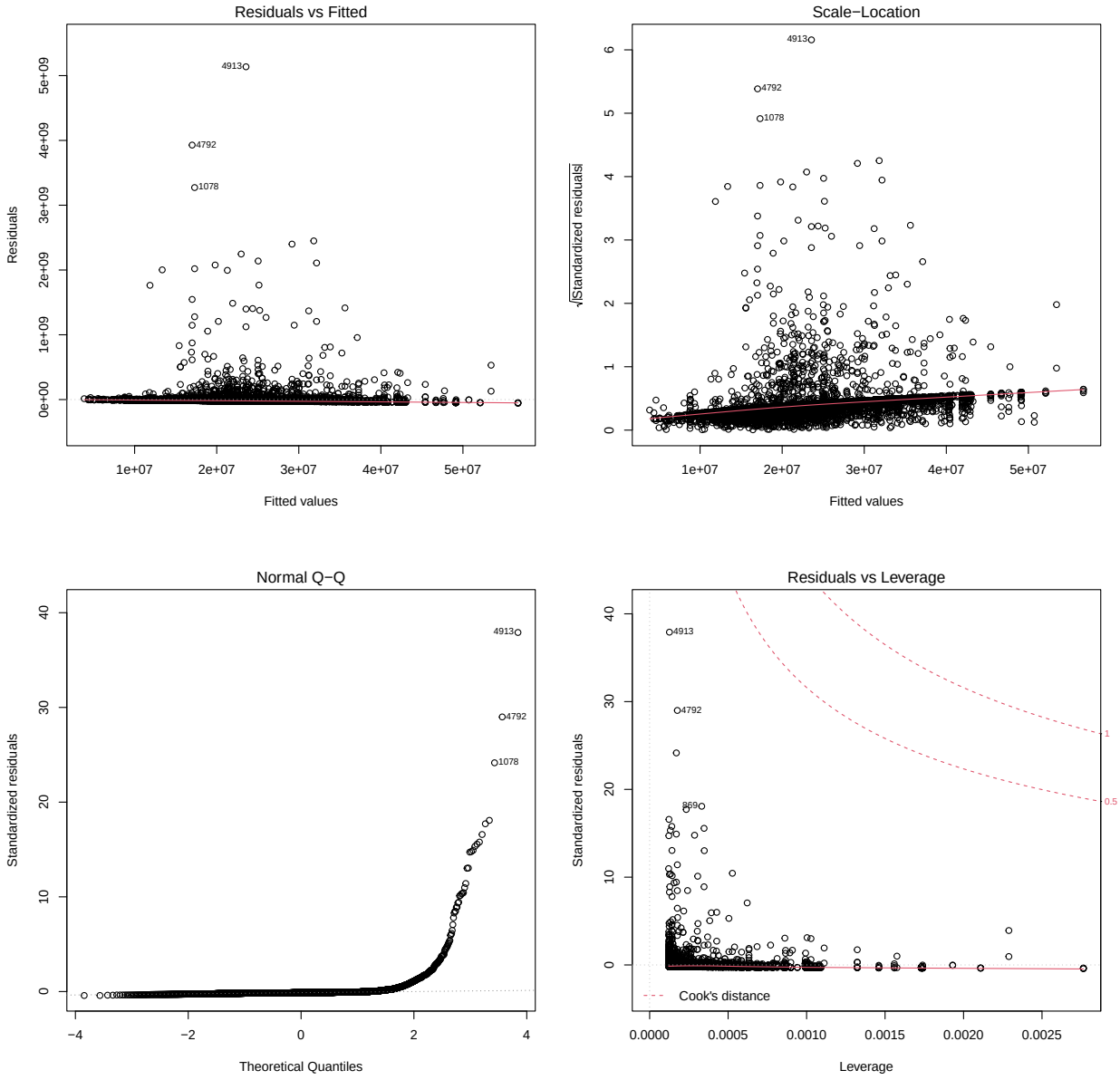
```
confint(reg2, level=0.95)
```

```
##                     2.5 %       97.5 %
## (Intercept) -1.881380e+07 4797292.2045
## MEDINC09     2.986074e+02     686.6481
```

```
anova(reg2)
```

```
## Analysis of Variance Table
##
## Response: TOTAL_OFFICIAL_NEED
##             Df     Sum Sq    Mean Sq F value    Pr(>F)
## MEDINC09     1 4.5478e+17 4.5478e+17  24.772 6.582e-07 ***
## Residuals 8261 1.5166e+20 1.8358e+16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(reg2)
```




```
reg2.resid <- residuals(reg2)
reg2.pred <- predict(reg2)
reg2.zresid <- scale(reg2.resid, scale=T)
reg2.zpred <- scale(reg2.pred, scale=T)
```

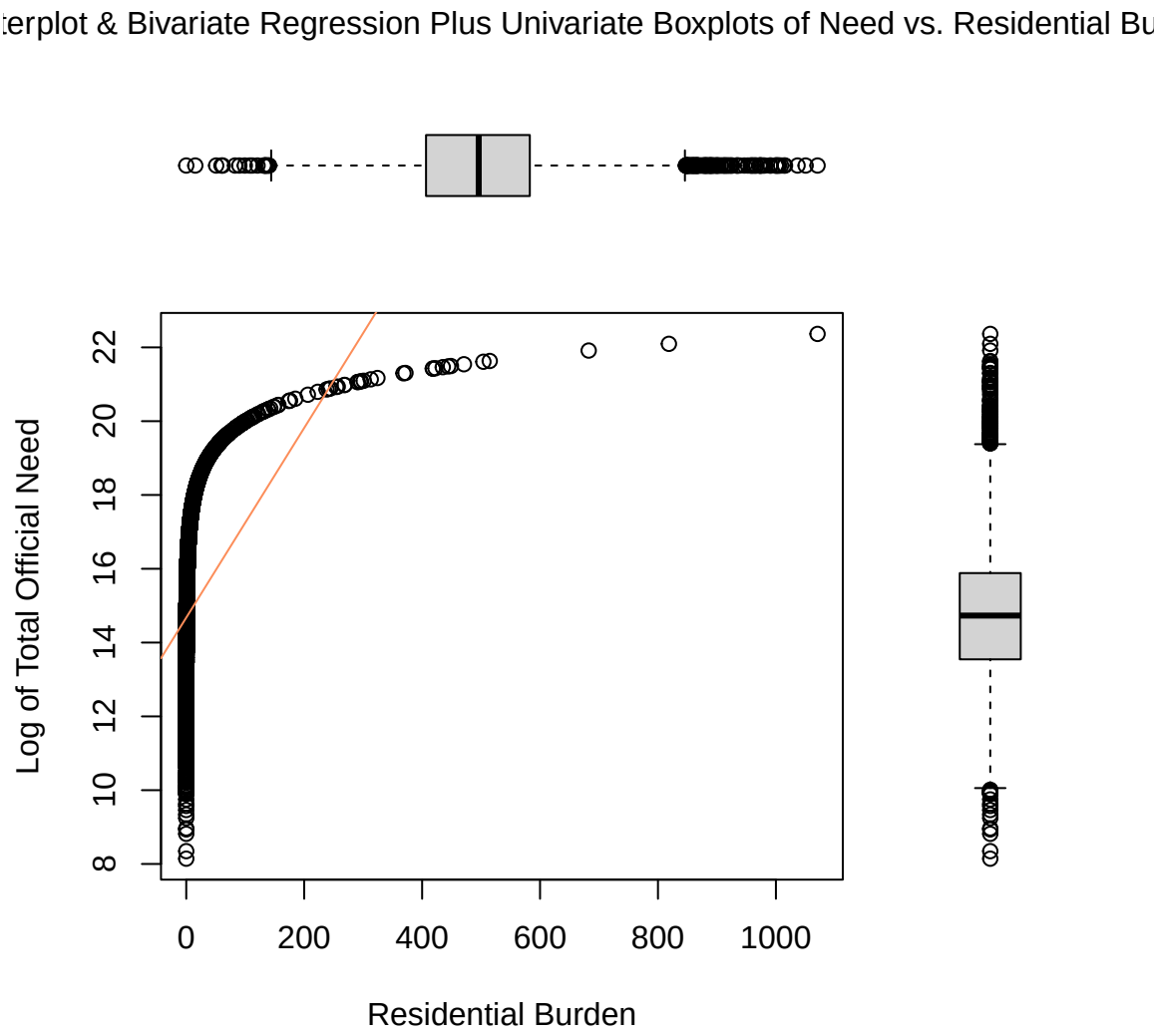
C. Regression 3: need and residential burden

```
cor.test(regress$LOG_NEED, regress$res_burden)
```

```
##
##  Pearson's product-moment correlation
##
## data:  regress$LOG_NEED and regress$res_burden
```

```
## t = 39.183, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3775407 0.4139091
## sample estimates:
##       cor
## 0.3958801
```

```
par(fig=c(0,0.8,0,0.8))

plot(regress$res_burden, regress$LOG_NEED,
     xlab="Residential Burden",
     ylab="Log of Total Official Need")

abline(lm(LOG_NEED ~ res_burden, data = regress),
       col=brewer.pal (n=3,name = "Spectral"))

par(fig=c(0,0.8,0.55,1), new=TRUE)

boxplot(regress$log_res_burden, horizontal=TRUE, axes=FALSE)
par(fig=c(0.65,1,0,0.8),new=TRUE)
boxplot(regress$LOG_NEED,axes=FALSE)
mtext("Scatterplot & Bivariate Regression Plus Univariate Boxplots of Need vs. Residential Burden", sid
```



Regression 3:

```
reg3<-lm (TOTAL_OFFICIAL_NEED ~ res_burden, data = regress)
summary(reg3)
```

```
##
## Call:
## lm(formula = TOTAL_OFFICIAL_NEED ~ res_burden, data = regress)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.293e-05  2.000e-08  3.300e-08  3.800e-08  9.629e-05
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -3.357e-07  1.251e-08 -2.683e+01   <2e-16 ***
```

```
## res_burden   4.820e+06  4.387e-10  1.099e+16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.123e-06 on 8261 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.207e+32 on 1 and 8261 DF,  p-value: < 2.2e-16
```

```
confint(reg3, level=0.95)
```

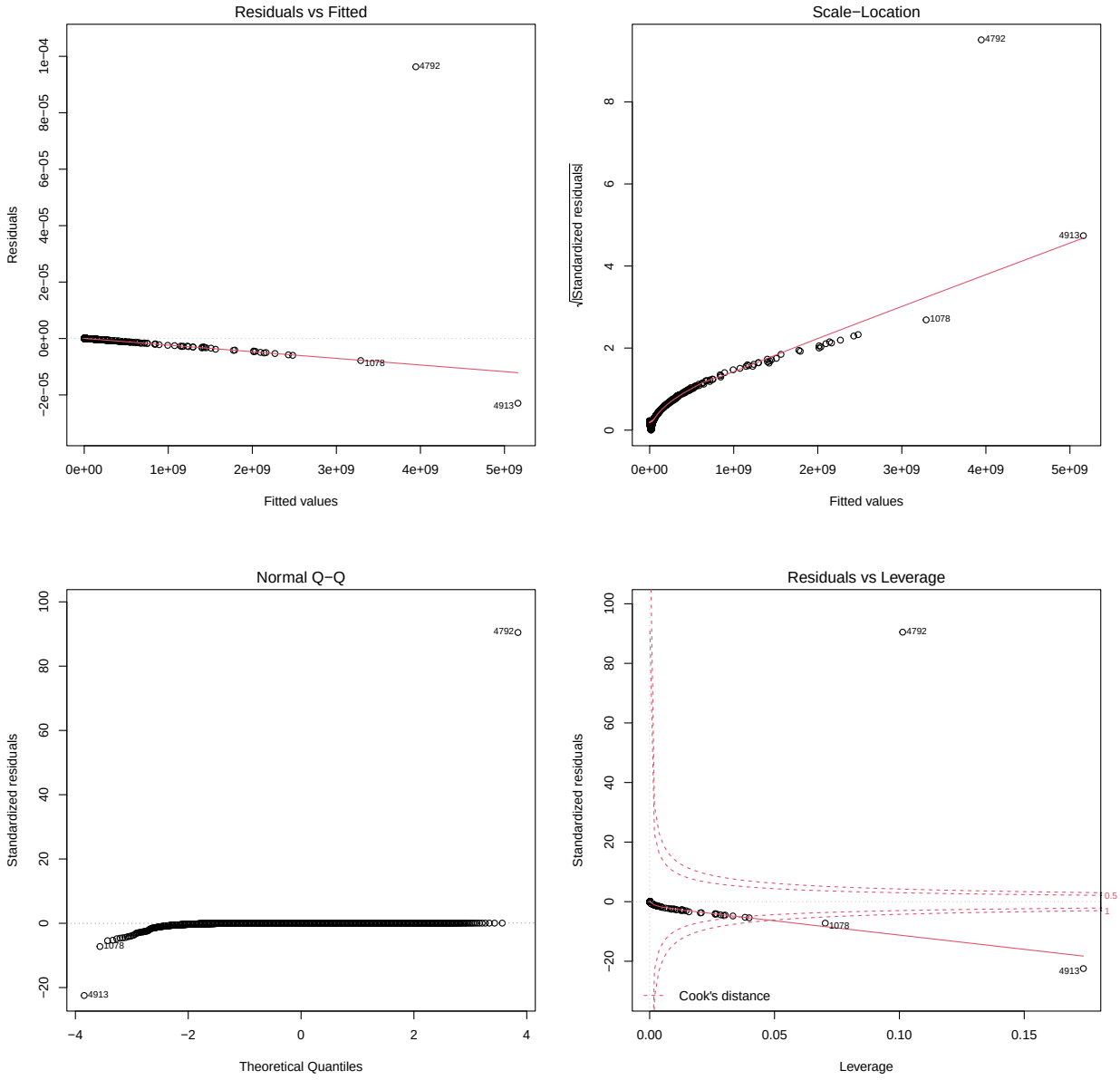
```
##                     2.5 %        97.5 %
## (Intercept) -3.602492e-07 -3.111969e-07
## res_burden   4.819874e+06  4.819874e+06
```

```
anova(reg3)
```

```
## Warning in anova.lm(reg3): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```
## Analysis of Variance Table
##
## Response: TOTAL_OFFICIAL_NEED
##              Df     Sum Sq    Mean Sq   F value    Pr(>F)
## res_burden    1 1.5211e+20 1.5211e+20 1.207e+32 < 2.2e-16 ***
## Residuals  8261 0.0000e+00 0.0000e+00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(reg3)
```




```
reg3.resid <- residuals(reg3)
reg3.pred <- predict(reg3)
reg3.zresid <- scale(reg3.resid, scale=T)
reg3.zpred <- scale(reg3.pred, scale=T)
```

D. Regression 4:

```
reg4<-lm (TOTAL_OFFICIAL_NEED ~ dens_city, data = regress)
summary(reg4)
```

```
##
## Call:
## lm(formula = TOTAL_OFFICIAL_NEED ~ dens_city, data = regress)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -834779543  -13797869   -7420550   -3246693 4630415745
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8066838    1458276   5.532 3.27e-08 ***
## dens_city     167167       4934  33.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127200000 on 8261 degrees of freedom
## Multiple R-squared:  0.122,  Adjusted R-squared:  0.1219
## F-statistic:  1148 on 1 and 8261 DF,  p-value: < 2.2e-16
```

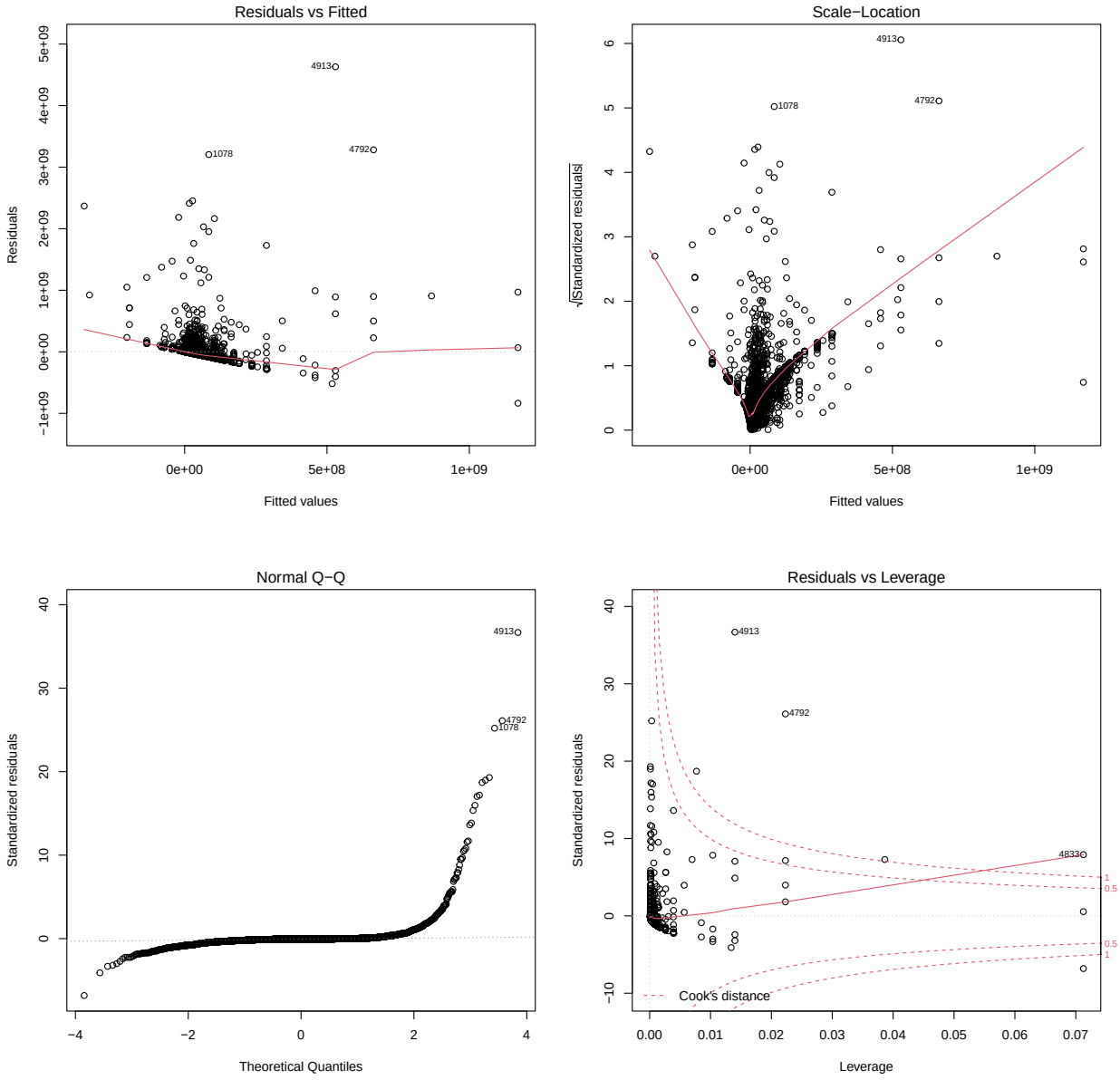
```
confint(reg4, level=0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 5208250.7 10925424.6
## dens_city    157494.3   176839.8
```

```
anova(reg4)
```

```
## Analysis of Variance Table
##
## Response: TOTAL_OFFICIAL_NEED
##             Df     Sum Sq    Mean Sq F value    Pr(>F)
## dens_city    1 1.8555e+19 1.8555e+19  1147.7 < 2.2e-16 ***
## Residuals 8261 1.3356e+20 1.6167e+16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
layout(matrix(c(1,2,3,4),2,2))
plot(reg4)
```




```
reg4.resid <- residuals(reg4)
reg4.pred <- predict(reg4)
reg4.zresid <- scale(reg4.resid, scale=T)
reg4.zpred <- scale(reg4.pred, scale=T)
```

5.0 Multivariate Regressions Building multivariate regression models for each measure of need. The chosen variables are those that have coincided with the statistical models as seen in the code above. We began the process by sorting through the data and creating scatter plots for need vs. other independent variables.

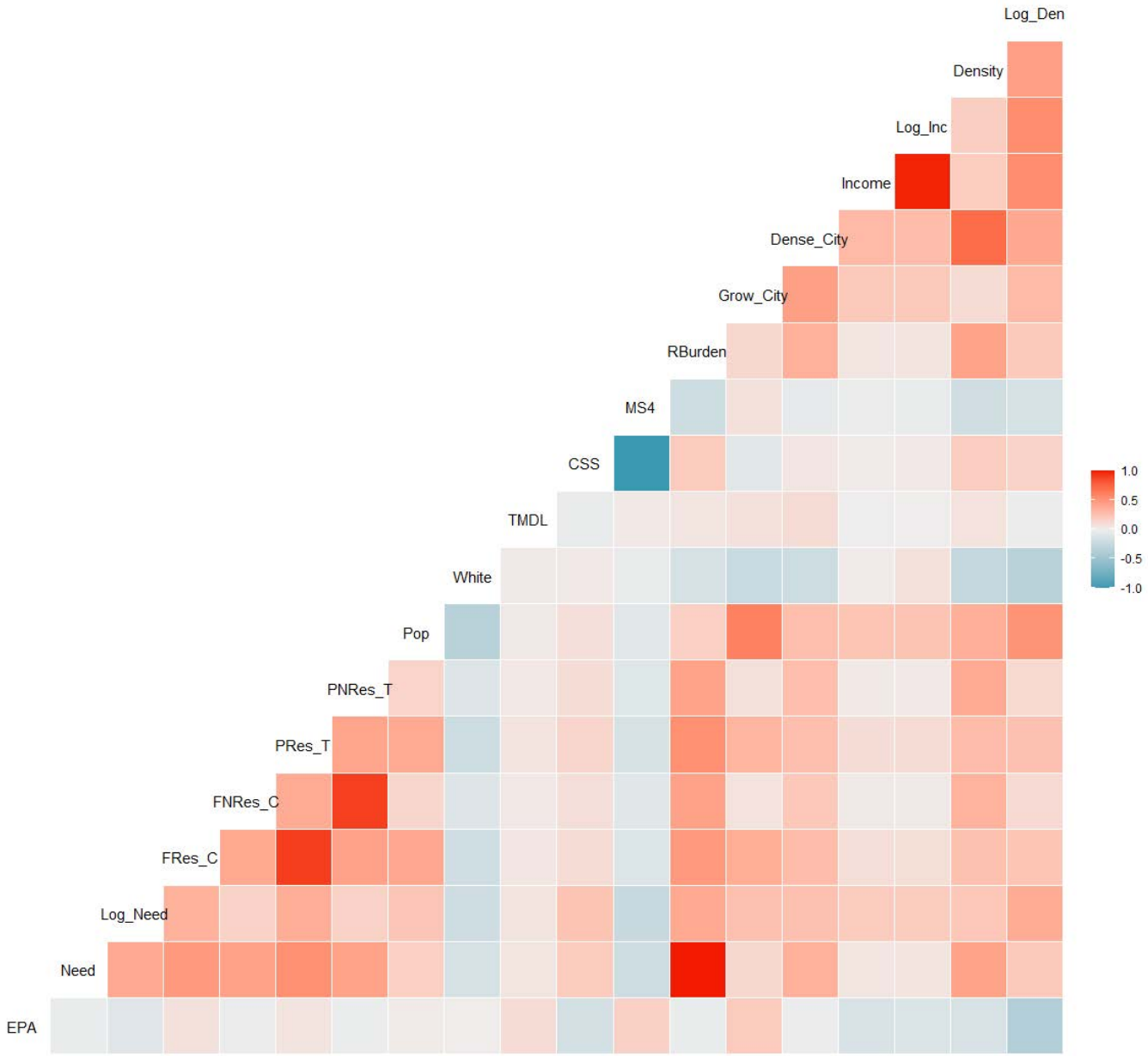
We have included two sets of scatter plots and regressions- one comapring Philadelphia to the US and the other comparing Philly to the state of Pennsylvania. We wanted to observe the differences that a larger dataset would have on the regression results as well as the variables.

A. Clearing the environment and getting started

```
rm(least_need, most_need, PA_need, Philly_need,
   CSS, CSS_MS4, MS4, reg1, corr_vars_plot,
   EPAsum, ass_vars, corr_vars)

glimpse(PA_regress)

regress_vars_plot <- regress %>%
  summarise(EPA = (EPA_REGION),
            Need = (TOTAL_OFFICIAL_NEED),
            Log_Need = (LOG_NEED),
            FRes_C = (PROJ_RES_REC_COLLCTN),
            FNRes_C = (PROJ_N_RES_REC_COLLCTN),
            PRes_T = (PRES_RES_REC_TRMT),
            PNRes_T = (PRES_N_RES_REC_TRTM),
            Pop = (POP00),
            White = (PCTWHITE10),
            TMDL = (TMDL),
            CSS = (CSS),
            MS4 = (MS4),
            RBurden= (res_burden),
            Grow_City = (grow_city),
            Dense_City = (dens_city),
            Income = (MEDINC09),
            Log_Inc =(log_inc),
            Density = (res_pop_den),
            Log_Den = (log_res_pop_den))
```

Correlation Matrix

```
plot.new(); dev.off()
```




```
GGally::ggcorr(regress_vars_plot)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

5.2 Scatterplots for Need vs. Other Variables

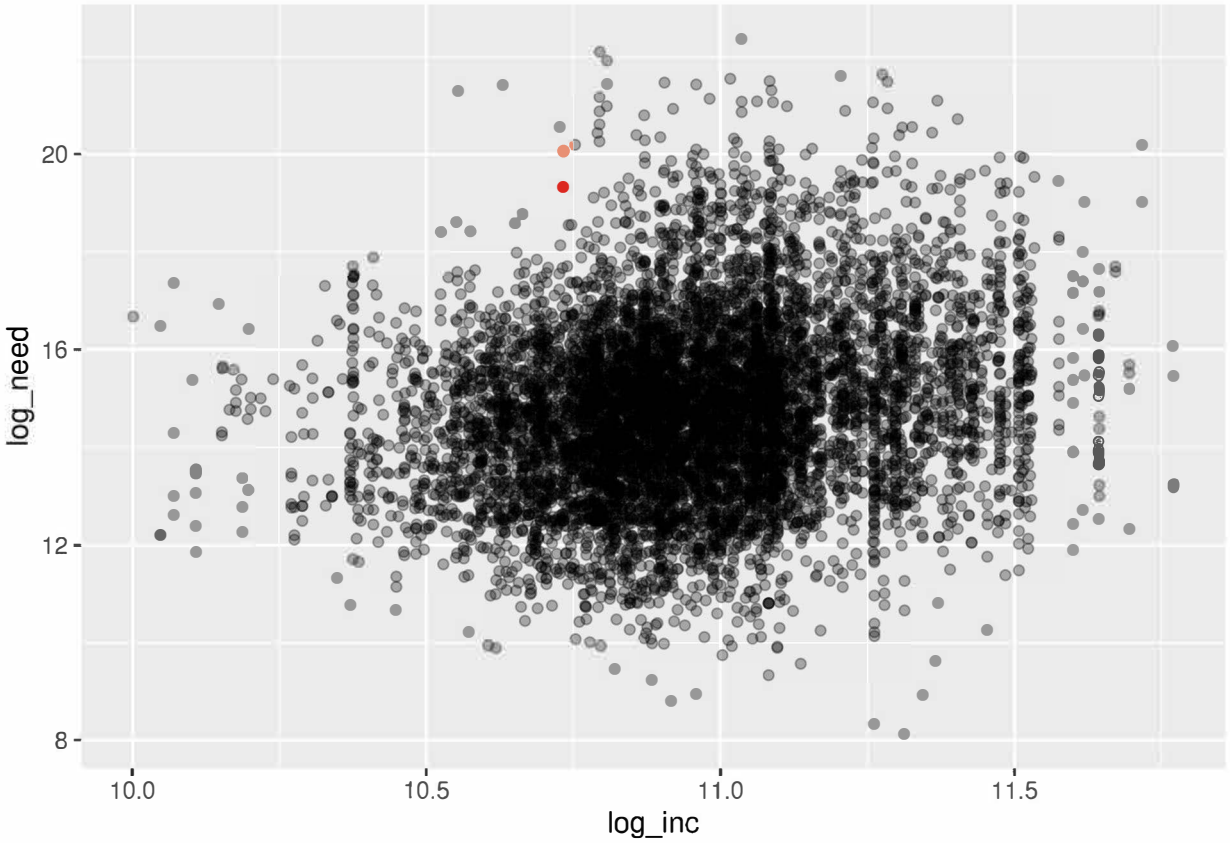
5.2.1 Need vs. Residential Burden

```
reg4.zpred <- scale(reg1.pred, scale=T)

#plot.new()
#par(mfrow=c(1, 1))
#plotNormalHistogram(reg4.zresid, breaks = 100, )
```

6.1 Multivariate Regressions

6.1.1 Need X Income - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=log_inc, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_inc, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```



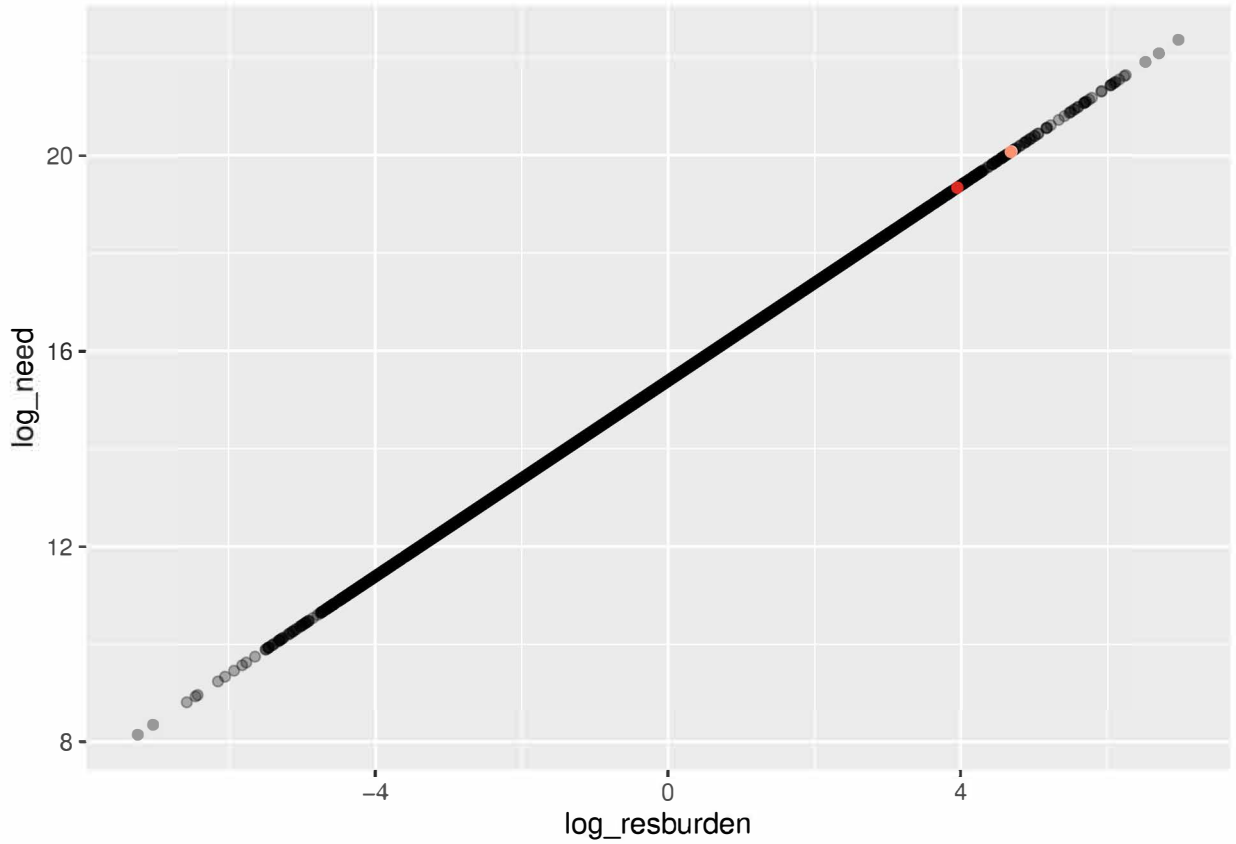

```
cor.test(Regress$log_inc, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$log_inc and Regress$log_need
## t = 18.058, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.1740362 0.2155236
## sample estimates:
##       cor
## 0.1948671
```

6.1.2 Need X Residential Burden - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=log_resburden, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_resburden, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```



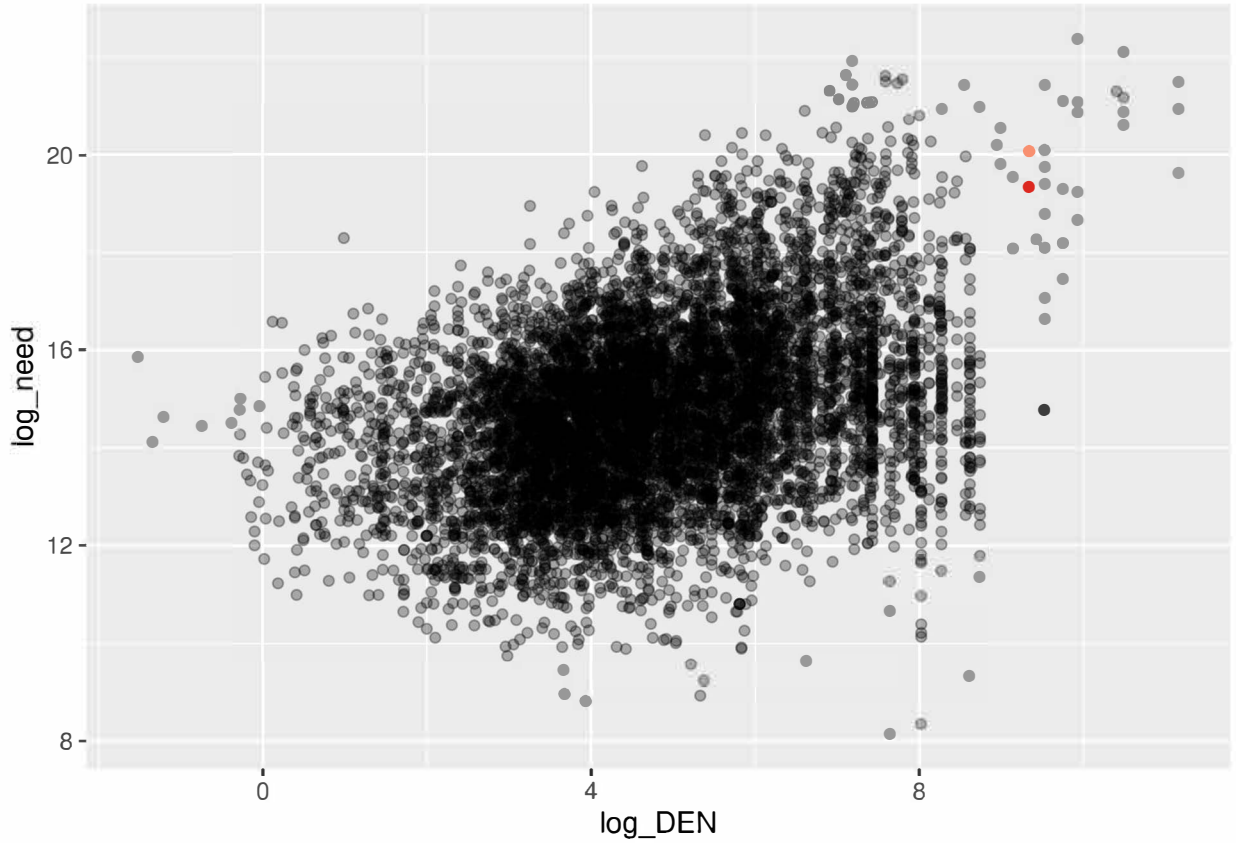

```
cor.test(Regress$log_resburden, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$log_resburden and Regress$log_need
## t = 6099527565, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  1 1
## sample estimates:
```

```
## cor
##   1
```

6.1.3 Need X Pop Density - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=log_DEN, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_DEN, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```

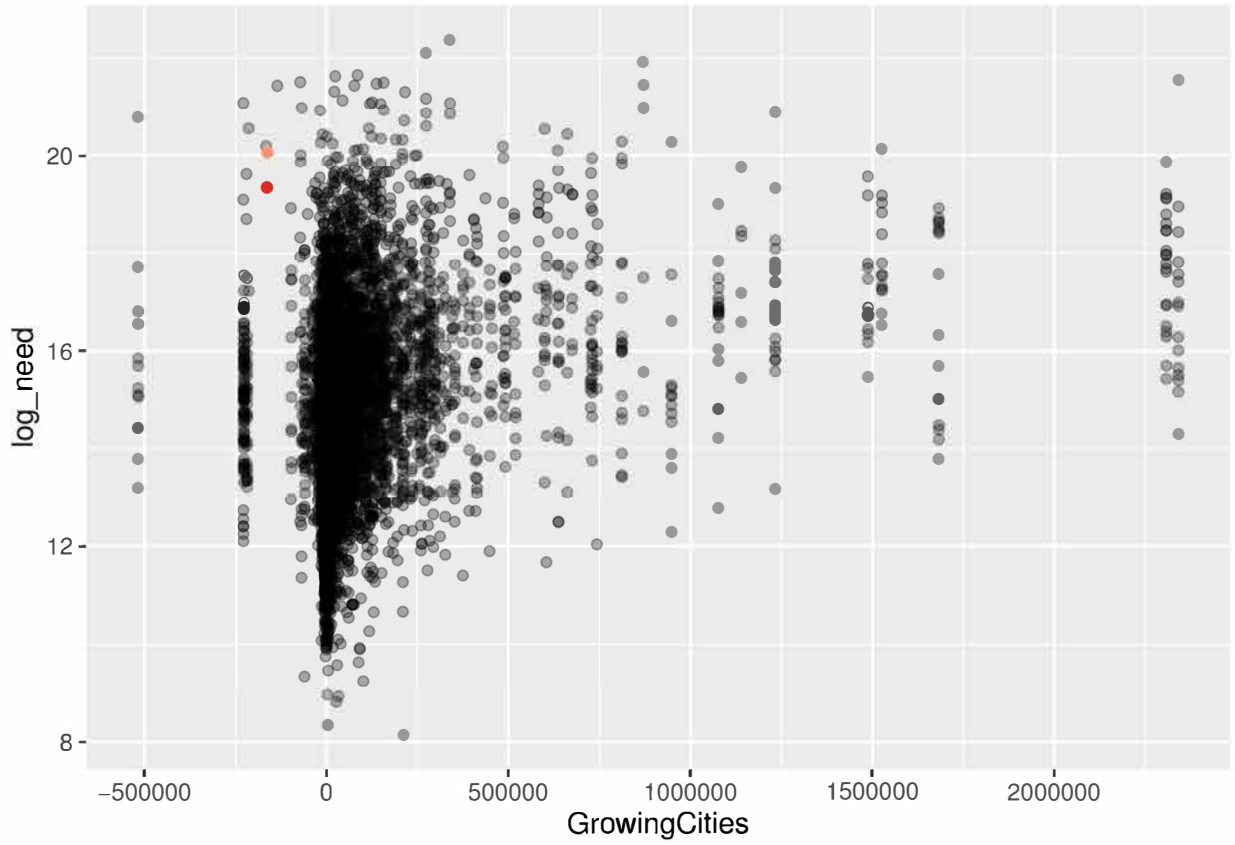



```
cor.test(Regress$log_DEN, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$log_DEN and Regress$log_need
## t = 37.449, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3623722 0.3992403
## sample estimates:
##       cor
## 0.3809577
```

6.1.4 Need X Growing Cities - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=GrowingCities, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=GrowingCities, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```

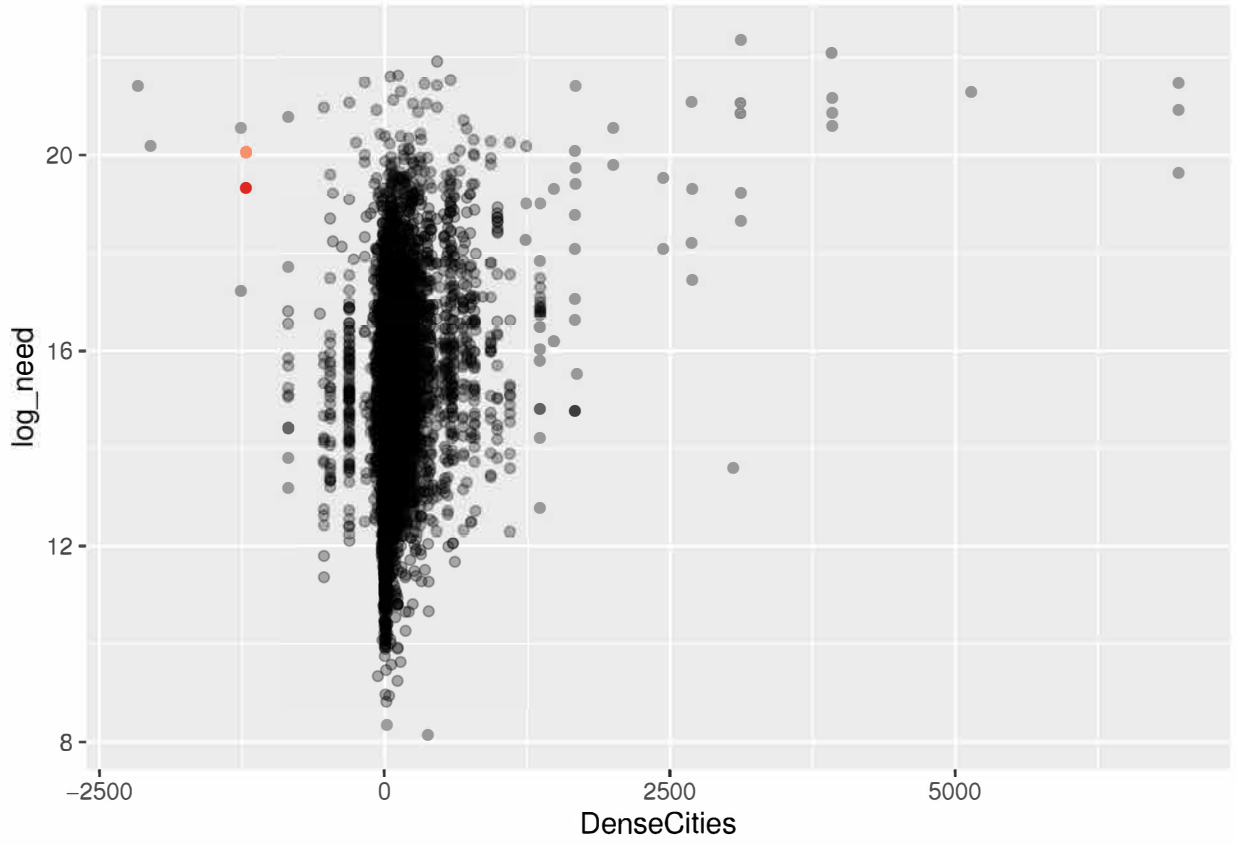



```
cor.test(Regress$GrowingCities, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$GrowingCities and Regress$log_need
## t = 24.459, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2396438 0.2798571
## sample estimates:
##       cor
## 0.2598632
```

6.1.5 Need X Dense Cities - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=DenseCities, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=DenseCities, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```

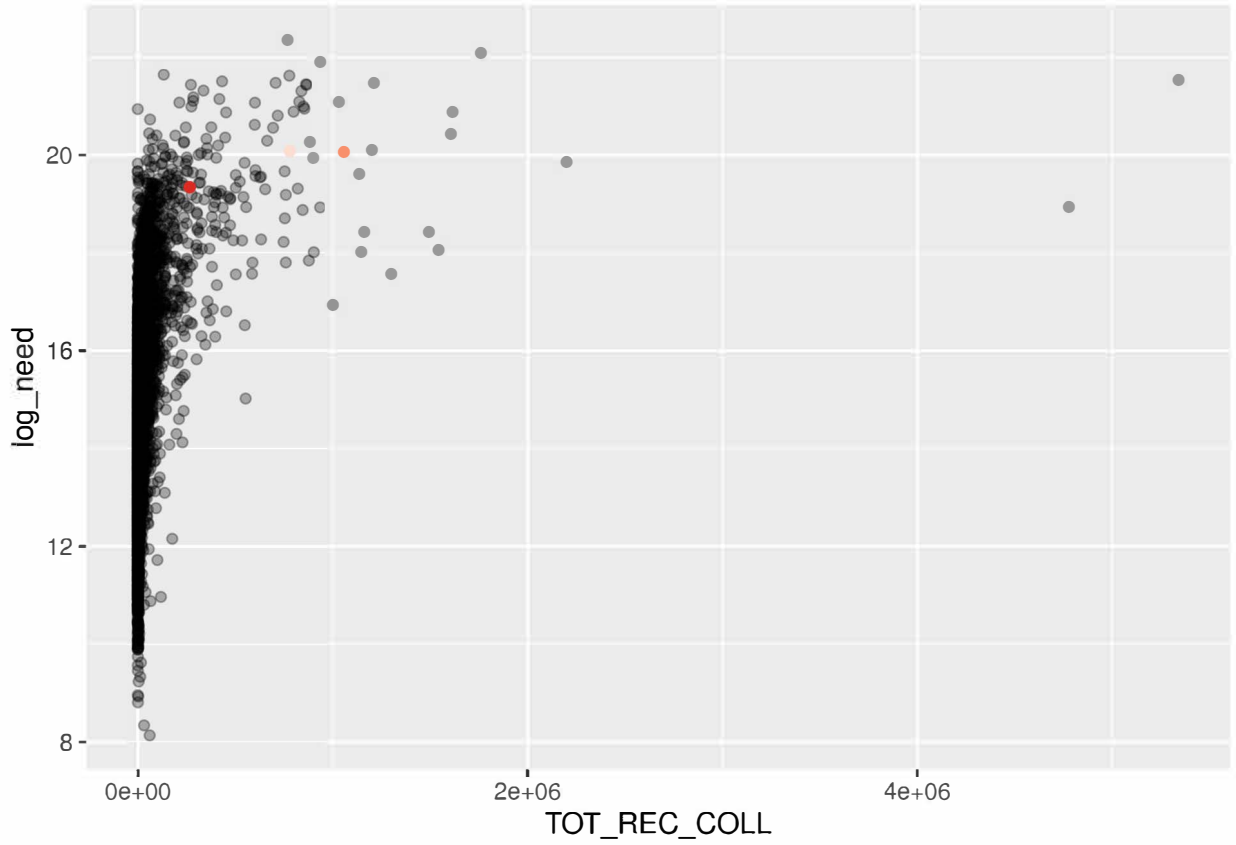



```
cor.test(Regress$DenseCities, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$DenseCities and Regress$log_need
## t = 24.596, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2410133 0.2811962
## sample estimates:
##       cor
## 0.2612179
```

6.1.6 Need X Total Residence Receiving Collection - Plots comparing Philadelphia to the US

```
ggplot(Regress, aes(x=TOT_REC_COLL, y=log_need))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=TOT_REC_COLL, y=log_need),
             col=brewer.pal (n=3,name = "Reds"))
```

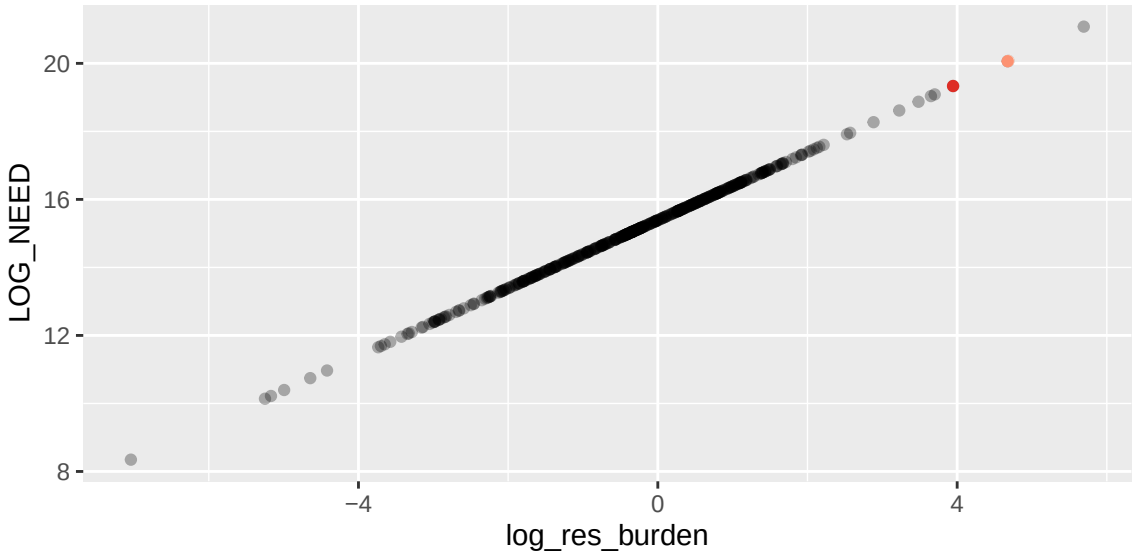



```
cor.test(Regress$TOT_REC_COLL, Regress$log_need)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Regress$TOT_REC_COLL and Regress$log_need
## t = 33.697, df = 8261, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3285193 0.3664345
## sample estimates:
##      cor
## 0.347619
```

Need and residential burden are perfectly correlated, with the cor.test yeilding a result of 1. This can be seen in t he scatter plot given below. This high correlation can lead to multicollinearity between the variables in a multivariate regression. To avoid skewing of the results, we have chosen to omit residential burden from the forward and backward selections.

```
ggplot(PA_regress, aes(x=log_res_burden, y=LOG_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_res_burden, y=LOG_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```




```
cor (PA_regress$log_res_burden, PA_regress$LOG_NEED)
```

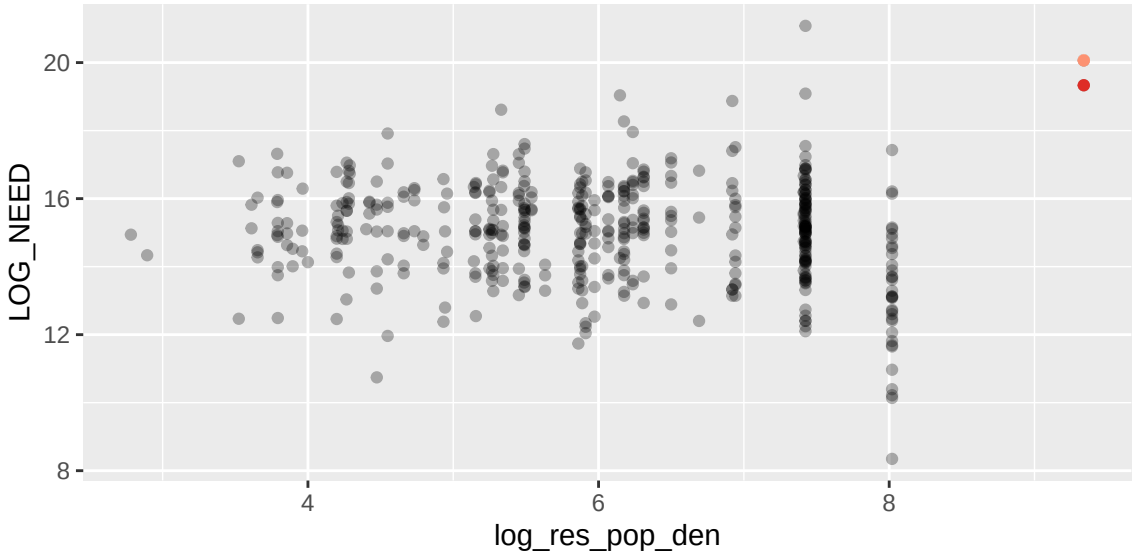
```
## [1] 1
```

```
cor.test (PA_regress$log_res_burden, PA_regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PA_regress$log_res_burden and PA_regress$LOG_NEED
## t = 1021069691, df = 463, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  1 1
## sample estimates:
## cor
##   1
```

5.2.2 Need vs. Population Density

```
ggplot(PA_regress, aes(x=log_res_pop_den, y=LOG_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_res_pop_den, y=LOG_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```




```
cor (PA_regress$log_res_pop_den, PA_regress$LOG_NEED)
```

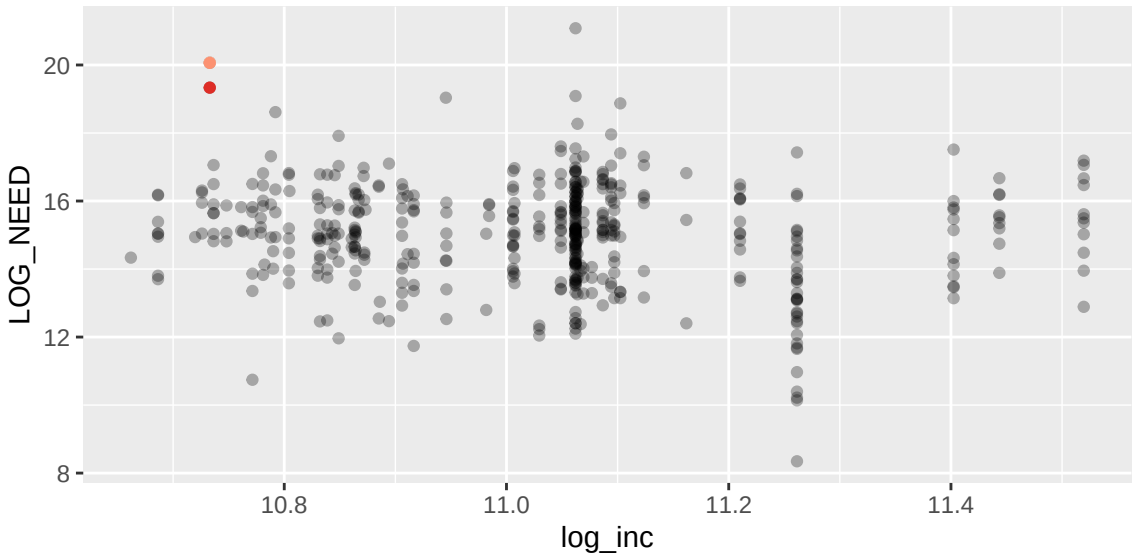
```
## [1] -0.08882227
```

```
cor.test (PA_regress$log_res_pop_den, PA_regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PA_regress$log_res_pop_den and PA_regress$LOG_NEED
## t = -1.9188, df = 463, p-value = 0.05562
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.178315899  0.002128812
## sample estimates:
##         cor
## -0.08882227
```

5.2.3 Need vs. Income

```
ggplot(PA_regress, aes(x=log_inc, y=LOG_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_inc, y=LOG_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```




```
cor (PA_regress$log_inc, PA_regress$LOG_NEED)
```

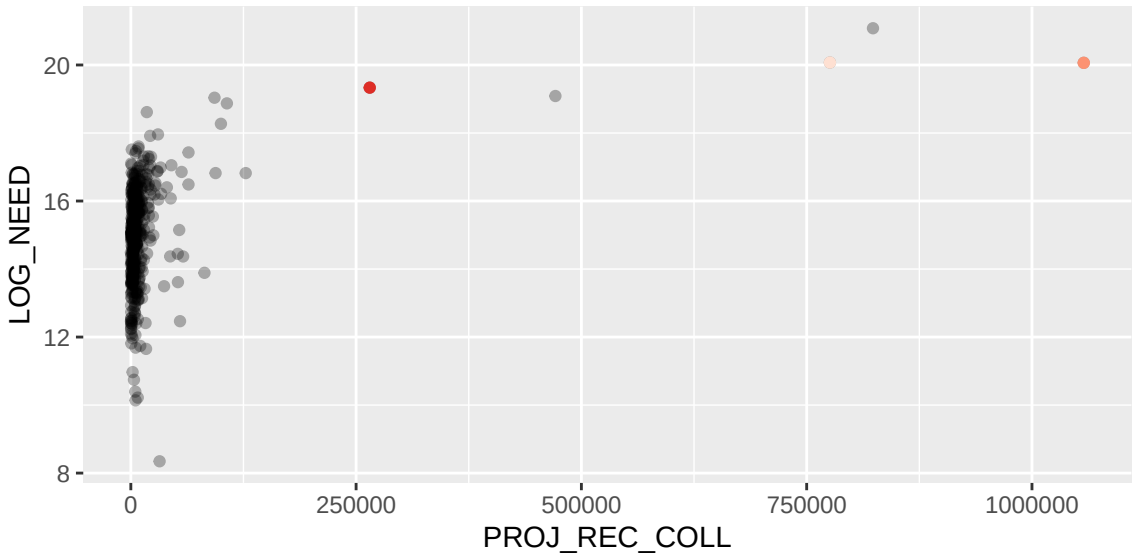
```
## [1] -0.1378445
```

```
cor.test (PA_regress$log_inc, PA_regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PA_regress$log_inc and PA_regress$LOG_NEED
## t = -2.9946, df = 463, p-value = 0.002895
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.22594618 -0.04750607
## sample estimates:
##        cor
## -0.1378445
```

5.2.4 Need vs. Projected Residences & Non-residences Receiving Collection

```
ggplot(PA_regress, aes(x=PROJ_REC_COLL, y=LOG_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=PROJ_REC_COLL, y=LOG_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```




```
cor (PA_regress$PROJ_REC_COLL, PA_regress$LOG_NEED)
```

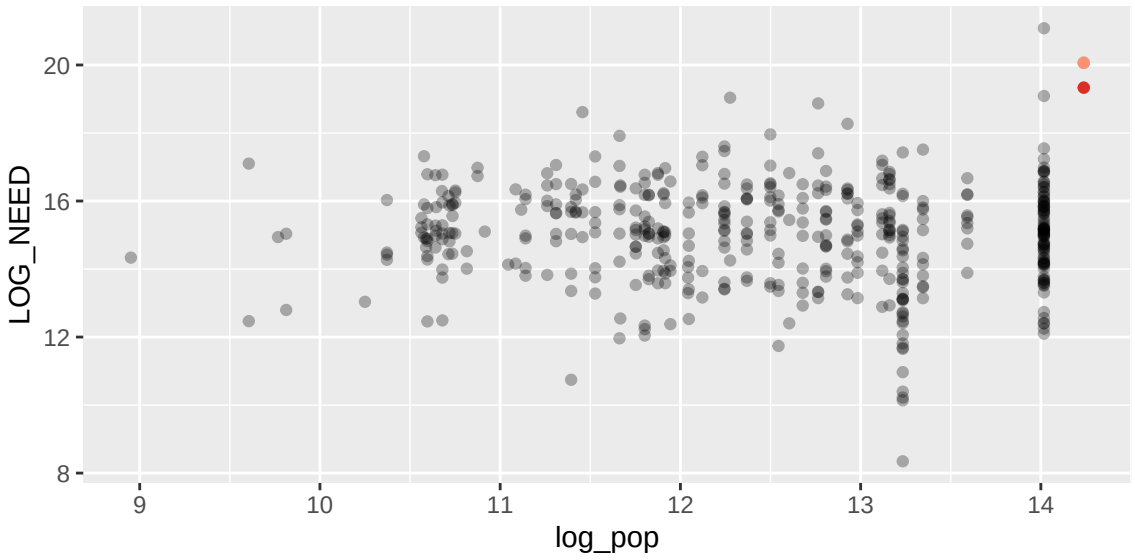
```
## [1] 0.3662297
```

```
cor.test (PA_regress$PROJ_REC_COLL, PA_regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PA_regress$PROJ_REC_COLL and PA_regress$LOG_NEED
## t = 8.4687, df = 463, p-value = 3.314e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2847797 0.4424295
## sample estimates:
##       cor
## 0.3662297
```

5.2.5 Need vs. Population

```
ggplot(PA_regress, aes(x=log_pop, y=LOG_NEED))+
  geom_point(alpha=0.3)+
  geom_point(Philly_regress,
             mapping =  aes(x=log_pop, y=LOG_NEED),
             col=brewer.pal (n=3,name = "Reds"))
```




```
cor (PA_regress$log_pop, PA_regress$LOG_NEED)
```

```
## [1] -0.01933858
```

```
cor.test (PA_regress$log_pop, PA_regress$LOG_NEED)
```

```
##
##  Pearson's product-moment correlation
##
## data:  PA_regress$log_pop and PA_regress$LOG_NEED
## t = -0.41619, df = 463, p-value = 0.6775
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.11007889  0.07172142
## sample estimates:
##         cor
## -0.01933858
```

5.3 Some Steps for Regressions

5.3.1 Model 1

```
mod_1 <- lm (LOG_NEED ~ log_res_burden + PROJ_REC_COLL,
                    data = PA_regress)
```

VIF test results

log_res_burden PROJ_REC_COLL 1.1549 1.1549

5.3.2 Model 2

```
mod_2 <- lm (LOG_NEED ~ log_res_burden + PROJ_REC_COLL + log_inc,
             data = PA_regress)
```

VIF test results

log_res_burden PROJ_REC_COLL log_inc 1.172677 1.155156 1.019595

5.3.3 Model 3

```
mod_3 <- lm (LOG_NEED ~ PROJ_REC_COLL +
               log_inc + log_res_pop_den,
             data = PA_regress)
```

VIF test results

PROJ_REC_COLL log_res_pop_den 1.048578 1.048578

5.3.4 Model 4

```
mod_4 <- lm (LOG_NEED ~ log_res_burden + PROJ_REC_COLL + log_res_pop_den,
             data = PA_regress)
```

VIF test results

PROJ_REC_COLL log_res_pop_den 1.048578 1.048578

Anova Models to predict Statistical Significance

```
anova(mod_1, mod_2, mod_3)
```

```
## Analysis of Variance Table
##
## Model 1: LOG_NEED ~ log_res_burden + PROJ_REC_COLL
## Model 2: LOG_NEED ~ log_res_burden + PROJ_REC_COLL + log_inc
## Model 3: LOG_NEED ~ PROJ_REC_COLL + log_inc + log_res_pop_den
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    462   0.00
## 2    461   0.00  1      0.00 0.9548  0.329
## 3    461 882.38  0   -882.38
```

```
anova(mod_4)
```

```
## Warning in anova.lm(mod_4): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```
## Analysis of Variance Table
##
## Response: LOG_NEED
##                  Df Sum Sq Mean Sq    F value Pr(>F)
## log_res_burden    1   1055    1055 4.5320e+30 <2e-16 ***
## PROJ_REC_COLL     1      0       0 1.2627e+02 <2e-16 ***
## log_res_pop_den   1      0       0 2.0242e+00 0.1555
## Residuals       461      0       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.4 Backward Regressions

We have chosen the backward regression as our preferred model because of the lack of supressor effects compared to forward regressions. Median income and cities experiencing population growth over the decade were the two variables dropped in this regression. Interestingly, income was kept in the regression done on the US dataset, meaning that it has a greater influence on need when compared to PA and Philly.

```
step( lm ( LOG_NEED ~ CSS + log_res_pop_den + PROJ_REC_COLL +
             log_inc + log_pop + grow_city +
             ALAND + MEDINC09, data = PA_regress),
      direction="backward")
```

```
## Start:  AIC=260.36
## LOG_NEED ~ CSS + log_res_pop_den + PROJ_REC_COLL + log_inc +
##     log_pop + grow_city + ALAND + MEDINC09
##
##                   Df Sum of Sq    RSS    AIC
## - MEDINC09         1     0.180 783.26 258.46
## - log_inc          1     0.376 783.45 258.58
## <none>                         783.08 260.36
## - grow_city        1     4.938 788.02 261.28
## - CSS              1    17.368 800.45 268.56
## - ALAND            1    31.730 814.81 276.83
## - log_pop          1    54.347 837.42 289.56
## - log_res_pop_den  1    58.879 841.96 292.07
## - PROJ_REC_COLL    1   149.462 932.54 339.58
##
## Step:  AIC=258.46
## LOG_NEED ~ CSS + log_res_pop_den + PROJ_REC_COLL + log_inc +
##     log_pop + grow_city + ALAND
##
##                   Df Sum of Sq    RSS    AIC
## - log_inc          1     0.774 784.03 256.92
## <none>                         783.26 258.46
## - grow_city        1     5.414 788.67 259.67
## - CSS              1    17.674 800.93 266.84
## - ALAND            1    31.693 814.95 274.91
## - log_pop          1    54.222 837.48 287.59
## - log_res_pop_den  1    59.005 842.26 290.24
## - PROJ_REC_COLL    1   153.831 937.09 339.84
##
## Step:  AIC=256.92
## LOG_NEED ~ CSS + log_res_pop_den + PROJ_REC_COLL + log_pop +
##     grow_city + ALAND
##
##                   Df Sum of Sq    RSS    AIC
## <none>                         784.03 256.92
## - grow_city        1     5.723 789.75 258.30
## - CSS              1    18.184 802.22 265.58
## - ALAND            1    31.002 815.03 272.95
## - log_pop          1    54.429 838.46 286.13
## - log_res_pop_den  1    58.389 842.42 288.32
## - PROJ_REC_COLL    1   173.043 957.07 347.66

##
```

```
## Call:
## lm(formula = LOG_NEED ~ CSS + log_res_pop_den + PROJ_REC_COLL +
##     log_pop + grow_city + ALAND, data = PA_regress)
##
## Coefficients:
##     (Intercept)              CSS  log_res_pop_den    PROJ_REC_COLL
##       1.202e+00        5.377e-01       -2.651e+00        8.598e-06
##         log_pop        grow_city            ALAND
##       2.578e+00        1.093e-06       -1.419e-09
```

5.4 Forward Regressions

```
fullmod<-lm(LOG_NEED ~ log_res_pop_den + log_inc +
              PROJ_REC_COLL + log_pop + grow_city,
            data = PA_regress)

intonly<-lm(LOG_NEED ~1, data = PA_regress)

step(intonly, scope=list(lower=intonly, upper=fullmod), direction="forward")
```

```
## Start:  AIC=382.98
## LOG_NEED ~ 1
##
##                   Df Sum of Sq     RSS    AIC
## + PROJ_REC_COLL    1   141.508  913.54 318.01
## + log_inc          1    20.047 1035.00 376.06
## + log_res_pop_den  1     8.324 1046.73 381.29
## <none>                         1055.05 382.98
## + log_pop          1     0.395 1054.66 384.80
## + grow_city        1     0.311 1054.74 384.84
##
## Step:  AIC=318.01
## LOG_NEED ~ PROJ_REC_COLL
##
##                   Df Sum of Sq    RSS    AIC
## + log_res_pop_den  1   31.0938 882.45 303.91
## + log_inc          1   13.8486 899.69 312.91
## + log_pop          1    6.2472 907.30 316.82
## <none>                         913.54 318.01
## + grow_city        1    0.7800 912.76 319.61
##
## Step:  AIC=303.91
## LOG_NEED ~ PROJ_REC_COLL + log_res_pop_den
##
##             Df Sum of Sq    RSS    AIC
## + log_pop    1    48.317 834.13 279.72
## <none>                   882.45 303.91
## + grow_city  1     0.800 881.65 305.49
## + log_inc    1     0.070 882.38 305.87
##
## Step:  AIC=279.72
## LOG_NEED ~ PROJ_REC_COLL + log_res_pop_den + log_pop
##
```

```
##             Df Sum of Sq    RSS    AIC
## <none>                    834.13 279.72
## + grow_city  1   0.88817 833.24 281.23
## + log_inc    1   0.47163 833.66 281.46
```

```
##
## Call:
## lm(formula = LOG_NEED ~ PROJ_REC_COLL + log_res_pop_den + log_pop,
##     data = PA_regress)
##
## Coefficients:
##     (Intercept)    PROJ_REC_COLL  log_res_pop_den          log_pop
##      10.6041660        0.0000085       -0.8370650        0.7513127
```

In conclusion, we reaffirmed these findings through our multivariate regression models which showed that independent variables such as population, population density and the number of residences with access to water infrastructure have a greater impact on official reported need. Other variables such as median income and cities with growing population were dropped from the multivariate regression model through automated variable selection techniques. Due to its high correlation to need, we excluded residential burden from the multivariate regression models to avoid multicollinearity1 in the results.