

# 7: Lab - Generalized Linear Models

Environmental Data Analytics | John Fay and Luana Lima | Developed by Kateri Salk

Fall 2025

## Objectives

1. Answer questions on M5/A5
2. Answer questions on M6 - GLMs
3. Practice more application GLM to real datasets

## Set up

```
library(tidyverse)
library(agricolae)
library(here)
here()

## [1] "/home/guest/EDE_Fall2025_s1"

EPAair <- read.csv(here("Data/Processed_KEY/EPAair_03_PM25_NC1819_ProCESSED.csv"), stringsAsFactors = TRUE)
# Set date to date format
EPAair$date <- as.Date(EPAair$date, format = "%Y-%m-%d")

Litter <- read.csv(here("Data/Processed_KEY/NEON_NIWO_Litter_mass_trap_ProCESSED.csv"), stringsAsFactors = TRUE)
# Set date to date format
Litter$collectDate <- as.Date(Litter$collectDate , format = "%Y-%m-%d")

# Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

## Visualization and interpretation challenge

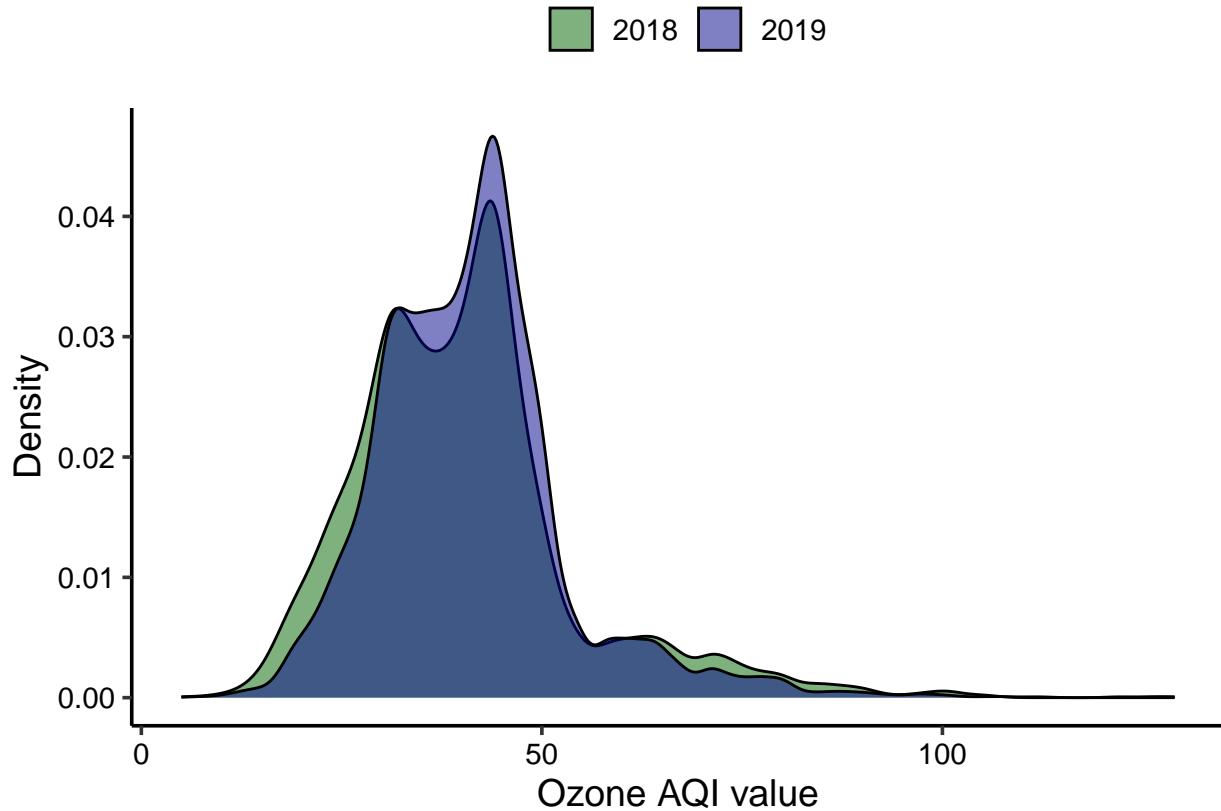
Create three plots, each with appropriately formatted axes and legends. Choose a non-default color palette.

1. geom\_density of ozone divided by year (distinguish between years by adding transparency to the geom\_density layer).
2. geom\_boxplot of ozone divided by year. Add letters representing a significant difference between 2018 and 2019 (hint: stat\_summary).

3. geom\_violin of ozone divided by year, with the 0.5 quantile marked as a horizontal line. Add letters representing a significant difference between 2018 and 2019.

```
#Exercise 1:  
03.density <-  
  ggplot(EPAair, aes(x = Ozone,  
                     fill = as.factor(Year))) +  
  geom_density(alpha = 0.5) +  
  scale_fill_manual(values = c("darkgreen", "darkblue")) +  
  labs(x = "Ozone AQI value",  
       y = "Density",  
       fill = "")  
print(03.density)
```

```
## Warning: Removed 2146 rows containing non-finite outside the scale range  
## ('stat_density()').
```



```
03.boxplot <-  
  ggplot(EPAair, aes(x = as.factor(Year),  
                     y = Ozone)) +  
  geom_boxplot() +  
  stat_summary(geom = "text",  
               fun = max,  
               vjust = -1,  
               size = 4,
```

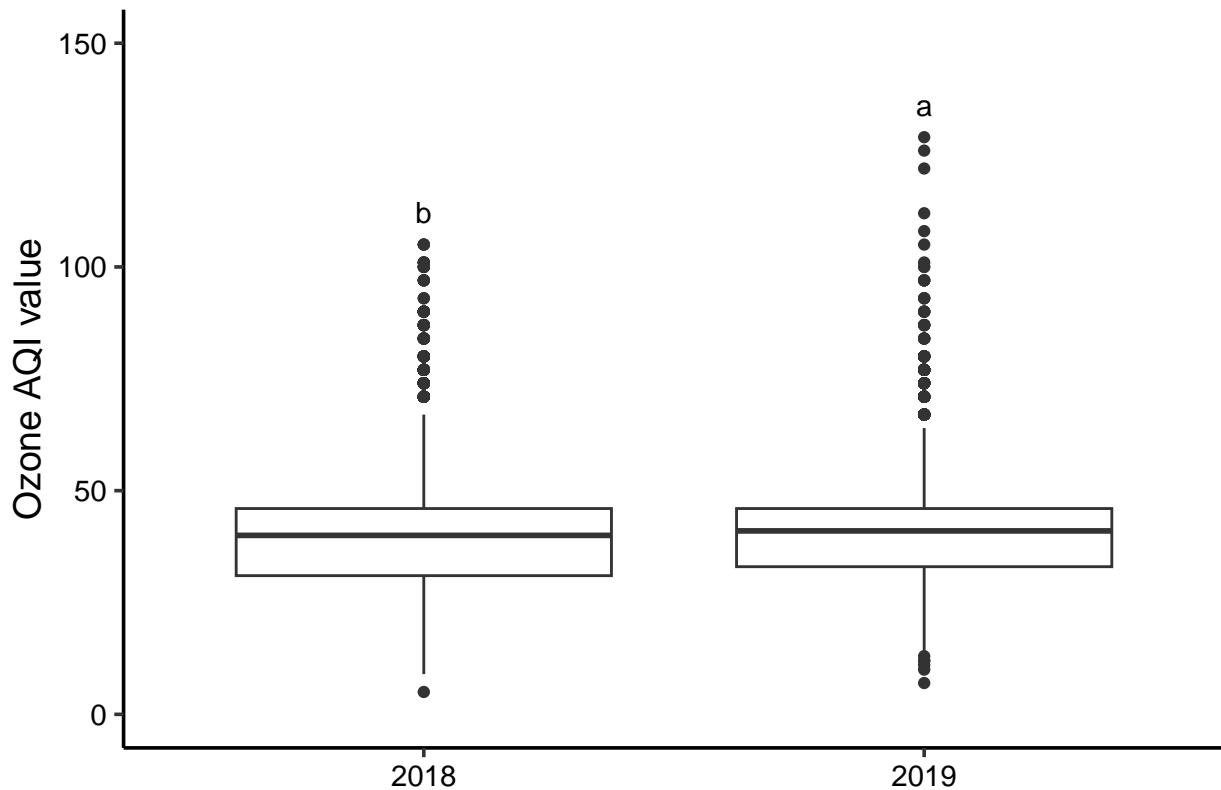
```

            label = c("b", "a")) +
  labs(x = "",
       y = "Ozone AQI value") +
  ylim(0,150)
print(03.boxplot)

## Warning: Removed 2146 rows containing non-finite outside the scale range
## ('stat_boxplot()').

## Warning: Removed 2146 rows containing non-finite outside the scale range
## ('stat_summary()').

```



```

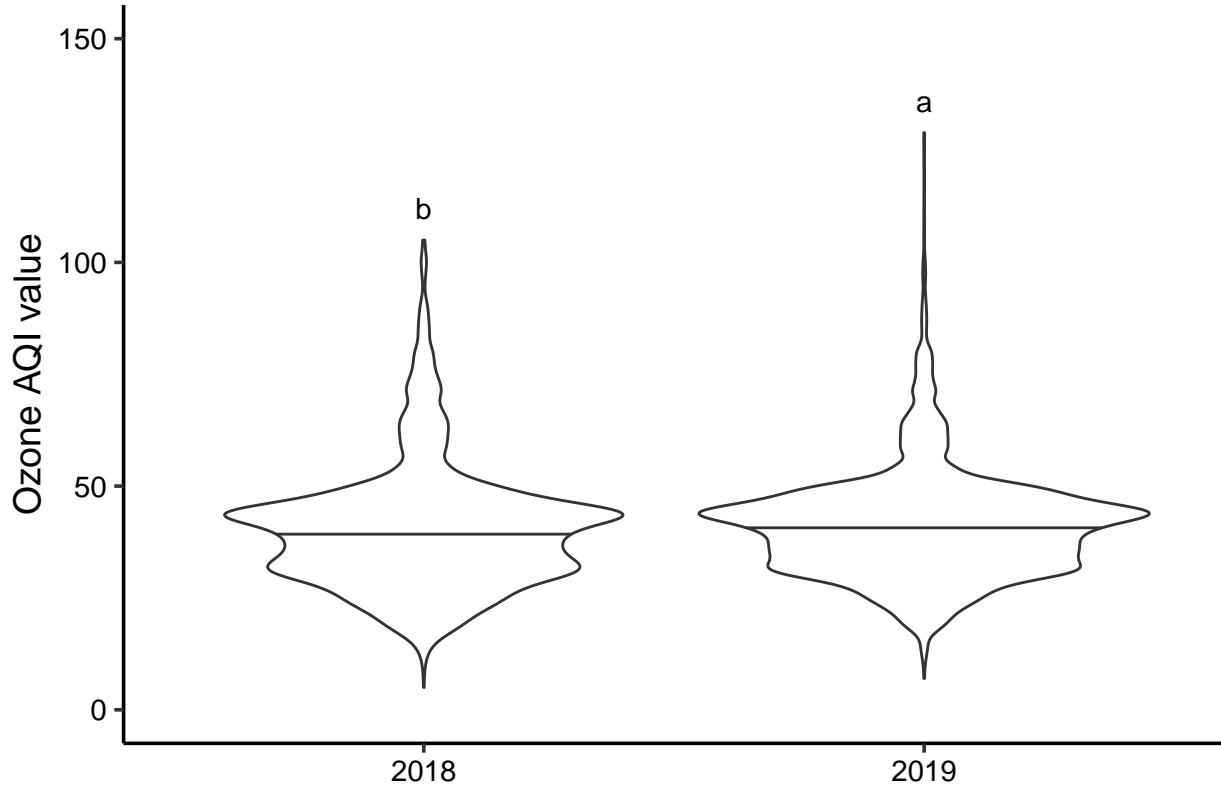
03.violin <-
  ggplot(EPAair, aes(x = as.factor(Year),
                      y = Ozone)) +
  geom_violin(draw_quantiles = 0.5) +
  stat_summary(geom = "text",
               fun = max,
               vjust = -1,
               size = 4,
               label = c("b", "a")) +
  labs(x = "",
       y = "Ozone AQI value") +
  ylim(0,150)
print(03.violin)

```

```

## Warning: Removed 2146 rows containing non-finite outside the scale range
## ('stat_ydensity()').
## Removed 2146 rows containing non-finite outside the scale range
## ('stat_summary()').

```



## Linear Regression

Important components of the linear regression are the correlation and the R-squared value. The **correlation** is a number between -1 and 1, describing the relationship between the variables. Correlations close to -1 represent strong negative correlations, correlations close to zero represent weak correlations, and correlations close to 1 represent strong positive correlations. The **R-squared value** is the correlation squared, becoming a number between 0 and 1. The R-squared value describes the percent of variance accounted for by the explanatory variables.

For the NTL-LTER dataset, can we predict PM2.5 from Ozone?

```
#Exercise 2: Run a linear regression PM2.5 by Ozone. Find the p-value and R-squared value.

PM2.5byOzone.regression <- lm(data = EPAair, PM2.5 ~ Ozone)
summary(PM2.5byOzone.regression)
```

```
##
## Call:
## lm(formula = PM2.5 ~ Ozone, data = EPAair)
##
```

```

## Residuals:
##      Min     1Q Median     3Q    Max
## -37.204 -8.931 -0.613  8.463 48.473
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.63824   0.55556  28.15 <2e-16 ***
## Ozone        0.38384   0.01298  29.58 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.06 on 5774 degrees of freedom
## (3200 observations deleted due to missingness)
## Multiple R-squared:  0.1316, Adjusted R-squared:  0.1314
## F-statistic: 874.9 on 1 and 5774 DF, p-value: < 2.2e-16

```

*#Exercise 3: Build a scatterplot. Add a line and standard error for the linear regression. Add the regression equation and R-squared value to the plot.*

```

PM2.5byOzone.Plot <-
  ggplot(EPAair, aes(x = Ozone,
                      y = PM2.5)) +
  geom_point() +
  geom_smooth(method = "lm",
              col="blue") +
  geom_text(
    x=100,
    y=75,
    label=expression("PM2.4=0.38*Ozone+15.64")
  )
print(PM2.5byOzone.Plot)

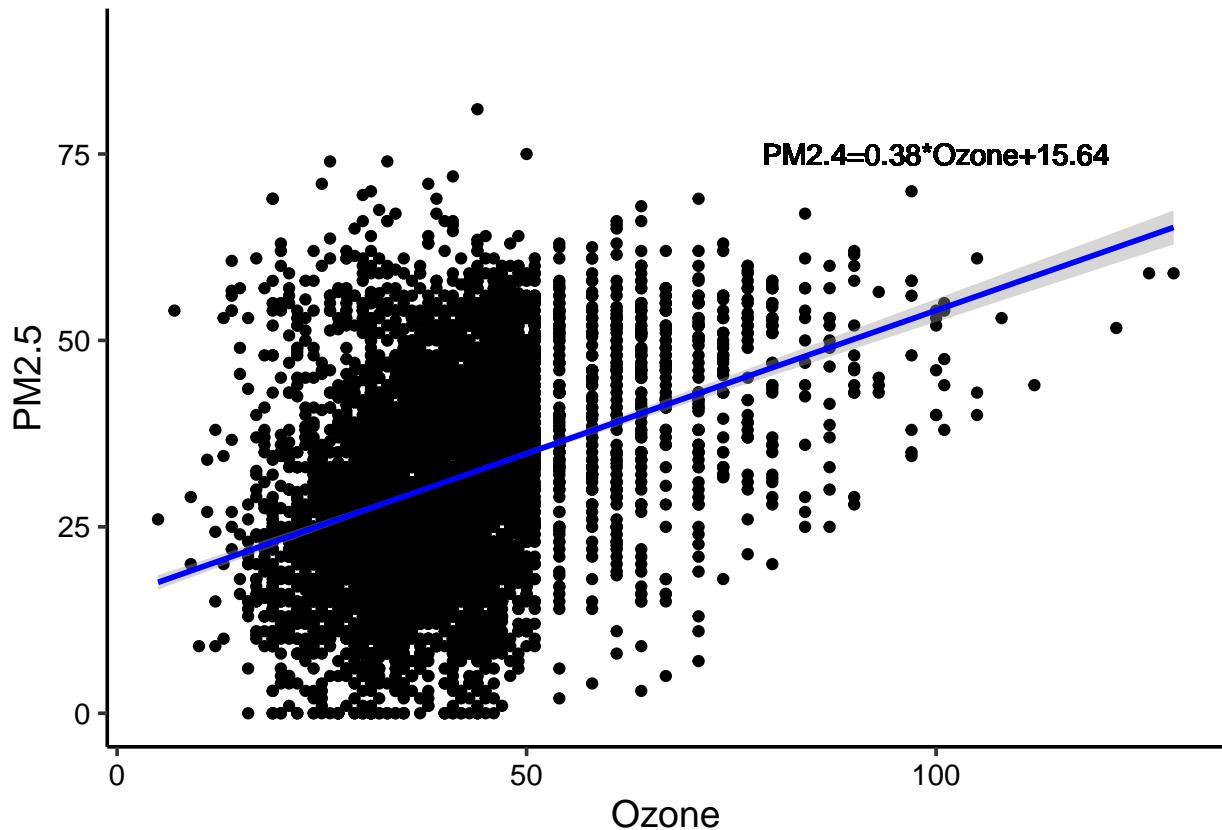
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 3200 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 3200 rows containing missing values or values outside the scale range
## ('geom_point()').

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

```



## AIC to select variables

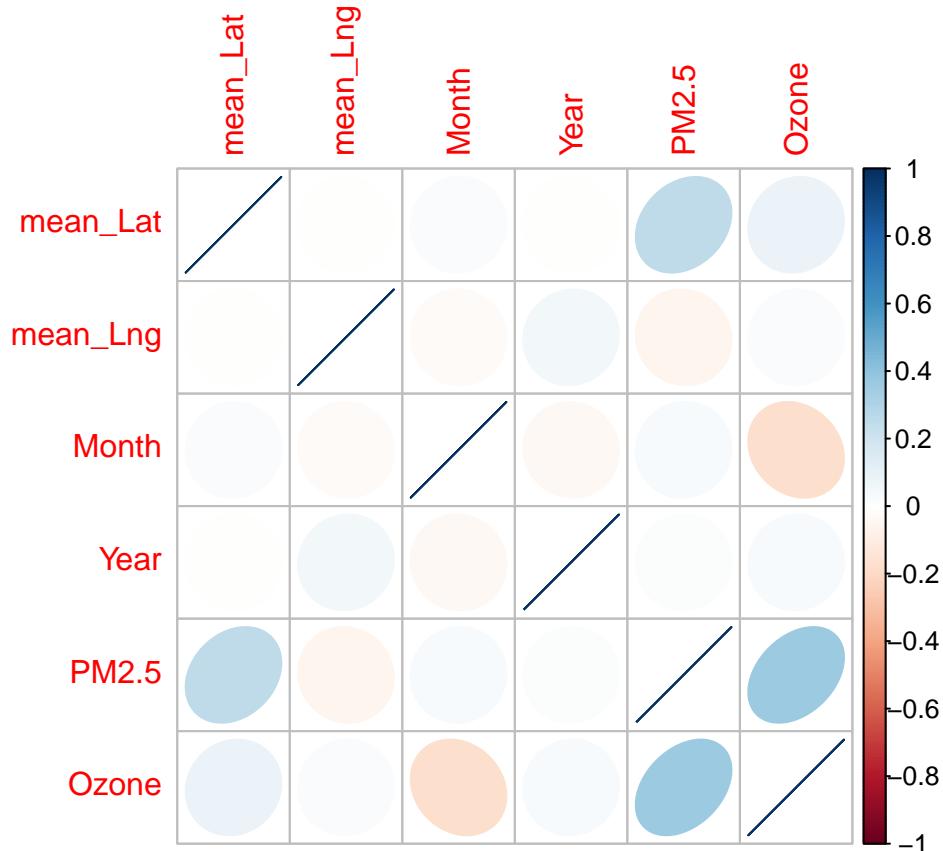
What other variables can we add to improve model?

```
#Exercise 4: Build correlation plots and identify more possible explanatory variables to add to the reg
library(corrplot)
```

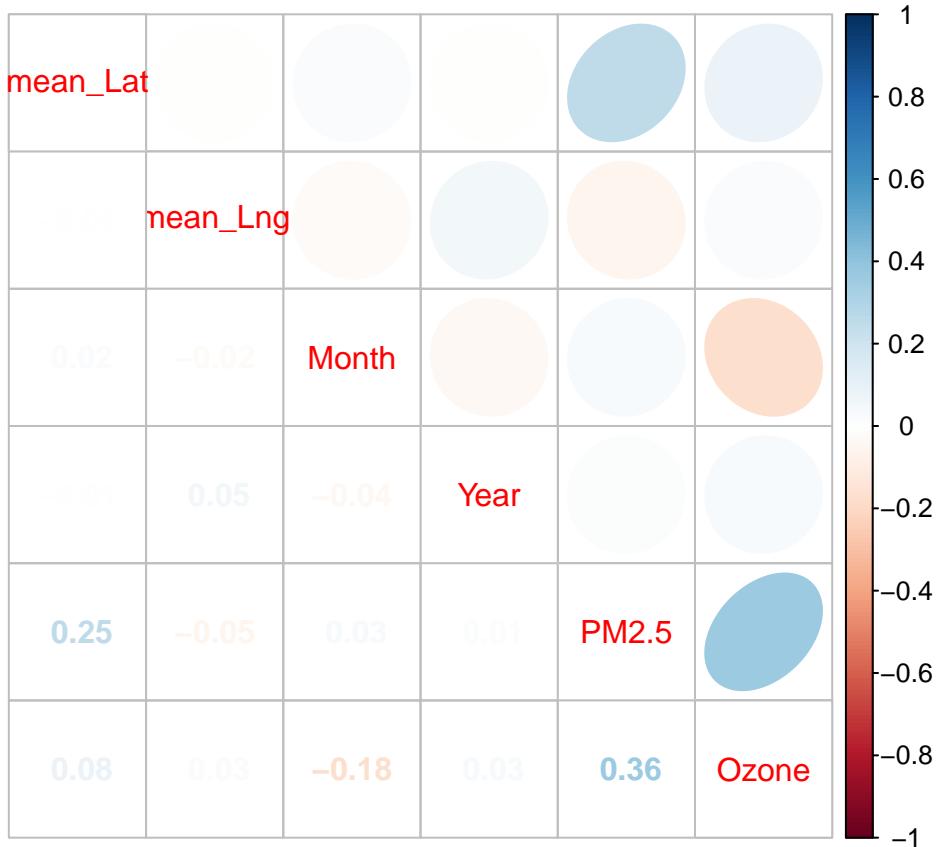
```
## corrplot 0.94 loaded

library(dplyr)
EPAair.subset <-
  EPAair %>%
  select(mean_Lat:Ozone) %>%
  na.omit()

EPAairCorr <- cor(EPAair.subset)
corrplot(EPAairCorr, method = "ellipse")
```



```
corrplot.mixed(EPAairCorr, upper = "ellipse")
```



#Exercise 5: Choose a model by AIC in a Stepwise Algorithm. Do the results from AIC match the variables

```
OzonebyALL.regression <- lm(data = EPAair.subset,
```

```
PM2.5 ~ Ozone + Year + Month + mean_Lng + mean_Lat)
```

```
step(OzonebyALL.regression)
```

```
## Start: AIC=29272.11
## PM2.5 ~ Ozone + Year + Month + mean_Lng + mean_Lat
##
##          Df Sum of Sq      RSS      AIC
## - Year     1    149  915695 29271
## <none>           915545 29272
## - mean_Lng  1    4087  919632 29296
## - Month    1    8874  924420 29326
## - mean_Lat  1    54272  969818 29603
## - Ozone    1   142142 1057688 30104
##
## Step: AIC=29271.05
## PM2.5 ~ Ozone + Month + mean_Lng + mean_Lat
##
##          Df Sum of Sq      RSS      AIC
## <none>           915695 29271
## - mean_Lng  1    4017  919712 29294
## - Month    1    8815  924510 29324
## - mean_Lat  1    54223  969918 29601
```

```

## - Ozone      1    142470 1058165 30104

##
## Call:
## lm(formula = PM2.5 ~ Ozone + Month + mean_Lng + mean_Lat, data = EPAair.subset)
##
## Coefficients:
## (Intercept)      Ozone        Month      mean_Lng      mean_Lat
## -259.2766      0.3826      0.4643     -0.4956       6.5210

#Exercise 6: Run another regression using the variables selected on Exercise 5. Compare r-squared value

OzoneBest.regression <- lm(data = EPAair.subset,
                           PM2.5 ~ Ozone + Month + mean_Lng + mean_Lat)
summary(OzoneBest.regression)

##
## Call:
## lm(formula = PM2.5 ~ Ozone + Month + mean_Lng + mean_Lat, data = EPAair.subset)
##
## Residuals:
##    Min     1Q   Median     3Q    Max
## -35.806 -8.846 -0.948  7.777 52.098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -259.27663  14.74368 -17.586 < 2e-16 ***
## Ozone        0.38257   0.01277  29.965 < 2e-16 ***
## Month        0.46427   0.06229   7.454 1.04e-13 ***
## mean_Lng     -0.49563   0.09850  -5.032 5.01e-07 ***
## mean_Lat      6.52098   0.35275  18.486 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.6 on 5771 degrees of freedom
## Multiple R-squared:  0.1926, Adjusted R-squared:  0.192
## F-statistic: 344.2 on 4 and 5771 DF,  p-value: < 2.2e-16

#rsquare imporved to 0.19

```

## Litter Exercise

```

# Wrangle the data
Litter.Totals <- Litter %>%
  group_by(plotID, collectDate, nlcdClass) %>%
  summarise(dryMass = sum(dryMass))

## 'summarise()' has grouped output by 'plotID', 'collectDate'. You can override
## using the '.groups' argument.

```

```

# Format ANOVA as aov
Litter.Totals.anova <- aov(data = Litter.Totals, dryMass ~ plotID)
summary(Litter.Totals.anova)

##           Df Sum Sq Mean Sq F value    Pr(>F)
## plotID      11  7584   689.5   4.813 1.45e-06 ***
## Residuals  198 28363   143.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

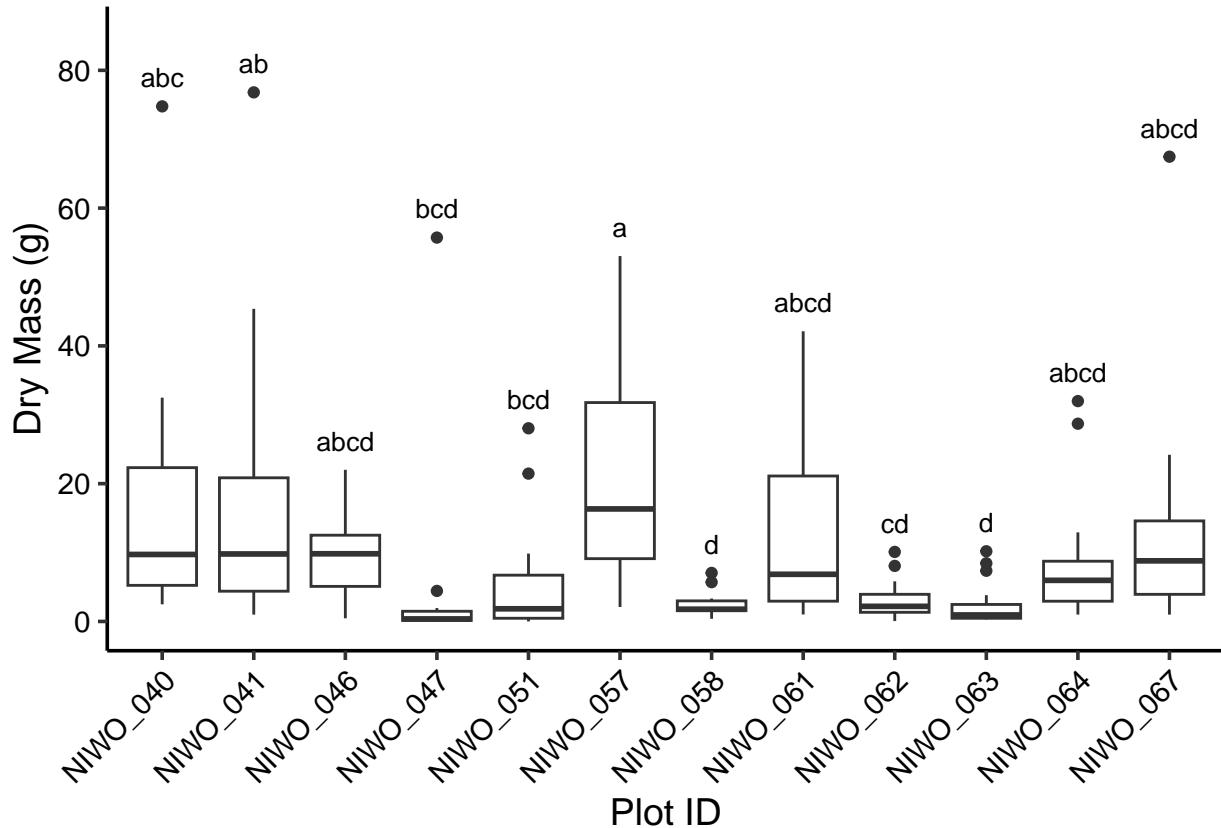
# Extract groupings for pairwise relationships
Litter.Totals.groups <- HSD.test(Litter.Totals.anova, "plotID", group = TRUE)
Litter.Totals.groups$groups

##          dryMass groups
## NIWO_057 20.685833     a
## NIWO_041 16.979063   ab
## NIWO_040 15.680000   abc
## NIWO_061 13.186111  abcd
## NIWO_067 12.565938  abcd
## NIWO_046  9.956176  abcd
## NIWO_064  8.015789  abcd
## NIWO_051  5.668750   bcd
## NIWO_047  4.476333   bcd
## NIWO_062  3.047632    cd
## NIWO_058  2.398421     d
## NIWO_063  2.393889     d

Litter.Totals <- Litter.Totals %>%
  mutate( treatgroups = Litter.Totals.groups$groups[plotID,2])

# Graph the results
Litter.Totals.plot <- ggplot(Litter.Totals, aes(x = plotID, y = dryMass)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  stat_summary(geom = "text", fun = max, vjust = -1, size = 3.5,
               label = c("abc", "ab", "abcd", "bcd", "bcd", "a",
                         "d", "abcd", "cd", "d", "abcd", "abcd")) +
  labs(x = "Plot ID", y = "Dry Mass (g)") +
  ylim(0, 85)
print(Litter.Totals.plot)

```



#Exercise 7: Improve the plot

```

# a. order by groups from high to low mean
assign.group <- function(site)
  return(Litter.Totals.groups$groups[as.character(site),2])

treatgroups=array(0,length(Litter.Totals))
for( i in length(Litter.Totals)){
  treatgroups = assign.group(Litter.Totals$plotID)
}

Litter.Totals <- cbind(Litter.Totals,groups = treatgroups)
Litter.Totals$plotID <- reorder(Litter.Totals$plotID,Litter.Totals$dryMass,FUN=mean)

Litter.Totals.plot2 <- ggplot(Litter.Totals, aes(x = plotID, y = dryMass)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  stat_summary(geom = "text", fun = max, vjust = -1, size = 3.5,
               label = Litter.Totals.groups$groups[,2]) +
  labs(x = "Plot ID", y = "Dry Mass (g)") +
  ylim(0, 85)
print(Litter.Totals.plot2)

```

