HW5_regression_inference

AUTHOR
Darrell Sonntag

PUBLISHED
February 27, 2024

Fit a multiple linear regression model to conduct statistical inference.

Do vehicles that pass emissions have lower emissions than those that don’t?

First, we are going to control for multiple factors that impact emissions

We will visually evaluate the impact that potential factors such as model year, location, fuel type and vehicle emission tests have on emissions

Then, we’ll fit a multiple linear regression to see if the vehicle emission test status has an impact on emissions.

First read in the data

summer.2022.IM.csv summer.2022.nonIM

In the data folder in the class public repository,

```r
summer.2022.IM <- read_csv("../../CE594R_25_data_science_class/data/summer.2022.IM.csv")
```

```
Rows: 32394 Columns: 35
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (21): LICENSE, DATE, VIN, Vehicle Type, Make, Model, Fuel, FuelGroup, W...
dbl   (9): Year, Zip, Registered Weight, SPEED, ACCEL, id, ER, Max GVWR, Cur...
lgl   (2): LICENSE_outofstate, Veh.info.corrected
date  (2): Last Emissions Date, Last Test Date Required
time  (1): TIME

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summer.2022.nonIM <- read_csv("../../CE594R_25_data_science_class/data/summer.2022.nonIM.csv")
```

```
Rows: 2082 Columns: 30
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (18): LICENSE, DATE, VIN, Vehicle Type, Make, Model, Fuel, FuelGroup, W...
dbl   (8): Year, Zip, Registered Weight, SPEED, ACCEL, id, ER, Max GVWR
lgl   (2): LICENSE_outofstate, Veh.info.corrected
date  (1): Last Emissions Date
time  (1): TIME
```

```
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## combine summer.2022.IM and summer.2022.nonIM

summer.2022 <- bind_rows(summer.2022.IM, summer.2022.nonIM)
```

This file includes the 4 dates of vehicle measurements from summer 2022. Amber has updated the files to include data on whether the vehicle is current with it's vehicle emission test requirement

Combine the two data files using bind_rows

```
summer.2022 <- bind_rows(summer.2022.IM, summer.2022.nonIM)
```

Reorder the compliance variable, Compliant, Not Compliant, Exempt, and NonIM

Calculate the a Complicance variable as a factor with levels Compliant, Not Compliant, Exempt, NonIM (but set ordered = False) (This is so the regression we will calculate later on, doesn't think there is an order to the factors)

```
summer.2022 <- summer.2022 %>%
  mutate(Compliance = factor(Compliance, levels = c("Compliant", "Not Compliant", "Exempt", "NonI
```

Just filter to the gasoline vehicles Add a variable called year_cuts Use the cut function to cut the Year variable, into intervals between the following break points: 1963, 1982, 1994, 2001, 2016, and 2023.

?cut

```
summer.2022.gas <- summer.2022 %>%
  filter(FuelGroup == "Gasoline") %>%
  filter(!is.na(Year)) %>%
  mutate(year_cuts = cut(Year, breaks = c(1963, 1982, 1994, 2001, 2016, 2023)))
```

Calculate mean, count, sd, and 95% CI by pollutant, year_cuts, and compliance

```
summer.2022.compliance.summary <- summer.2022.gas %>%
  group_by(pollutant, Compliance, year_cuts) %>%
  summarize(mean = mean(ER, na.rm=T),median = median(ER,na.rm=T),
            sd=sd(ER,na.rm=T),n=sum(!is.na(ER)),
            min=min(ER,na.rm=T),max=max(ER,na.rm=T)) %>%
  mutate(tcrit = qt(.975,df=(n-1))) %>%
  mutate(bound = tcrit*sd/sqrt(n)) %>%
  mutate(lower.95 = mean-bound) %>%
  mutate(upper.95 = mean+bound)
```

```
Warning: There were 4 warnings in `summarize()`.
The first warning was:
ℹ In argument: `min = min(ER, na.rm = T)`.
```

```
ℹ In group 76: `pollutant = "NO2"`, `Compliance = NonIM`, `year_cuts =
  "(1963,1982]"`.
Caused by warning in `min()`:
! no non-missing arguments to min; returning Inf
ℹ Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

```
`summarise()` has grouped output by 'pollutant', 'Compliance'. You can override
using the `.groups` argument.

Warning: There were 9 warnings in `mutate()`.
The first warning was:
ℹ In argument: `tcrit = qt(0.975, df = (n - 1))`.
ℹ In group 4: `pollutant = "CO"` and `Compliance = NonIM`.
Caused by warning in `qt()`:
! NaNs produced
ℹ Run `dplyr::last_dplyr_warnings()` to see the 8 remaining warnings.
```

Look at the summary file - How many exempt vehicles are there with CO measurements? - Only 11 vehicles

Graph the means with year_cuts on the x-axis

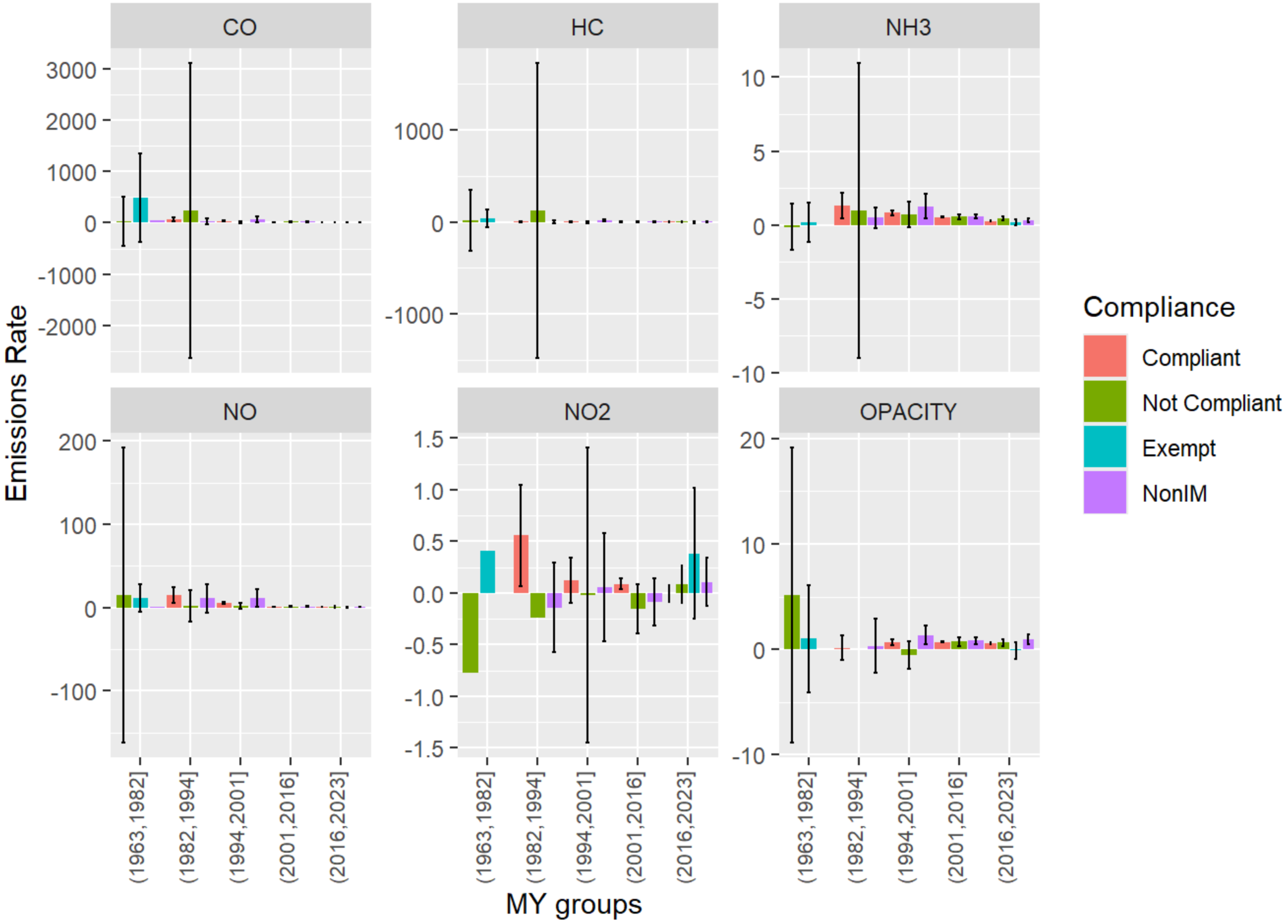
- Using geom_col, with dodge to plot all the compliance levels
- Different colors for compliance
- Year cuts on the x-axis
- Dodge the compliance
- Separate panels for each pollutant

```
names(summer.2022.compliance.summary)
```

```
 [1] "pollutant"  "Compliance" "year_cuts"  "mean"       "median"    
 [6] "sd"         "n"          "min"        "max"        "tcrit"     
[11] "bound"      "lower.95"   "upper.95"  
```

```
ggplot(data = summer.2022.compliance.summary,aes(x = year_cuts, y = mean, fill= Compliance)) +
geom_col(position = position_dodge(width = 1)) +
    geom_errorbar(position = position_dodge(width = 1), aes(ymin = lower.95, ymax = upper.95), wi
facet_wrap(~ pollutant, scales = "free_y") +
labs( x = "MY groups",
     y = "Emissions Rate") +
theme(axis.text.x = element_text(size=8, angle=90,hjust = 1, vjust =0.2))
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_col()`).
```



Plot again, but just for 1994 and later model year groups

- Just plot CO, HC, NH3, and NO,
- Use geom_errorbar to add 95% confidence intervals to the plot

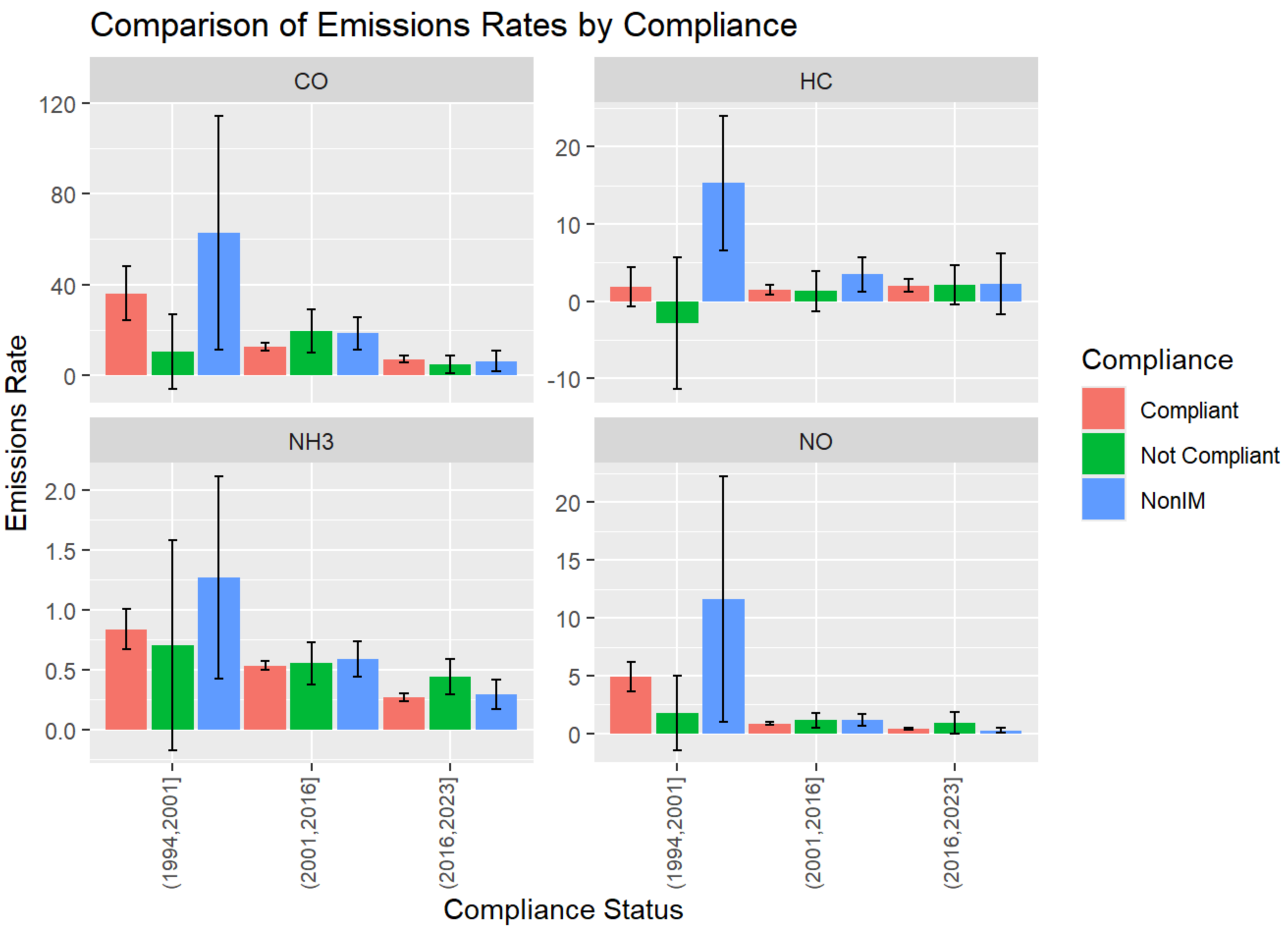
Also just plot the “Compliant”, “Not Compliant”,“NonIM”)

Re-create the following graph using ggplot

```
names(summer.2022.compliance.summary)
```

```
 [1] "pollutant"   "Compliance" "year_cuts"  "mean"        "median"
 [6] "sd"          "n"          "min"        "max"         "tcrit"
[11] "bound"       "lower.95"   "upper.95"
```

```
summer.2022.compliance.summary %>%
  filter(pollutant %in% c('CO','HC','NH3','NO')) %>%
  filter(Compliance %in% c("Compliant", "Not Compliant","NonIM")) %>%
  filter(year_cuts %in% c("(1994,2001]", "(2001,2016]" ,"(2016,2023]")) %>%
  ggplot(aes(x = year_cuts, y = mean, fill= Compliance)) +
  geom_col(position = position_dodge(width = 1)) +
  geom_errorbar(position = position_dodge(width = 1), aes(ymin = lower.95, ymax = upper.95), widt
  facet_wrap(~ pollutant, scales = "free_y") +
  labs(title = "Comparison of Emissions Rates by Compliance",
       x = "Compliance Status",
       y = "Emissions Rate") +
  theme(axis.text.x = element_text(size=8, angle=90,hjust = 1, vjust =0.2))
```



```
ggsave("../../CE594R_data_science_R/figs/ER_compliance.png")
```

```
Saving 7 x 5 in image
```

Do you observe any systematic or significant differences in mean Not Compliant or NonIM emission rates compared to the Compliant emission rates?

For the 1994-2001, the NonIM vehicles tend to be higher than the Compliant vehicles for all for CO, HC, NH3, and NO. However, the difference only appears significant for HC. For the newer model year groups, The Compliant means tends to be the lowest for all pollutants, but it is not significantly different.

Let’s look at some potential confounding variables:

Let’s plot the mean emission rates by year_cut and by location

```
summer.2022.loc.summary <- summer.2022.gas %>%
  group_by(pollutant, location, year_cuts) %>%
  summarize(mean = mean(ER, na.rm=T),median = median(ER,na.rm=T),
            sd=sd(ER,na.rm=T),n=sum(!is.na(ER)),
            min=min(ER,na.rm=T),max=max(ER,na.rm=T)) %>%
  mutate(tcrit = qt(.975,df=(n-1))) %>%
  mutate(bound = tcrit*sd/sqrt(n)) %>%
  mutate(lower.95 = mean-bound) %>%
  mutate(upper.95 = mean+bound)
```

```
Warning: There were 6 warnings in `summarize()`.
The first warning was:
ℹ In argument: `min = min(ER, na.rm = T)`.
ℹ In group 61: `pollutant = "NO2"`, `location = "Timp Hwy East"`, `year_cuts =
  "(1963,1982]"`.
Caused by warning in `min()`:
! no non-missing arguments to min; returning Inf
ℹ Run `dplyr::last_dplyr_warnings()` to see the 5 remaining warnings.
```

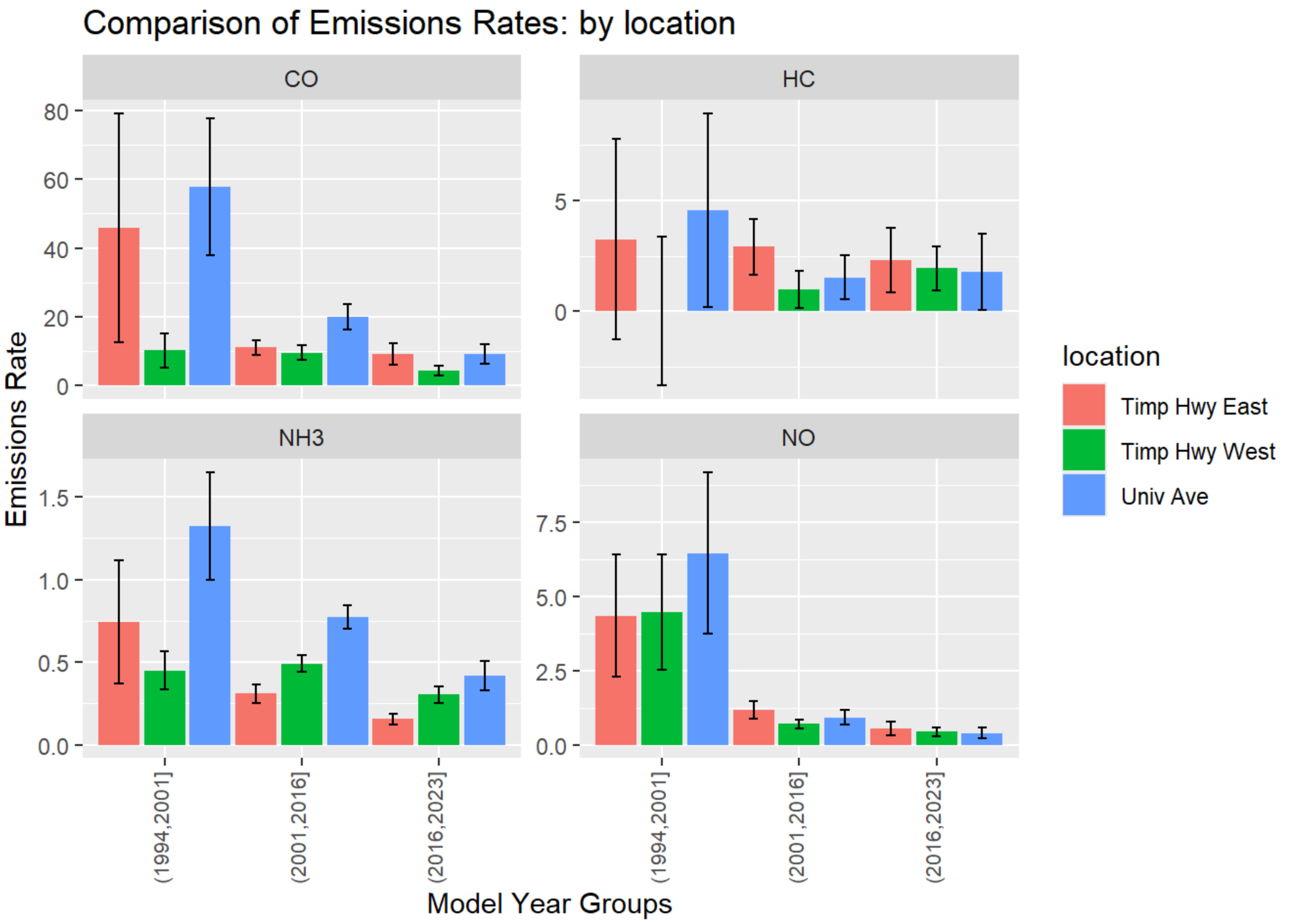
```
`summarise()` has grouped output by 'pollutant', 'location'. You can override
using the `.groups` argument.
```

```
Warning: There were 7 warnings in `mutate()`.
The first warning was:
ℹ In argument: `tcrit = qt(0.975, df = (n - 1))`.
ℹ In group 1: `pollutant = "CO"` and `location = "Timp Hwy East"`.
Caused by warning in `qt()`:
! NaNs produced
ℹ Run `dplyr::last_dplyr_warnings()` to see the 6 remaining warnings.
```

Graph the means by location (just for the newer model years)

Does there tend to be a systematic difference in the mean emission rates by location?

```
summer.2022.loc.summary %>%
  filter(pollutant %in% c('CO','HC','NH3','NO')) %>%
  filter(year_cuts %in% c("(1994,2001]", "(2001,2016]" ,"(2016,2023]")) %>%
  ggplot(aes(x = year_cuts, y = mean, fill= location)) +
  geom_col(position = position_dodge(width = 1)) +
  geom_errorbar(position = position_dodge(width = 1), aes(ymin = lower.95, ymax = upper.95), widt
  facet_wrap(~ pollutant, scales = "free_y") +
  labs(title = "Comparison of Emissions Rates: by location",
       x = "Model Year Groups",
       y = "Emissions Rate") +
  theme(axis.text.x = element_text(size=8, angle=90,hjust = 1, vjust =0.2))
```



```
ggsave("../../CE594R_data_science_R/figs/ER_location.png")
```

```
Saving 7 x 5 in image
```

Let’s fit a model to the data that controls for the other factors, location, model year

Let’s just focus on NH3 for now.

Create a new summer.2022.gas.NH3 data.frame from the summer.2022.gas

- Filter to just NH3
- Remove any rows with missing NH3 emission rate (column ER)
- Just filter on the Compliant, non compliant and nonIM vehicles (exclude the exempt, since there are so few..)
- Filter to Years less than 1993 (the newer vehicles which are capable of electronic emission tests)

```
summer.2022.gas.NH3 <- summer.2022.gas %>%
    filter(Compliance %in% c("Compliant", "Not Compliant","NonIM")) %>%
    filter(pollutant == 'NH3') %>%
    filter(Year>1994) %>%
    filter(!is.na(ER))
```

Fit a linear model

Use the model form: ER = intercept + location + Compliance + year + year^2

```
names(summer.2022.gas.NH3)
```

```
 [1] "LICENSE"                "DATE"                  
 [3] "TIME"                   "LICENSE_outofstate"    
 [5] "Veh.info.corrected"     "VIN"                   
 [7] "Vehicle Type"           "Make"                  
 [9] "Model"                  "Year"                  
[11] "Fuel"                   "FuelGroup"             
[13] "Zip"                    "Weight Rating"         
[15] "Registered Weight"      "Last Emissions Date"   
[17] "SPEED_FLAG"             "SPEED"                 
[19] "ACCEL"                  "TAG_NAME"              
[21] "location"               "id"                    
[23] "pollutant"              "ER"                    
[25] "CO2_FLAG"               "POLLUTANT_FLAG"        
[27] "County"                 "City"                  
[29] "Max GVWR"               "GVWR Requirement"      
[31] "Current Year"           "Required"              
[33] "Frequency"              "Last Test Date Required"
[35] "Compliance"             "year_cuts"             
```

```
lm.NH3 = lm(formula = ER ~  location + Compliance + poly(Year,2),
            data = summer.2022.gas.NH3 )
```

Look at summary

```
summary(lm.NH3)
```

```

Call:
lm(formula = ER ~ location + Compliance + poly(Year, 2), data = summer.2022.gas.NH3)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.3329 -0.4071 -0.1884  0.0879 13.0681 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)              0.274449   0.024255  11.315  < 2e-16 ***
locationTimp Hwy West    0.147274   0.030420   4.841 1.33e-06 ***
locationUniv Ave         0.408143   0.033949  12.022  < 2e-16 ***
ComplianceNot Compliant  0.038804   0.054604   0.711    0.477    
ComplianceNonIM         -0.008992   0.054502  -0.165    0.869    
poly(Year, 2)1          -9.250140   0.899297 -10.286  < 2e-16 ***
poly(Year, 2)2          -1.391429   0.892046  -1.560    0.119    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.8912 on 5118 degrees of freedom
Multiple R-squared:  0.05495,	Adjusted R-squared:  0.05384 
F-statistic:  49.6 on 6 and 5118 DF,  p-value: < 2.2e-16
```

How good is our model at explaining the variability of individual vehicles? What is the R2?, Is it good?

0.05

Not very good at explaining the variability… but perhaps, it can help explain differences the impact the average emissions.

Is the Compliance coefficient significant?

- No

Let’s look at our model coefficients

```
lm.Nh3.coef <- lm.NH3 %>%
               get_regression_table()
```

Get the predictions of your model to the data

```
library(broom)
```

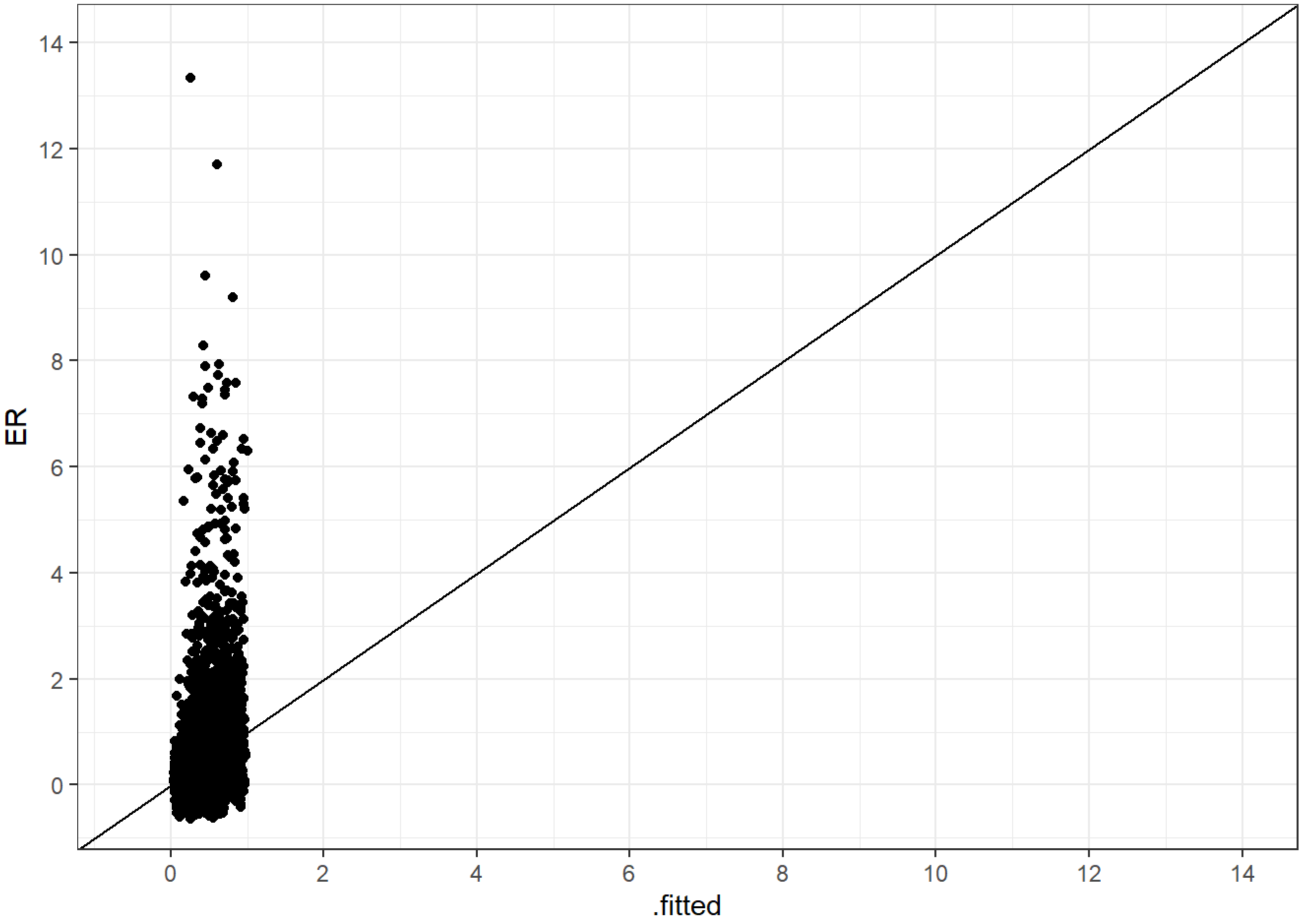
```
Warning: package 'broom' was built under R version 4.4.3
```

```
predict.summer.2022.gas.NH3 <- lm.NH3 %>%
                               augment(interval = 'confidence')
```

Plot your model estimates

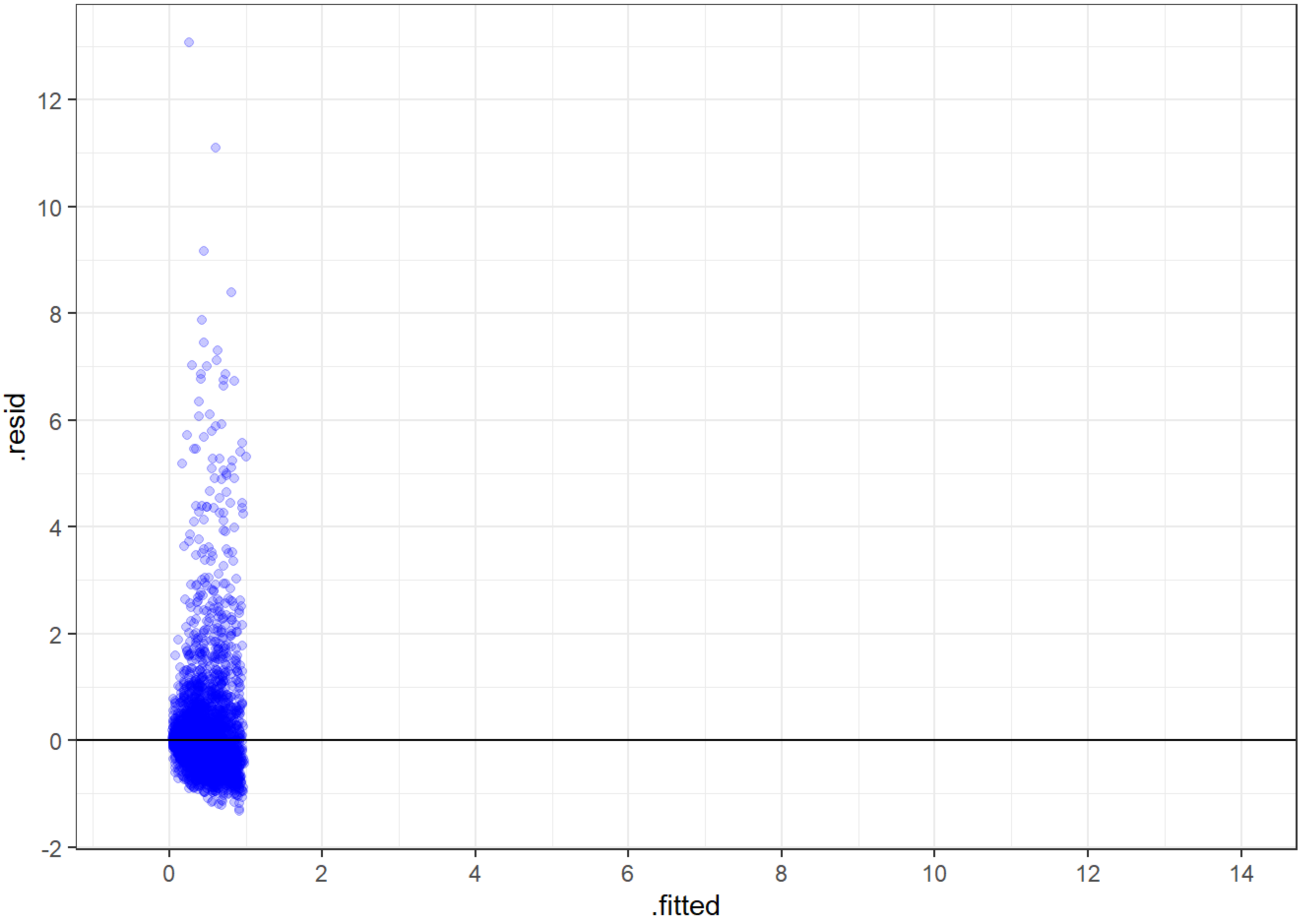
Plot the NH3 emission rate on the y-axis Plot the model estimated (fitted) emission rates on the x-axis

```
ggplot(data = predict.summer.2022.gas.NH3,aes(x = .fitted,y=ER)) +
    geom_point() +
    geom_abline(intercept=0, slope=1)+
    coord_cartesian(xlim = c(-0.5,14),ylim=c(-0.5,14))+
    scale_y_continuous(breaks = seq(0,14,2))+
    scale_x_continuous(breaks = seq(0,14,2))+
    theme_bw()
```



Now plot the predicted values (x-axis) vs. the residuals (y-axis) How do my residuals look like they are independent, and normally distributed with a constant variance across the predictors?

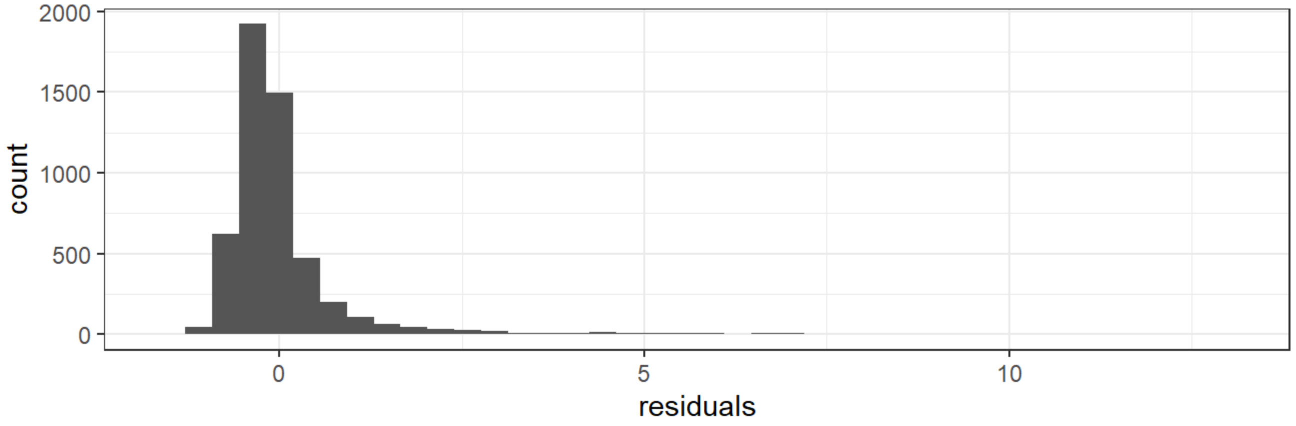
```
ggplot(data = predict.summer.2022.gas.NH3,aes(y = .resid,x=.fitted)) +
    geom_point(alpha = 0.2,color='blue') +
    geom_abline(intercept=0, slope=0)+
    coord_cartesian(xlim = c(-0.5,14))+
    scale_y_continuous(breaks = seq(-2,14,2))+
    scale_x_continuous(breaks = seq(0,14,2))+
    theme_bw()
```

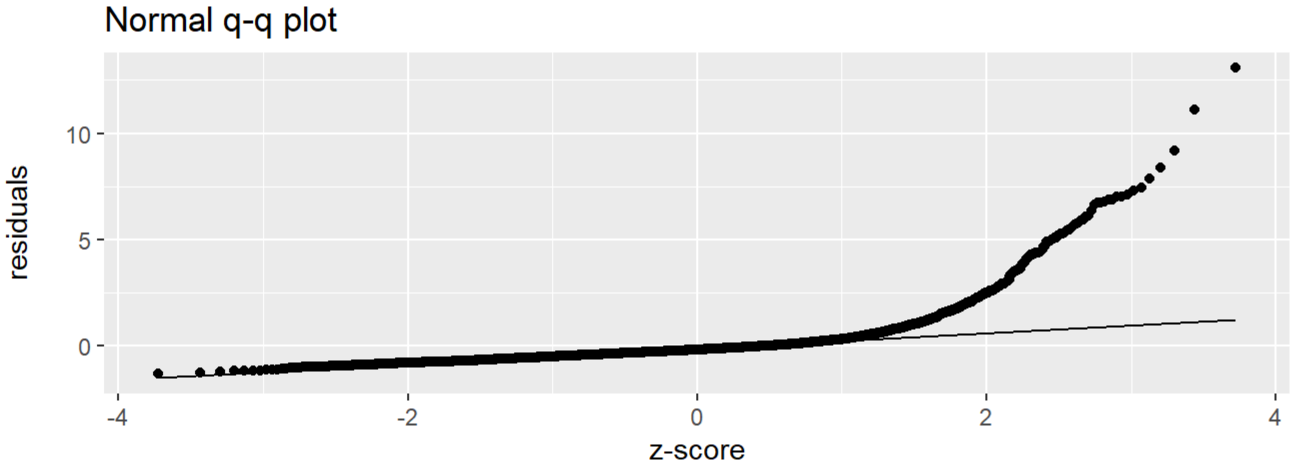


Not– they don’t look look normally distributed, and the variation appears to be changing

Also, look at the histogram and q-q plot of the residuals

```
hg.resid <- ggplot(data = predict.summer.2022.gas.NH3,aes(x = .resid)) +
  geom_histogram(bins=40)+
  theme_bw()+
  labs(x='residuals', y= 'count')  +
  theme(legend.position= 'none')


qq.resid <- ggplot(predict.summer.2022.gas.NH3, aes(sample=.resid)) +
    geom_qq()+
    geom_qq_line()+
    labs(title = 'Normal q-q plot',y='residuals', x='z-score') +
    theme(legend.position = 'none')

hg.resid + qq.resid + plot_layout(nrow = 2)
```





How do the residuals look? Do you think I can assume that the residuals are approximately normal?

No, they are highly skewed

Should I trust my statistical tests from the multiple linear regression model?

Probably not, unless the p-values are very small

Our linear model seems to be the wrong model fit to to our data. For example, It seems the the impact of location would be better modeled as a multiplicative factor.

Let’s try a log transformation of our y data.

Now, let’s look if if we can have find a log-linear relationship

There is a good discussion of log-normal transformation here:

https://smogdr.github.io/edar_coursebook/transform.html#transformation

And an example of using transformations in linear regression here: https://smogdr.github.io/edar_coursebook/model.html#example-ols-linear-regression

Let’s calculate the ln(ER)

Note: you can’t take the LN of any negative values.

Create a new data.frame called summer.2022.gas.NH3.pos

- First, Remove all the negative NH3 emission rates (ER)
- Calculate the ln(ER)

```
summer.2022.gas.NH3.pos <- summer.2022.gas.NH3 %>%
                           filter(ER > 0) %>%
                           mutate(ln.ER = log(ER))
```

How many obs do we have now? How many did we lose?

```
dim(summer.2022.gas.NH3)
```

```
[1] 5125   36
```

```
dim(summer.2022.gas.NH3.pos)
```

```
[1] 4187   37
```

```
dim(summer.2022.gas.NH3) - dim(summer.2022.gas.NH3.pos)
```

```
[1] 938  -1
```

We had 5125 NH3 obs. We lost 938, and now have 4187 obs.

Fit a linear model to predict the ln(ER) with additive terms for location, model year, and compliance, and interaction terms for each

```
lm.ln.NH3 = lm(formula = ln.ER ~  location + Compliance + Year,
            data = summer.2022.gas.NH3.pos )

summary(lm.ln.NH3)
```

```

Call:
lm(formula = ln.ER ~ location + Compliance + Year, data = summer.2022.gas.NH3.pos)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.6790 -0.7400  0.0770  0.8347  4.1547 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)             76.840942   6.812156  11.280   <2e-16 ***
locationTimp Hwy West    0.446746   0.050152   8.908   <2e-16 ***
locationUniv Ave         0.726241   0.054078  13.429   <2e-16 ***
ComplianceNot Compliant -0.005845   0.086145  -0.068    0.946    
ComplianceNonIM          0.015776   0.085955   0.184    0.854    
```

```
Year                   -0.039036   0.003383 -11.541   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.283 on 4181 degrees of freedom
Multiple R-squared:  0.07901,   Adjusted R-squared:  0.07791
F-statistic: 71.74 on 5 and 4181 DF,  p-value: < 2.2e-16
```

Not that these factors are additive in log-space. But multiplicative in real-space.

$$ln(ER) = a * location + b * year + c * compliance$$

$$\exp(\ln(ER)) = \exp(a * location + b * year + c * compliance)$$

$$ER = \exp(a * location) * \exp(b * year) * \exp(c * compliance)$$

In real-space, the impact of location, year, and compliance in our model are multiplicative.

Looking at the graph of the mean emission rates by model year and location A multiplicative effect of location on the emission rates looks like what we want. (For example, the Univ. Ave on NH3 looks ~ 2 times higher than the Timp HWY East for NH3 for each of the model year groups)

Graph to see if there are potential interaction terms between compliance and model year, and between model year and location, and between

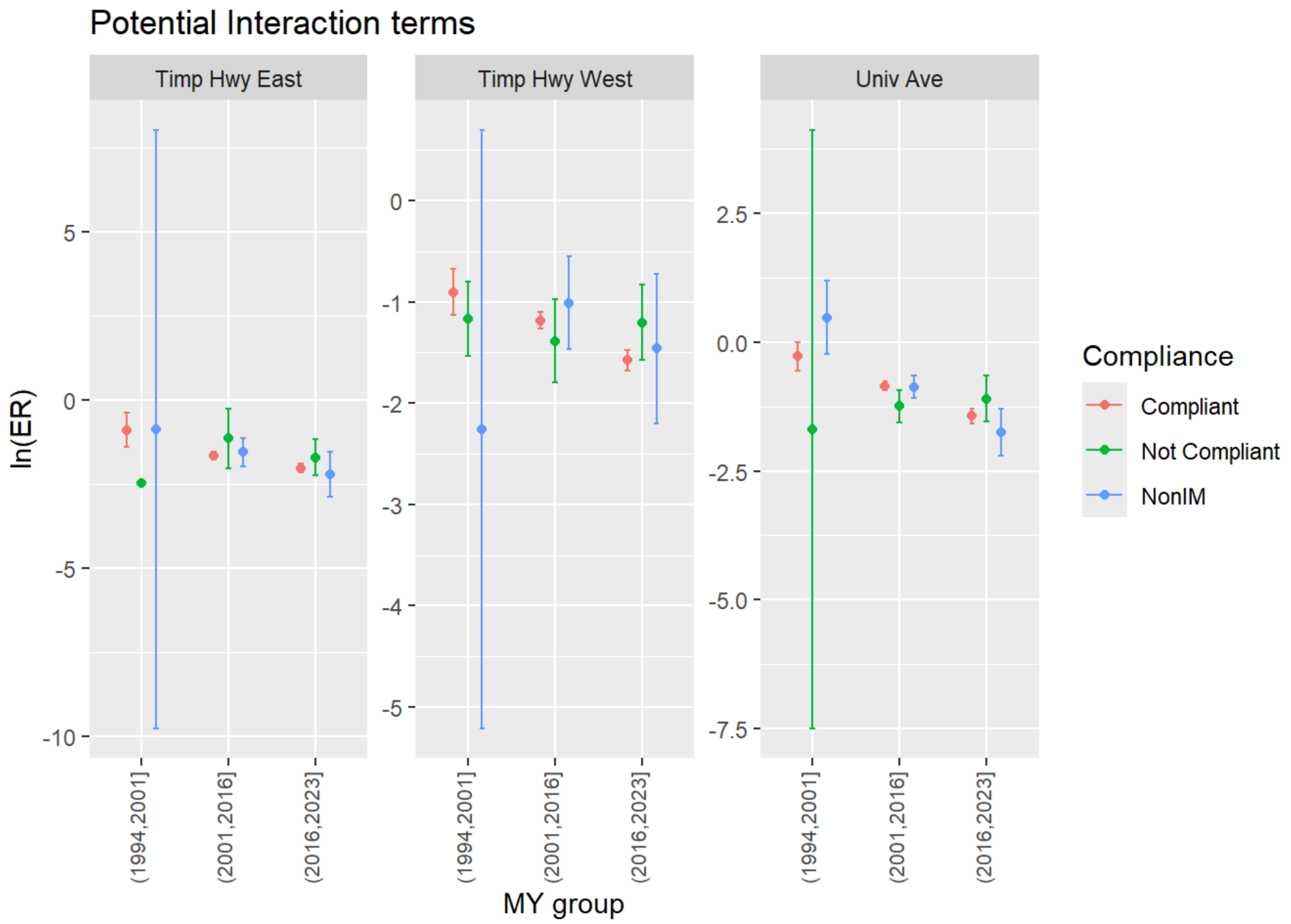
```
names(summer.2022.gas.NH3.pos)
```

```
 [1] "LICENSE"                "DATE"
 [3] "TIME"                   "LICENSE_outofstate"
 [5] "Veh.info.corrected"     "VIN"
 [7] "Vehicle Type"           "Make"
 [9] "Model"                  "Year"
[11] "Fuel"                   "FuelGroup"
[13] "Zip"                    "Weight Rating"
[15] "Registered Weight"      "Last Emissions Date"
[17] "SPEED_FLAG"             "SPEED"
[19] "ACCEL"                  "TAG_NAME"
[21] "location"               "id"
[23] "pollutant"              "ER"
[25] "CO2_FLAG"               "POLLUTANT_FLAG"
[27] "County"                 "City"
[29] "Max GVWR"               "GVWR Requirement"
[31] "Current Year"           "Required"
[33] "Frequency"              "Last Test Date Required"
[35] "Compliance"             "year_cuts"
[37] "ln.ER"
```

```
ln.NH3.year.loc.comp.summary <- summer.2022.gas.NH3.pos %>%
  group_by(pollutant, location, Compliance, year_cuts) %>%
  summarize(mean = mean(ln.ER, na.rm=T),median = median(ln.ER,na.rm=T),
            sd=sd(ln.ER,na.rm=T),n=sum(!is.na(ln.ER)),
            min=min(ln.ER,na.rm=T),max=max(ln.ER,na.rm=T)) %>%
  mutate(tcrit = qt(.975,df=(n-1))) %>%
  mutate(bound = tcrit*sd/sqrt(n)) %>%
  mutate(lower.95 = mean-bound) %>%
  mutate(upper.95 = mean+bound)
```

```
`summarise()` has grouped output by 'pollutant', 'location', 'Compliance'. You
can override using the `.groups` argument.

Warning: There was 1 warning in `mutate()`.
ℹ In argument: `tcrit = qt(0.975, df = (n - 1))`.
ℹ In group 2: `pollutant = "NH3"`, `location = "Timp Hwy East"`, `Compliance =
  Not Compliant`.
Caused by warning in `qt()`:
! NaNs produced
```

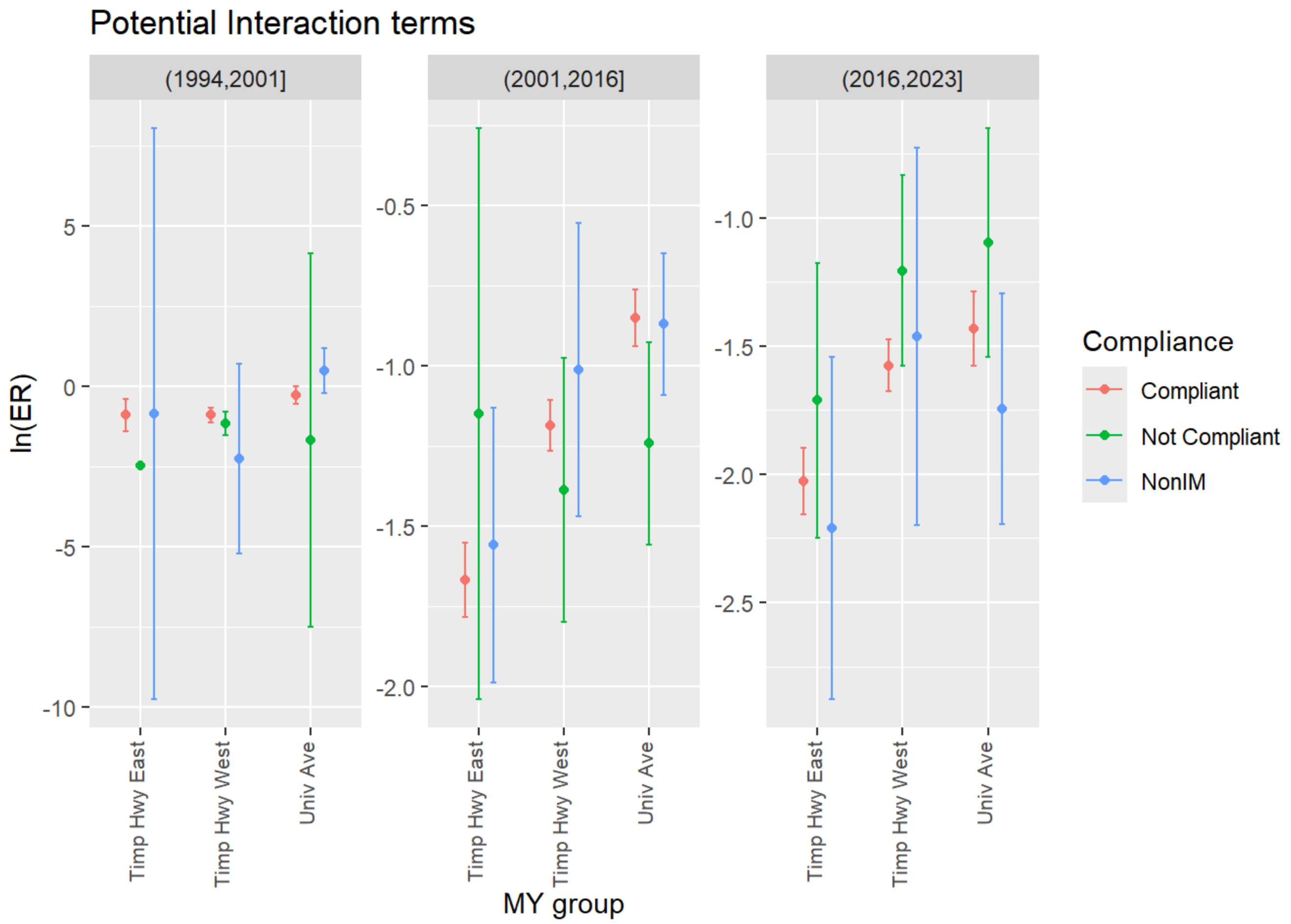
```
  ggplot(data = ln.NH3.year.loc.comp.summary , aes(x = year_cuts, y = mean, color= Compliance)) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbar(position = position_dodge(width = 0.5), aes(ymin = lower.95, ymax = upper.95), wi
  facet_wrap(.~ location, scales = "free_y") +
  labs(title = "Potential Interaction terms",
       x = "MY group",
       y = "ln(ER)") +
  theme(axis.text.x = element_text(size=8, angle=90,hjust = 1, vjust =0.2))
```



Looking at the graph, does it look like there are potential interaction terms?

- The slope with year, looks to be the same across model years
- The slope between compliance types looks signifiantly different. The Not Compliant slope with year looks like it is lower than the others
- I made another plot to look if there is a compliance X location effect. It looks like it is difficult to say…

```
ggplot(data = ln.NH3.year.loc.comp.summary , aes(x =location, y = mean, color= Compliance)) +
geom_point(position = position_dodge(width = 0.5)) +
geom_errorbar(position = position_dodge(width = 0.5), aes(ymin = lower.95, ymax = upper.95), wi
facet_wrap(.~ year_cuts, scales = "free_y") +
labs(title = "Potential Interaction terms",
   x = "MY group",
   y = "ln(ER)") +
theme(axis.text.x = element_text(size=8, angle=90,hjust = 1, vjust =0.2))
```



Add 3 different interaction terms between Compliance, Year, and Location, are they significant?

Note: Section 14.2 describes how to fit an interaction term:

Linear Model With R, e-book at byu library

https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1910500

```
lm.ln.NH3 = lm(formula = ln.ER ~  location + Compliance + Year + Year:location + Year:Compliance +
            data = summer.2022.gas.NH3.pos )

summary(lm.ln.NH3)
```

```
Call:
lm(formula = ln.ER ~ location + Compliance + Year + Year:location +
    Year:Compliance + location:Compliance, data = summer.2022.gas.NH3.pos)

Residuals:
    Min      1Q  Median      3Q     Max
-5.6986 -0.7397  0.0804  0.8301  4.1139

Coefficients:
```

```
                                        Estimate Std. Error t value
(Intercept)                            9.253e+01  1.398e+01   6.617
locationTimp Hwy West                 -2.951e+01  1.735e+01  -1.701
locationUniv Ave                      -1.187e+00  1.841e+01  -0.064
ComplianceNot Compliant               -6.904e+01  3.024e+01  -2.283
ComplianceNonIM                        2.862e+01  2.898e+01   0.987
Year                                  -4.683e-02  6.944e-03  -6.744
locationTimp Hwy West:Year             1.488e-02  8.614e-03   1.727
locationUniv Ave:Year                  9.554e-04  9.146e-03   0.104
ComplianceNot Compliant:Year           3.441e-02  1.501e-02   2.293
ComplianceNonIM:Year                  -1.424e-02  1.439e-02  -0.989
locationTimp Hwy West:ComplianceNot Compliant -2.241e-01  2.491e-01  -0.899
locationUniv Ave:ComplianceNot Compliant      -4.358e-01  2.616e-01  -1.666
locationTimp Hwy West:ComplianceNonIM          9.962e-02  2.607e-01   0.382
locationUniv Ave:ComplianceNonIM               6.842e-03  2.436e-01   0.028
                                              Pr(>|t|)
(Intercept)                                   4.14e-11 ***
locationTimp Hwy West                           0.0889 .
locationUniv Ave                                0.9486
ComplianceNot Compliant                         0.0225 *
ComplianceNonIM                                 0.3235
Year                                          1.75e-11 ***
locationTimp Hwy West:Year                      0.0842 .
locationUniv Ave:Year                           0.9168
ComplianceNot Compliant:Year                    0.0219 *
ComplianceNonIM:Year                            0.3226
locationTimp Hwy West:ComplianceNot Compliant   0.3685
locationUniv Ave:ComplianceNot Compliant        0.0959 .
locationTimp Hwy West:ComplianceNonIM           0.7024
locationUniv Ave:ComplianceNonIM                0.9776
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.282 on 4173 degrees of freedom
Multiple R-squared:  0.08255,   Adjusted R-squared:  0.07969
F-statistic: 28.88 on 13 and 4173 DF,  p-value: < 2.2e-16
```

Remove any interaction terms that are not significant (use p-value of 0.05).

Based on your graphs, do the results make sense?

```
lm.ln.NH3 = lm(formula = ln.ER ~  location + Compliance + Year + Year:Compliance,
               data = summer.2022.gas.NH3.pos )

summary(lm.ln.NH3)
```

```
Call:
lm(formula = ln.ER ~ location + Compliance + Year + Year:Compliance,
    data = summer.2022.gas.NH3.pos)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-5.6749 -0.7409  0.0782  0.8388  4.1623 

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)    
(Intercept)                  79.167282   7.224266  10.959   <2e-16 ***
locationTimp Hwy West         0.447329   0.050118   8.926   <2e-16 ***
locationUniv Ave              0.726255   0.054037  13.440   <2e-16 ***
ComplianceNot Compliant     -78.368369  29.855871  -2.625   0.0087 ** 
ComplianceNonIM              32.257341  28.503030   1.132   0.2578    
Year                         -0.040192   0.003587 -11.204   <2e-16 ***
ComplianceNot Compliant:Year  0.038914   0.014826   2.625   0.0087 ** 
ComplianceNonIM:Year         -0.016026   0.014166  -1.131   0.2580    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.282 on 4179 degrees of freedom
Multiple R-squared:  0.0809,	Adjusted R-squared:  0.07936 
F-statistic: 52.55 on 7 and 4179 DF,  p-value: < 2.2e-16
```

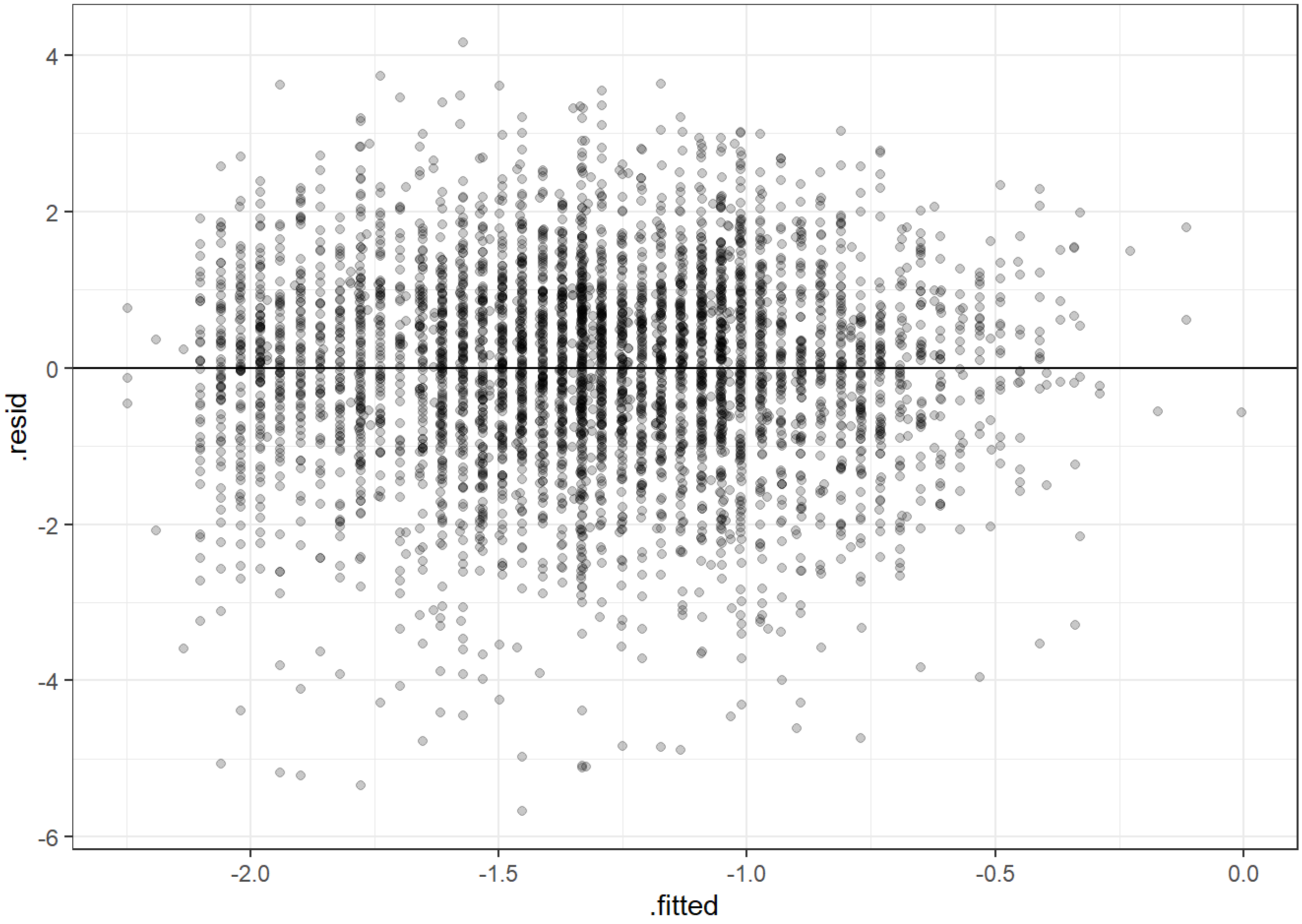
Plot the residuals from this model

First put them into a data.frame

```r
predict.lm.ln.NH3 <- lm.ln.NH3 %>%
                     augment(data = summer.2022.gas.NH3.pos,interval = 'confidence') %>%
                     arrange(.fitted)
```

Then plot the fitted (x-axis) and the residuals (y-axis)

```r
ggplot(data = predict.lm.ln.NH3,aes(y = .resid,x=.fitted)) +
    geom_point(alpha=0.2) +
    geom_abline(intercept=0, slope=0)+
    theme_bw()
```

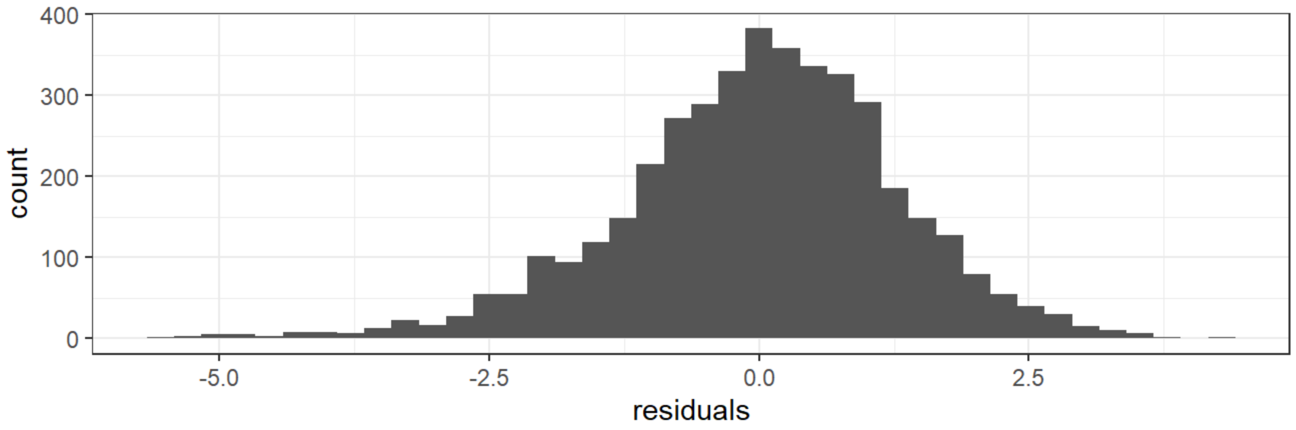


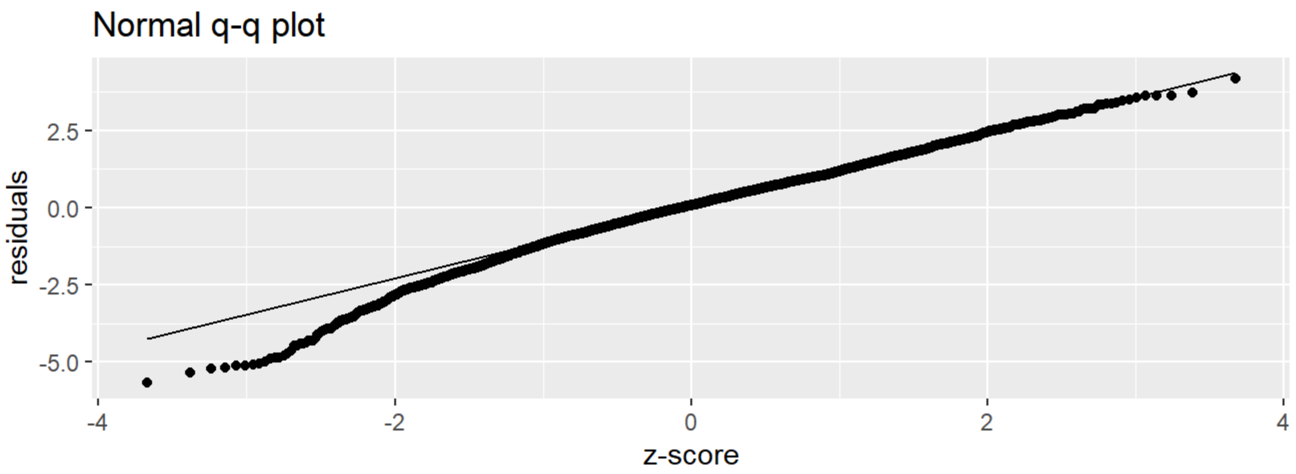
How do my residuals vs. predicted values looks?

They look much better, they look independent with constant variation

Let’s also look at the histogram and q-q plot of the residuals

```
hg.resid <- ggplot(data = predict.lm.ln.NH3,aes(x = .resid)) +
  geom_histogram(bins=40)+
  theme_bw()+
  labs(x='residuals', y= 'count')  +
  theme(legend.position= 'none')


qq.resid <- ggplot(predict.lm.ln.NH3, aes(sample=.resid)) +
    geom_qq()+
    geom_qq_line()+
    labs(title = 'Normal q-q plot',y='residuals', x='z-score') +
    theme(legend.position = 'none')

hg.resid + qq.resid + plot_layout(nrow = 2)
```





Look at the model coefficients, is the effect of compliance significant?

Store the coefficients in a data.frame table and then print them

```
lm.ln.NH3.coef <- lm.ln.NH3%>%
              get_regression_table()
```

```
library(tinytable)
```

```
Warning: package 'tinytable' was built under R version 4.4.2
```

```
tt(lm.ln.NH3.coef)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	79.167	7.224	10.959	0.000	65.004	93.331
location: Timp Hwy West	0.447	0.050	8.926	0.000	0.349	0.546

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
location: Univ Ave	0.726	0.054	13.440	0.000	0.620	0.832
Compliance: Not Compliant	-78.368	29.856	-2.625	0.009	-136.902	-19.835
Compliance: NonIM	32.257	28.503	1.132	0.258	-23.624	88.138
Year	-0.040	0.004	-11.204	0.000	-0.047	-0.033
Compliance: Not Compliant:Year	0.039	0.015	2.625	0.009	0.010	0.068
Compliance: NonIM:Year	-0.016	0.014	-1.131	0.258	-0.044	0.012

Question: What do the significant interaction terms mean?

Answer

Compliance: Not Compliant

The ln(ER) intercept term is significantly lower for the Not Compliant emissions Non-IM doesn't appear to be significantly different intercept than Compliant.

Compliance: Not Compliant:Year

The ln(ER) slope for the Not Compliant group is significantly different than the Compliant group.l For the Not Compliant Group the observations the slope is close to zero.

To help us understand our model coefficients better, let's graph the results.

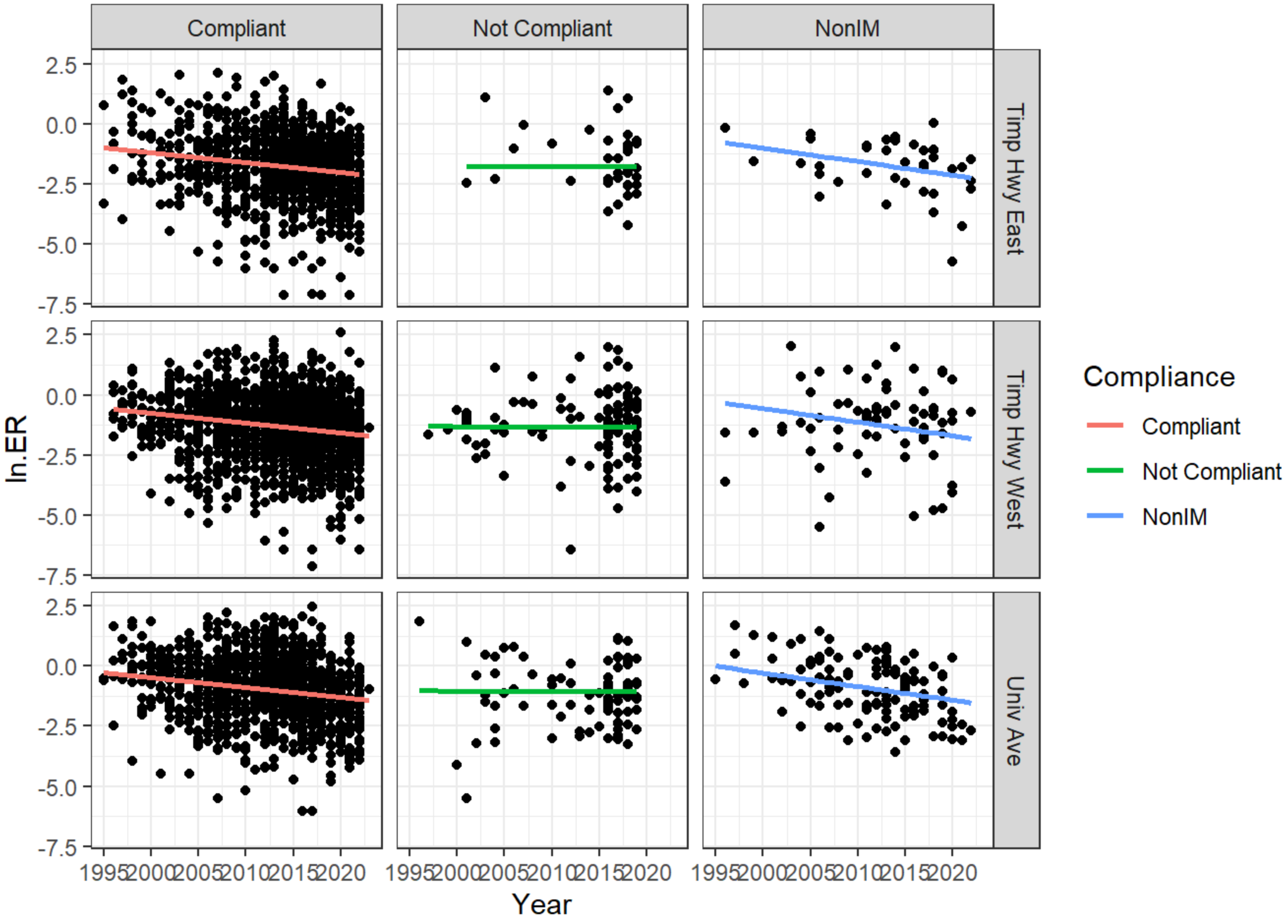
Plot the model fit, like the picture below

Plot different panels for each location

- Plot Year on the x-axis
- Plot the observed data on the y-axis using geom_point
- Plot the predicted emission rate on the y-axis using geom_line

```
ggplot(data = predict.lm.ln.NH3,aes(x = Year)) +
    geom_point(aes(y=ln.ER))+
    geom_line(aes(y=.fitted, color = Compliance), size = 1)+
    facet_grid(location ~ Compliance)+
    theme_bw()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```



```
#ggsave("../../CE594R_data_science_R/figs/Year_lnER_predict.png")
```

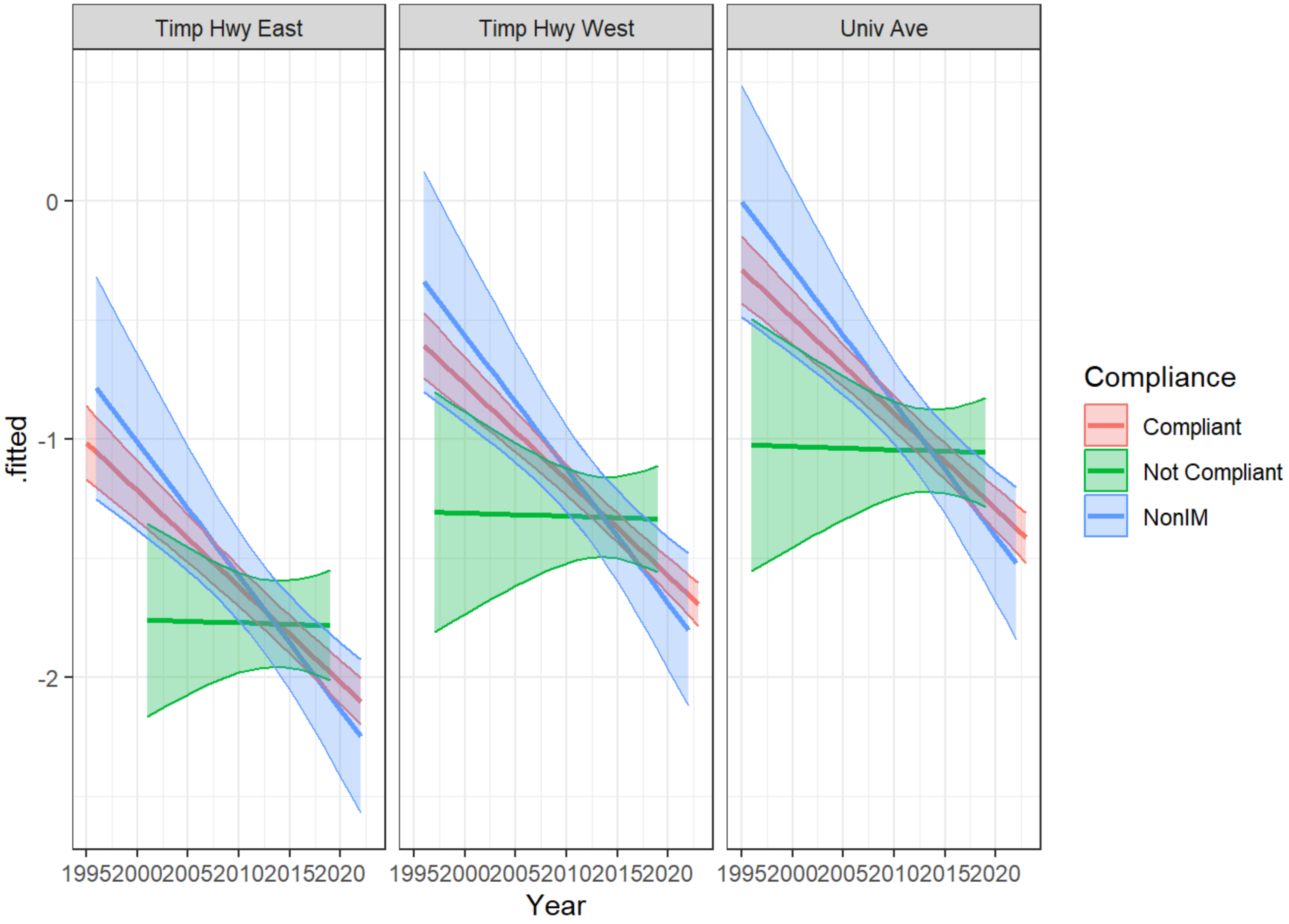
- Create the following plot of predictions and the standard error of the mean predictions

-Use geom_line for the predictions Use geom_ribbon for the standard error of the mean predictions

Only plot the predictions, so you can see differences.

Like InClass10

```
ggplot(data = predict.lm.ln.NH3,aes(x = Year, color=Compliance)) +
    geom_line(aes(y=.fitted), size = 1)+
    geom_ribbon(aes(ymin = .lower, ymax = .upper,fill=Compliance),alpha=0.3) +
    facet_wrap(~location)+
    theme_bw()
```



```
#ggsave("../../CE594R_data_science_R/figs/predict_lnER_comp.png")
```

Are the mean predictions significantly different by model year?

Answer:

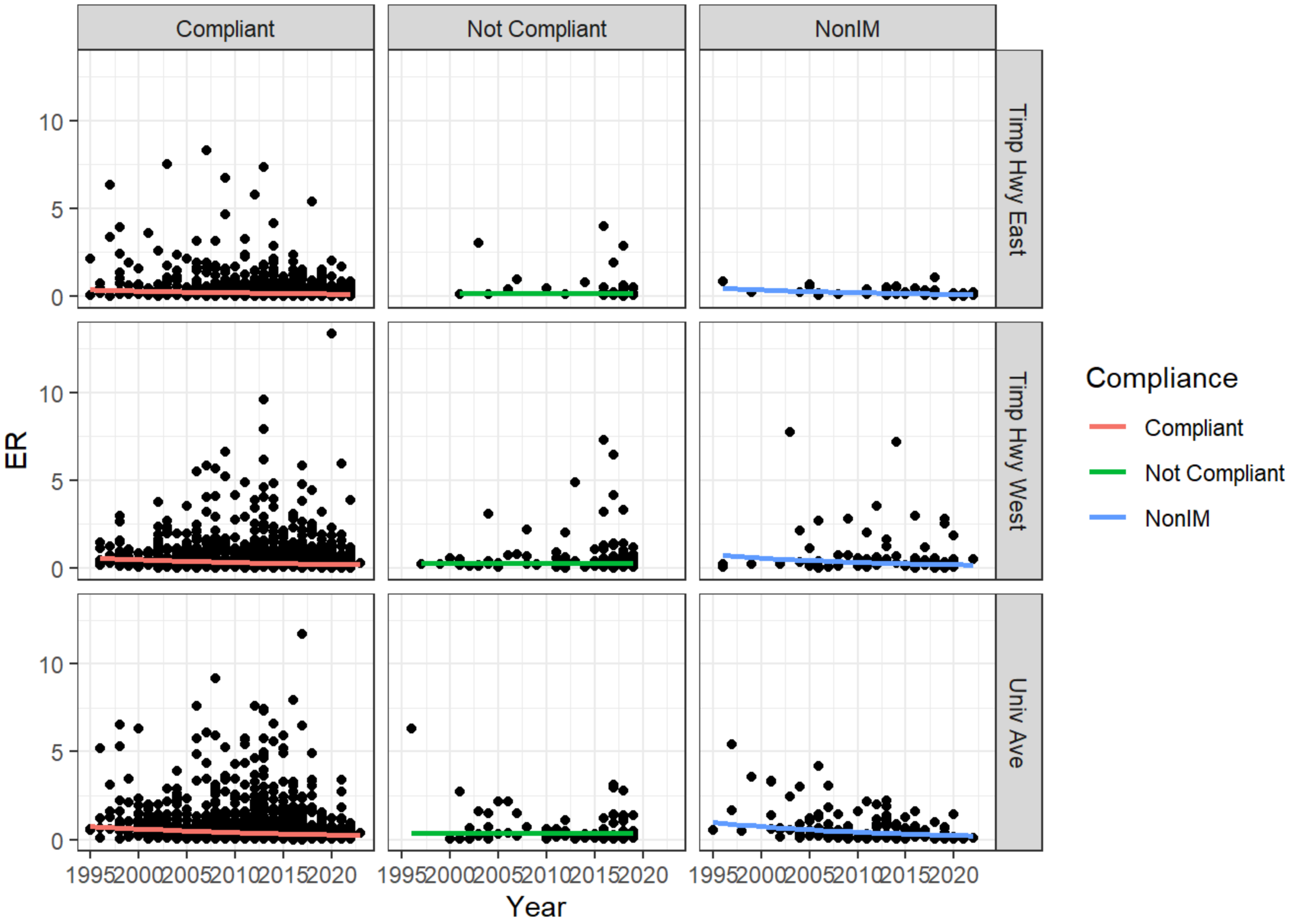
- The Compliant and non-IM vehicles don’t appear to be different
- The Non-Compliant have lower emission rates for the older Model years (pre-2005)

Next, Plot the model predictions in real-space,and compare to the real observations

Take the exponent of the predictions Plot the predicted values by Compliance Plot separate facets for Compliance level and Location

Re-create the following graph:

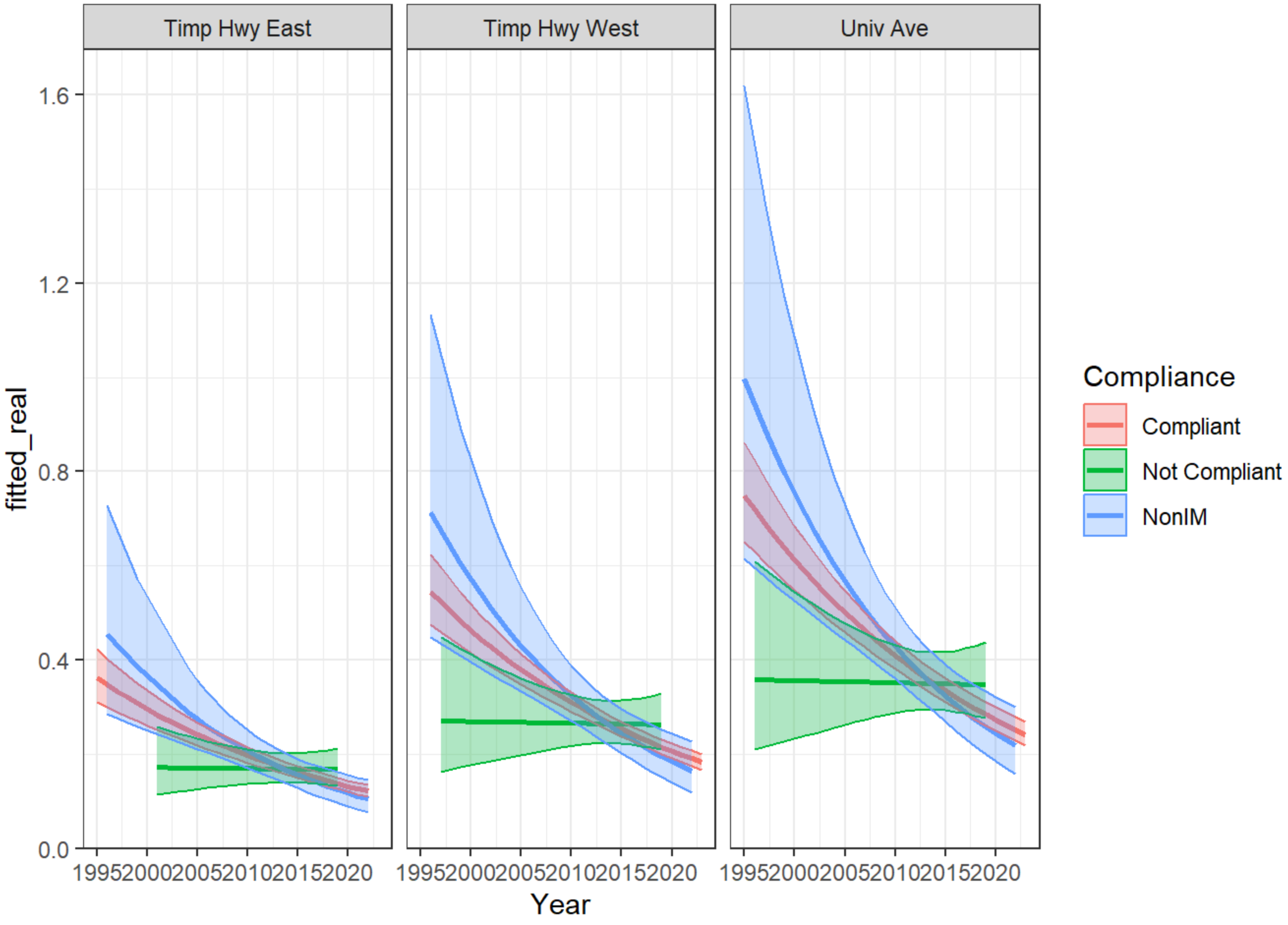
```
predict.lm.ln.NH3 <- predict.lm.ln.NH3 %>%
                     mutate(fitted_real = exp(.fitted)) %>%
                     mutate(lower_real = exp(.lower)) %>%
                     mutate(upper_real = exp(.upper))


ggplot(data = predict.lm.ln.NH3,aes(x = Year)) +
    geom_point(aes(y=ER))+
    geom_line(aes(y=fitted_real, color = Compliance),size=1)+
    facet_grid(location ~ Compliance)+
    theme_bw()
```



```
#ggsave("../../CE594R_data_science_R/figs/predict_lnER_comp_real.png")
```

Then compare just the mean model predictions and confidence intervals (remove the obs)

```
ggplot(data = predict.lm.ln.NH3,aes(x = Year, color=Compliance, fill=Compliance)) +
    geom_line(aes(y=fitted_real), size = 1)+
    geom_ribbon(aes(ymin = lower_real, ymax = upper_real,fill=Compliance),alpha=0.3) +
    facet_wrap(~location)+
    theme_bw()
```



```
#ggsave("../../CE594R_data_science_R/figs/predict_lnER_real_conf.png")
```

In the end, we fit a log-linear model that seemed to fit the data quite well. It also gave us some results that we were able to interpret.

Before we pad ourselves on the back. What are some issues with our model fitting steps?

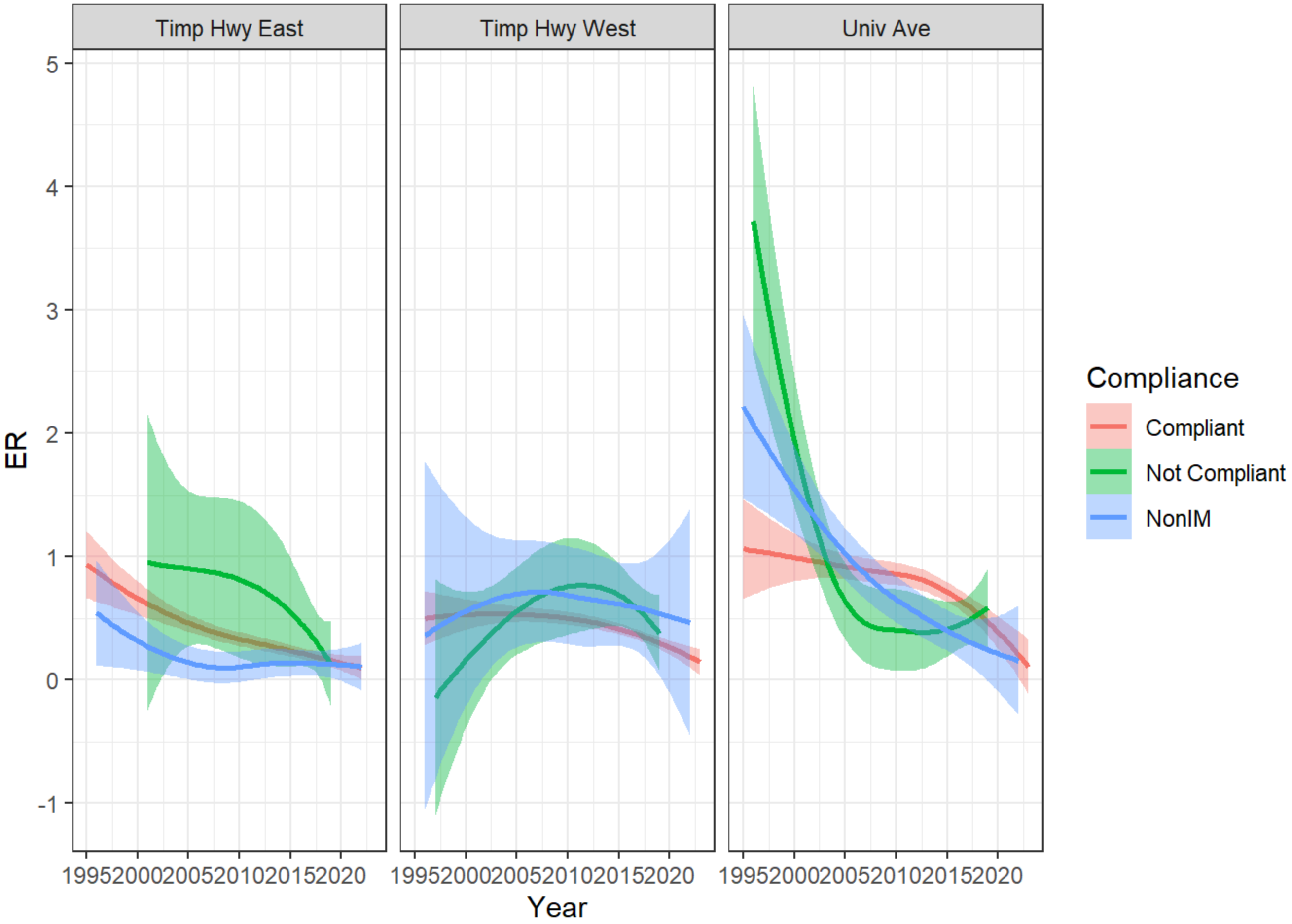
- We threw out ~1000 negative values. What if there tended to be more negative values on the compliant or not compliant vehicles?...
- Let's fit a non-linear smoothing function to the real (untransformed) data
- Let's smooth by location and Compliance type
- Let's plot those with the 95% confidence intervals
- let's use Loess (localized regression smoothing, the default smoother in geom_smooth
- You can adjust the 'span' to get a "good smooth"

```
names(summer.2022.gas.NH3)
```

```
 [1] "LICENSE"                "DATE"
 [3] "TIME"                   "LICENSE_outofstate"
 [5] "Veh.info.corrected"     "VIN"
 [7] "Vehicle Type"           "Make"
 [9] "Model"                  "Year"
[11] "Fuel"                   "FuelGroup"
[13] "Zip"                    "Weight Rating"
[15] "Registered Weight"      "Last Emissions Date"
[17] "SPEED_FLAG"             "SPEED"
[19] "ACCEL"                  "TAG_NAME"
[21] "location"               "id"
[23] "pollutant"              "ER"
[25] "CO2_FLAG"               "POLLUTANT_FLAG"
[27] "County"                 "City"
[29] "Max GVWR"               "GVWR Requirement"
[31] "Current Year"           "Required"
[33] "Frequency"              "Last Test Date Required"
[35] "Compliance"             "year_cuts"
```

```
ggplot(data = summer.2022.gas.NH3,aes(x = Year, y=ER,color=Compliance)) +
    geom_smooth(method = 'loess',span = 1.5,aes(fill = Compliance))+
    facet_wrap(~location)+
    theme_bw()
```

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



```
ggsave("../../CE594R_data_science_R/figs/NH3_smooth.png")
```

```
Saving 7 x 5 in image
`geom_smooth()` using formula = 'y ~ x'
```

How do the model estimates from our log-linear model compare to our non-linear model?

Do the magnitude of our predictions change?

Yes! The University avenue predictions in 1995 changed from "<" 1 mg. Now they are ">" 1 mg/kg

Wait… shouldn't the predictions with the real-data be lower (since we included all the negative values?)

Let's think about the mean of exponent values

The exponent of the mean of the ln(x) is called the geometric mean– and is not equal to the sample mean.

In other words:

exp(mean(ln(x))) != mean(x)

The mean of x will be more influenced by outliers. In our case, ln(x) reduces the distance to the positive outliers, the geometric mean is closer to the cloud of data at smaller values.

Do the relative order of the model estimates change regarding the compliance variables change?

Yes! Now the ‘Not compliant’ vehicles tend to have higher mean values across model years

Lessons:

Linear models with untransformed variables

Pros

- Can be easier interpret coefficients
- Fit will be to the mean of the data
- Don’t lose any negative variables

Cons

- The relationship may be non-linear
- Residuals may be highly non-normal and can’t run accurate diagnositcs
- May need to make a complex linear model to fit the data (polynomial terms and interaction terms) that is difficult to interpret

Transformed variables

Pros

- May be able to fit a linear form to a transformed variable (such as a log-linear relationship)
- May be able to correct the residuals to be approximately normal

Cons

- More difficult to interpret the results
- Our model doesn’t fit to the mean of the real-data, but fits the mean of the transformed data
- This may discount the impact of outlier values, that may be very important to the mean statistic
- We may lose data

Non-linear models

Pros

- Can fit the untransformed data, that have non-linear relationships

Cons

- Our residuals may still not meet the assumptions of normally distributed and identically distributed
- More complex methods to solve non-linear fits (we may not be guaranteed to find the optimal solution)
- Communication: Non-linear models are more complex. People are unfamiliar with them than linear models