1. Brief Summary

The paper studies the effect of different regulations on the ozone pollution. RVP ,RFG and CARB all aim to reduce the ozone formation, but targeted different chemical compound which could react with NO_x to form ozone.. For example, RVP standard only targeted on the total emission of VOC and, on the other hand, CARB targeted on more reactive VOC. The authors take the advantage that these standards happened at different time in different state to use DID (comparing different states using different regulation) and RD as method (comparing emission of same monitor for different regulation period) as their empirical strategy. The results show that CARB does have better effect on ozone reduction than other regulation. The author concluded that firms in different states react differently to respond the hexogenious standard: in RFG and RVP implemented area, the refiners could achieve the standard by removing lowly reactive compounds (butane) in their refining progress, only slightly increasing their cost.

What they do effectively is they use RD strategy to answer the difference of statistical significance between linear trend and quadratic trend, which should have been very similar. Also, they give mature consideration to their regression model, yet making complexity a lot. Besides, the motivation of the paper is pretty attentive, providing a point of view which we may ignore about the environmental standard. The questions I have for this paper is that when we want to study on pollution emission and use weather as one of our covariates, is it a must to use this complicated weather controls? Also, they include lots of fixed effects in their did model and one of the fix effect is interaction with DOY, what is the difference of Region- DOY fixed effect and Region FE- DOY interaction?

2. For the coding part, constrained by my laptop RAM, the replication for table 2 is divided into two parts. The first part only contain column(1) to column(5) of table 2 and the last 3 columns will be show on the last page. Also, I exported RMarkdown as .Tex file and then use Overleaf (an online Latex editor) to beautify the table2. The code I added in the .Tex file is

\{resizebox}{\textwidth}{!}{}

I do not make any other changes to the code and .Tex file. Followings are the coding (replication) part.

Replication1

(2) Tidyverse

```r
# Find out the monitors qualifiied more than25% during summer

qualified = unique((df %>%
  filter(year < 2004 &  pollutionReadings > 8) %>%
  filter(month(date) < 9 & month(date) > 5) %>%
  drop_na(ozone8Hr,ozoneMax) %>%
  filter(ozone8Hr >0 & ozoneMax > 0) %>%
  mutate(summer = if_else(month(date)>5 & month(date)<9, 1 ,0))%>%
  group_by(fips, year, monitorID) %>%
  summarise(sum = sum(summer)) %>%
  filter( sum/92 >= 0.75) %>%
  select(fips,monitorID,year) %>%
  mutate(qualified = paste(year,fips,monitorID)))$qualified)

# The CARB Region
a=levels(as.factor(df$standard_RFG))[2:3]

#Clean the data
df_clean = df %>%
  filter(year < 2004   &  pollutionReadings > 8) %>%
  filter(month(date) < 9 & month(date) > 5) %>%
  mutate(qualification = paste(year,fips,monitorID)) %>%
  filter(qualification %in% qualified &  is.na(neighborCountyTreated)) %>%
  drop_na(ozone8Hr,ozoneMax,urban,income,censusRegion,tMax,tMin,rain,snow,date,year) %>%
  filter(ozone8Hr >0 & ozoneMax > 0) %>%

  #Creat dummies for treatments
  mutate(
    RVP1_dummy=case_when(
      standard_RVP1 >0 & year>=1989 & year <=1991 ~ 1,
      TRUE~0),
    RVP2_dummy = case_when(
      standard_RVP2 <7.9 & year>=1992 ~ 1,
      TRUE ~0
    )) %>%
  mutate(CARB=if_else(standard_RFG %in% a & year >= 1996, 1, 0),
         RFG = case_when(
          standard_RFG %in% a ~ 0,
          is.na(standard_RFG) ~ 0,
          TRUE ~ 1)) %>%

  #create monitor-specific ID and Region-year specific ID
  mutate(monitor=paste(monitorID,fips),Region_year=paste(censusRegion,year))
```

(3) Create Lag Variables and dow/region- dow and region- doy fixed effect

```
df_clean=df_clean %>%
  arrange(monitor,date) %>%
  mutate(tMinL1 = if_else(as.Date(date) > lag(as.Date(date)),lag(tMin), as.double(NA)),
         tMaxL1 = if_else(as.Date(date) > lag(as.Date(date)),lag(tMax), as.double(NA)))


df_clean = df_clean %>%
  mutate(dow=weekdays(date),doy = yday(date))%>%
  mutate(Region_dow=paste(censusRegion,dow),Region_doy=as.factor(censusRegion*doy))
```

(4) Table 1

```
df_summary=df_clean %>%
  mutate(county=paste(state_code,county_code)) %>%
  mutate(urban_dummy=case_when(
    urban==1 ~ 1,
    TRUE ~ 0
  ),rural_dummy= case_when(
    urban==3 ~ 1,
    TRUE ~ 0
  ))  %>%
  group_by(year,fips,monitorID,county_code,state_code) %>%
  summarise(urban_sum=sum(urban_dummy)/n(),
            rural_sum=sum(rural_dummy)/n(),
            RVP1 = sum(RVP1_dummy)/n(),
            RVP2 = sum(RVP2_dummy)/n(),
            RFG = sum(RFG)/n(),
            CARB = sum (CARB)/n()
            ) %>%
  group_by(year,county_code,state_code) %>%
  summarise(urban=sum(urban_sum),
            rural=sum(rural_sum),
            RVP1=sum(RVP1),
            RVP2=sum(RVP2),
            RFG=sum(RFG),
            CARB=sum(CARB)) %>%
  group_by(year) %>%
  summarise(urban=sum(urban),rural=sum(rural),
            RVP1=sum(if_else(RVP1>0,1,0)),
            RVP2=sum(if_else(RVP2>0,1,0)),
            RFG=sum(if_else(RFG>0,1,0)),
            CARB=sum(if_else(CARB>0,1,0)))



df_year=df_clean %>%
  mutate(monitor=paste(fips,monitorID),county=paste(state_code,county_code)) %>%
  group_by(year) %>%
  summarise(Observations = length(unique(paste(date,monitor))),
            counties = length(unique(county)),
            TotalMonitors = length(unique(monitor))) %>%
  inner_join(df_summary,by="year")
```

```
tot = c("Total",sum(df_year$Observations),"-",rep("",7))

avg = c("Average",round(mean(df_year$Observations)),round(mean(df_year$counties)),
        round(mean(df_year$TotalMonitors)),round(mean(df_year$urban)),
        round(mean(df_year$rural)), rep("-",4))

df_year = df_year %>%
  rbind(tot,avg)




table1 = stargazer(df_year,type = "latex",
                   header = FALSE, summary = FALSE,rownames = FALSE)
```

Table 1:

year	Observations	counties	TotalMonitors	urban	rural	RVP1	RVP2	RFG	CARB
1989	63025	417	719	153	243	370	0	0	0
1990	66054	436	750	157	267	381	0	0	0
1991	69174	451	782	151	297	395	0	0	0
1992	69849	452	789	155	300	0	132	0	0
1993	72614	469	815	167	301	0	140	0	0
1994	74449	473	835	163	316	0	140	0	0
1995	77022	477	865	170	330	0	111	111	0
1996	76470	471	854	165	330	0	76	106	47
1997	78286	478	873	166	336	0	76	108	47
1998	79553	487	889	165	344	0	82	108	49
1999	80750	485	899	168	344	0	87	108	49
2000	82467	489	915	178	346	0	97	107	49
2001	83783	490	929	178	355	0	97	108	47
2002	85228	495	943	177	361	0	100	109	49
2003	85261	498	945	180	362	0	101	108	49
Total	1143985	-							
Average	76266	471	853	166	322	-	-	-	-

```
#Create time trend

df_clean = df_clean %>%
  mutate(ltrend = as.numeric(as.Date(date))/365, qtrend = ltrend^2)




df_treatment = df_clean %>%
  group_by(monitor) %>%
  summarise( treatment = paste(max(RVP2_dummy),
                               max(RFG),max(CARB),
                               max(censusRegion)))
```

```
df_clean = left_join(df_clean,df_treatment, by = "monitor")

df_clean$income = df_clean$income/1000000000
```

(5) Table 2

```
model1 = felm(log(ozoneMax)~RVP1_dummy+RVP2_dummy+RFG+CARB|
                Region_year+monitor|0|
                state_code+year,data=df_clean,psdef=FALSE)

model2 = felm(log(ozoneMax)~RVP1_dummy+RVP2_dummy+RFG+CARB+

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                  (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow)|
                Region_year+monitor+Region_dow+Region_doy|0|
                state_code+year,data=df_clean,psdef=FALSE)

model3 = felm(log(ozoneMax)~RVP1_dummy+RVP2_dummy+RFG+CARB+ income +

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow)|
                Region_year+monitor+Region_dow+Region_doy|0|
                state_code+year,data=df_clean,psdef=FALSE)

model4 =  felm(log(ozoneMax)~RVP1_dummy+RVP2_dummy+RFG+CARB+ income +

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow) + as.factor(treatment)*ltrend|
                Region_year+monitor+Region_dow+Region_doy|0|
```

```
                state_code+year,data=df_clean,psdef=FALSE)

model5 =   felm(log(ozoneMax)~RVP1_dummy+RVP2_dummy+RFG+CARB+ income +

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow) +
                as.factor(treatment)*ltrend+
                as.factor(treatment)*qtrend|
                Region_year+monitor+Region_dow+Region_doy|0|
                state_code+year,data=df_clean,psdef=FALSE)


  stargazer(model1,model2,model3,model4,model5, header = FALSE,
            omit = c(6:135),type="latex",
            add.lines =
              list(c("Monitor FEs", "Yes" , "Yes" ,"Yes", "Yes","Yes" ),
                  c("Region-year FEs", "Yes","Yes", "Yes", "Yes","Yes" ) ,
                  c("Region-DOW FEs"," No ","Yes","Yes" ,"Yes","Yes" ),
                  c("Region FEs- DOY interaction", "No","Yes" ,"Yes" ,"Yes","Yes" ),
                  c("Weather Controls","No", "Yes","Yes","Yes","Yes" ),
                  c("Income" ,"No" ,"No" ,"Yes" ,"Yes" ,"Yes"),
                  c("Regulation- Region trends" ,"No" ," No" ,"No" , "Yes","Yes" ),
                  c("Regulation- Region quad trends" ,"No" ,"No" ,"No" , "No","Yes" )))
```

(6) Figure 6(b)

```
# Qualified monitors

qualified = unique((df %>%

          #drop non-observed samples

  filter(year < 2004 &  pollutionReadings > 8) %>%
  drop_na(ozone8Hr,ozoneMax) %>%
  filter(ozone8Hr >0 & ozoneMax > 0) %>%

  mutate(#create seson variable
          season =
           case_when(

            month(date) >= 3 & month(date) <= 5 ~ 1,
            month(date) >= 6 & month(date) <= 8 ~ 2,
            month(date) >= 9 & month(date) <= 11 ~ 3,
            month(date) <= 2 | month(date) == 12 ~ 4
```

Table 2:

	Dependent variable:				
	log(ozoneMax)				
	(1)	(2)	(3)	(4)	(5)
RVP1_dummy	0.016	0.013	0.014	0.003	0.012
	(0.017)	(0.019)	(0.019)	(0.019)	(0.019)
RVP2_dummy	−0.007	−0.011**	−0.007	−0.011	−0.009
	(0.008)	(0.006)	(0.007)	(0.010)	(0.011)
RFG	−0.029**	−0.030***	−0.016	−0.036***	−0.017
	(0.012)	(0.010)	(0.013)	(0.012)	(0.015)
CARB	−0.094***	−0.090***	−0.077***	−0.068***	−0.067***
	(0.015)	(0.014)	(0.015)	(0.019)	(0.022)
income			−1.381**	−0.204	−0.279
			(0.660)	(0.140)	(0.189)
Monitor FEs	Yes	Yes	Yes	Yes	Yes
Region-year FEs	Yes	Yes	Yes	Yes	Yes
Region-DOW FEs	No	Yes	Yes	Yes	Yes
Region FEs- DOY interaction	No	Yes	Yes	Yes	Yes
Weather Controls	No	Yes	Yes	Yes	Yes
Income	No	No	Yes	Yes	Yes
Regulation- Region trends	No	No	No	Yes	Yes
Regulation- Region quad trends	No	No	No	No	Yes
Observations	1,143,985	1,142,656	1,142,656	1,142,656	1,142,656
R^2	0.316	0.489	0.490	0.492	0.492
Adjusted R^2	0.315	0.488	0.489	0.491	0.491
Residual Std. Error	0.336 (df = 1142471)	0.290 (df = 1140700)	0.290 (df = 1140699)	0.289 (df = 1140683)	0.289 (df = 1140664)
Note:					*p<0.1; **p<0.05; ***p<0.01

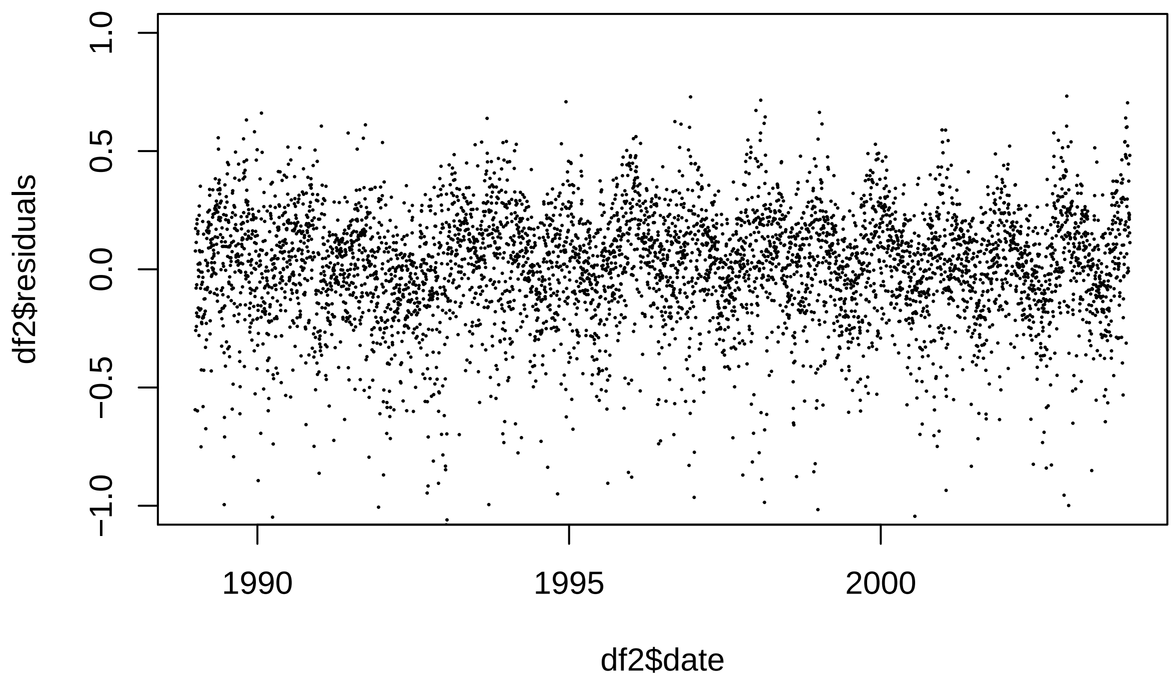
```
          ),

        # count the number of days in each season
         day_count =
          case_when(

            month(date) >= 3 & month(date) <= 5 ~ 92,
            month(date) >= 6 & month(date) <= 8 ~ 92,
            month(date) >= 9 & month(date) <= 11 ~ 91,
            year %% 4 == 3 & month(date) == 12 ~ 91,
            year %% 4 == 0  & month(date) <= 2 ~ 91,
            month(date) <= 2 & month(date) == 12 ~90

         )) %>%
            # drop valid readings of season- monitor for smaller than 25 %

  group_by(fips, year, monitorID, season, day_count) %>%
  summarise(total_day = n() ) %>%
  filter( total_day/day_count >= 0.75) %>%
  ungroup() %>%

            # drop monitors  less than 25 % of seasons

  group_by(fips, monitorID) %>%
  summarise(total_season = n()) %>%
    ungroup() %>%
  filter( total_season/60 >=0.75) %>%
```

```
          # create qualified ID

  dplyr::select(fips,monitorID) %>%
  mutate(qualified = paste(fips,monitorID)))$qualified)

# Camden County site 1001
qualified[94]
```

```
## [1] "34007 1001"
```

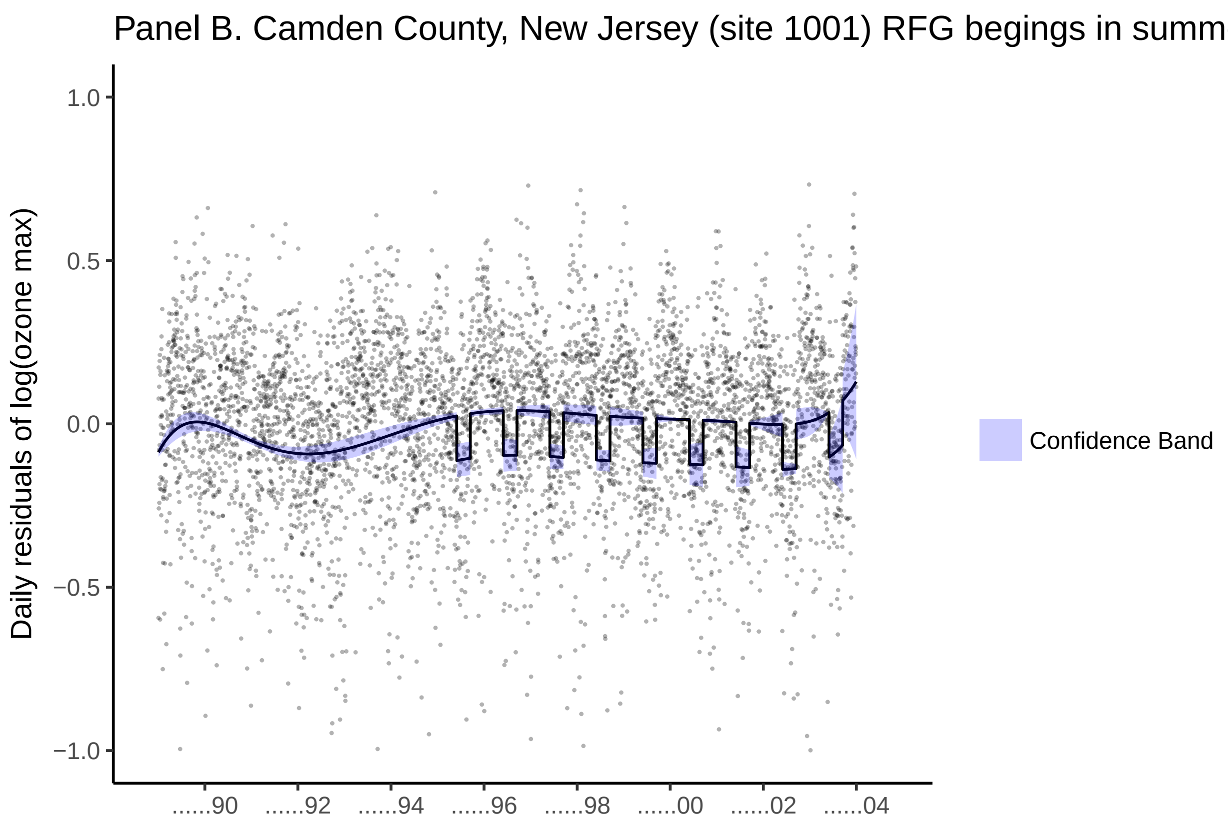
```
df2 = df %>%
  filter(year < 2004)%>%
  mutate(qualification = paste(fips,monitorID)) %>%
  filter(qualification == qualified[94]) %>%
  mutate(
    RVP1_dummy=case_when(
      standard_RVP1 >0 & year>=1989 & year <=1991 ~ 1,
      TRUE~0),
    RVP2_dummy = case_when(
      standard_RVP2 <7.9 & year>=1992 ~ 1,
      TRUE ~0
    )) %>%
  mutate(CARB=if_else(state_code == 6 & year >= 1996, 1, 0),
         RFG = case_when(
          standard_RFG %in% a ~ 0,
          is.na(standard_RFG) ~ 0,
          TRUE ~ 1))


df2 = df2 %>%
  filter(year < 2004   &  pollutionReadings > 8) %>%
  drop_na(ozone8Hr,ozoneMax,urban,income,censusRegion,tMax,tMin,rain,snow,date,year) %>%
  filter(ozone8Hr >0 & ozoneMax > 0) %>%
  mutate(#create seson variable
          season =
           case_when(

            month(date) >= 3 & month(date) <= 5 ~ 1,
            month(date) >= 6 & month(date) <= 8 ~ 2,
            month(date) >= 9 & month(date) <= 11 ~ 3,
            month(date) <= 2 | month(date) == 12 ~ 4

           ),
          trend = as.numeric(date)/365,
          max_date = max(trend),
          min_date = min(trend),
          z=2*(trend - min_date)/(max_date - min_date)-1) %>%
  mutate(
    time1 = z,
         time2 = 2*z^2 - 1,
         time3 = 4*z^3 - 3*z,
         time4 = 2*z*time3 - time2,
         time5 = 2*z*time4 - time3,
         time6 = 2*z*time5 - time4,
```

```
         time7 = 2*z*time6 - time5,
         time8 = 2*z*time7 - time6
  ) %>%
  arrange(date) %>%
  mutate(tMinL1 = if_else(as.Date(date) > lag(as.Date(date)),lag(tMin), as.double(NA)),
         tMaxL1 = if_else(as.Date(date) > lag(as.Date(date)),lag(tMax), as.double(NA))) %>%
  mutate(dow=weekdays(date),
         doy = yday(date),
         year_season = paste(year,season),
         season_dow = paste(season,dow))



# Running the RD regression
model_rd <- felm(log(ozoneMax) ~ RFG +
              # Chebychev time polynomial
              time1 + time2 + time3 + time4 + time5 + time6 + time7 + time8 +
              # Dow inter
              (tMax+tMin+rain+snow)*I(dow) +
              # Quarter interactions
            (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
            rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*as.factor(season)|
              # Fixed effects
              dow + month(date) + season + season_dow |0|
            year_season, data = df2)


beta = model_rd$coefficients[1:9]

df2  = df2 %>%
  mutate(fit_value = beta[1]*RFG +
            beta[2]*time1+
            beta[3]*time2+
            beta[4]*time3+
             beta[5]*time4+
            beta[6]*time5+
            beta[7]*time6+
            beta[8]*time7+
            beta[9]*time8) %>%
  filter(!is.na(tMinL1)) %>%
  mutate(residuals = resid(model_rd)+fit_value)

mean_resid = mean(df2$residuals)
mean_fit = mean(df2$fit_value)

df2$residuals =df2$residuals-mean_resid

plot(x=df2$date,y=df2$residuals,cex=0.1,ylim=c(-1,1))
```




```
beta_se = summary(model_rd)$coef[c(1:9),2]


df2 = df2 %>%
  mutate(fit_upper =  (beta[1]+1.96*beta_se[1])*RFG +
            (beta[2]+1.96*beta_se[2])*time1+
            (beta[3]+1.96*beta_se[3])*time2+
            (beta[4]+1.96*beta_se[4])*time3+
            (beta[5]+1.96*beta_se[5])*time4+
            (beta[6]+1.96*beta_se[6])*time5+
            (beta[7]+1.96*beta_se[7])*time6+
            (beta[8]+1.96*beta_se[8])*time7+
            (beta[9]+1.96*beta_se[9])*time8,
fit_lower = (beta[1]-1.96*beta_se[1])*RFG +
            (beta[2]-1.96*beta_se[2])*time1+
            (beta[3]-1.96*beta_se[3])*time2+
            (beta[4]-1.96*beta_se[4])*time3+
            (beta[5]-1.96*beta_se[5])*time4+
            (beta[6]-1.96*beta_se[6])*time5+
            (beta[7]-1.96*beta_se[7])*time6+
            (beta[8]-1.96*beta_se[8])*time7+
            (beta[9]-1.96*beta_se[9])*time8)


#Plot Graph6(b)
ggplot(data = df2,aes(x = date))+
```

```
  geom_point(aes(y = residuals), alpha = 0.3,size = 0.1)+
  geom_line(aes(y = fit_value))+
  geom_ribbon(aes(ymax = fit_upper, ymin = fit_lower,fill ="Confidence Band"),
              alpha = 0.2)+
  scale_y_continuous(limits = c(-1,1))+
  scale_x_date(labels = date_format( "%b%y") ,date_breaks = "24 months",
               limits = c(as.Date("1988-11-01"),as.Date("2004-11-01")))+
  labs(x="",
       y="Daily residuals of log(ozone max)",
       title =
         "Panel B. Camden County, New Jersey (site 1001) RFG begings in summer 1995" )+
  theme_classic()+
  scale_fill_manual(values = c("Confidence Band" = "blue" ),name = "")
```



Replication Table2 Last 3 Columns

(5) Table 2

```
model6 = felm(log(ozone8Hr)~RVP1_dummy+RVP2_dummy+RFG+CARB|
                Region_year+monitor|0|
                state_code+year,data=df_clean,psdef=FALSE)

model7 = felm(log(ozone8Hr)~RVP1_dummy+RVP2_dummy+RFG+CARB+

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                  (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow)|
                Region_year+monitor+Region_dow+Region_doy|0|
                state_code+year,data=df_clean,psdef=FALSE)

model8 = felm(log(ozone8Hr)~RVP1_dummy+RVP2_dummy+RFG+CARB+income+

                poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1+



                  (poly(tMax,3)+poly(tMin,3)+tMax*tMin+poly(rain,2)+poly(snow,2)+
                rain*tMax+rain*tMaxL1+rain*tMinL1+tMax*tMaxL1+tMax*tMinL1)*doy+


                (tMax+tMin+rain+snow)*I(dow)+ as.factor(treatment)*ltrend+as.factor(treatment)*qtrend|
                Region_year+monitor+Region_dow+Region_doy|0|
                state_code+year,data=df_clean,psdef=FALSE)

  stargazer(model6,model7,model8, header = FALSE,
            omit = c(6:135),type="latex",
            add.lines =
              list(c("Monitor FEs", "Yes" , "Yes" ,"Yes" ),
                  c("Region-year FEs", "Yes","Yes", "Yes") ,
                  c("Region-DOW FEs"," No ","Yes","Yes" ),
                  c("Region FEs- DOY interaction", "No","Yes" ,"Yes" ),
                  c("Weather Controls","No", "Yes","Yes" ),
                  c("Income" ,"No" ,"No" ,"Yes" ),
                  c("Regulation- Region trends" ,"No" ," No" ,"Yes" ),
                  c("Regulation- Region quad trends" ,"No" ,"No" ,"Yes" )))
```

Table 1:

	Dependent variable:		
	log(ozone8Hr)		
	(1)	(2)	(3)
RVP1_dummy	0.018	0.015	0.012
	(0.021)	(0.022)	(0.023)
RVP2_dummy	−0.005	−0.009*	−0.009
	(0.007)	(0.005)	(0.010)
RFG	−0.028**	−0.028**	−0.020
	(0.013)	(0.012)	(0.015)
CARB	−0.090***	−0.086***	−0.067***
	(0.015)	(0.016)	(0.020)
income			−0.083
			(0.167)
Monitor FEs	Yes	Yes	Yes
Region-year FEs	Yes	Yes	Yes
Region-DOW FEs	No	Yes	Yes
Region FEs- DOY interaction	No	Yes	Yes
Weather Controls	No	Yes	Yes
Income	No	No	Yes
Regulation- Region trends	No	No	Yes
Regulation- Region quad trends	No	No	Yes
Observations	1,143,985	1,142,656	1,142,656
R^2	0.328	0.494	0.497
Adjusted R^2	0.327	0.493	0.496
Residual Std. Error	0.329 (df = 1142471)	0.286 (df = 1140700)	0.285 (df = 1140664)
Note:			*p<0.1; **p<0.05; ***p<0.01