Introduction to the r4moves Package

Joseph Jakuta

MARAMA Mobile Source Workshop

March 18, 2021

DEPARTMENT OF ENERGY & ENVIRONMENT

WE ARE WASHINGTON DC

GOVERNMENT OF THE DISTRICT OF COLUMBIA

MURIEL BOWSER, MAYOR

Purpose

- To more easily:
  - Access and analyze data from the MOVES model using R
  - Manipulate MOVES inputs
  - Create and manipulate Runspecs and Batchfiles
  - Document the MOVES workflow
- Will be done "agilely" as is typical in open source

The end goal is to be able to complete all of your work in MOVES without ever leaving R

Problems r4moves Hope To Solve

1. Need to recreate and rewrite SQL queries to get MOVES data
2. Switching between analysis platforms
3. Allowing easier use of R data frames and R packages
4. Solid and replicable documentation

```
.widget-area-sidebar [illegible]
    font-size: 13px;
}


/* =Menu
-------------------------------------------------

#access {
    display: inline-block;
    height: 69px;
    float: right;
    margin: 11px 28px 0px 0px;
    max-width: 800px;
}

#access ul {
    font-size: 13px;
    list-style: none;
    margin: 0 0 0 -0.8125em;
    padding-left: 0;
    z-index: 99999;
    text-align: right;
}

#access li {
    display: inline-block;
    text-align: left;
```

Accessing and Installing

- Locations:
  - Currently on Github - https://github.com/hlinak/R-for-MOVES
  - CRAN planned for future
    - Want to get a little bit more feedback and make sure the code works with MOVES3 before putting it up on CRAN

What DOEE Has Used It For

Review of 2017 NEI Mobile Source Inventory Inputs

Development of Emission Rates Based on Local Data

DC Analysis of OTC Aftermarket Converter Model Rule Implementation

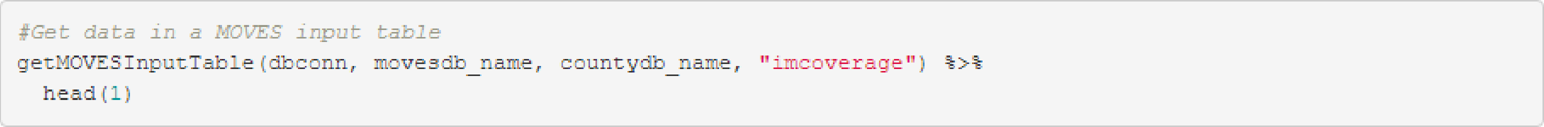
110(l) Demonstration for I/M Program Changes

Examples

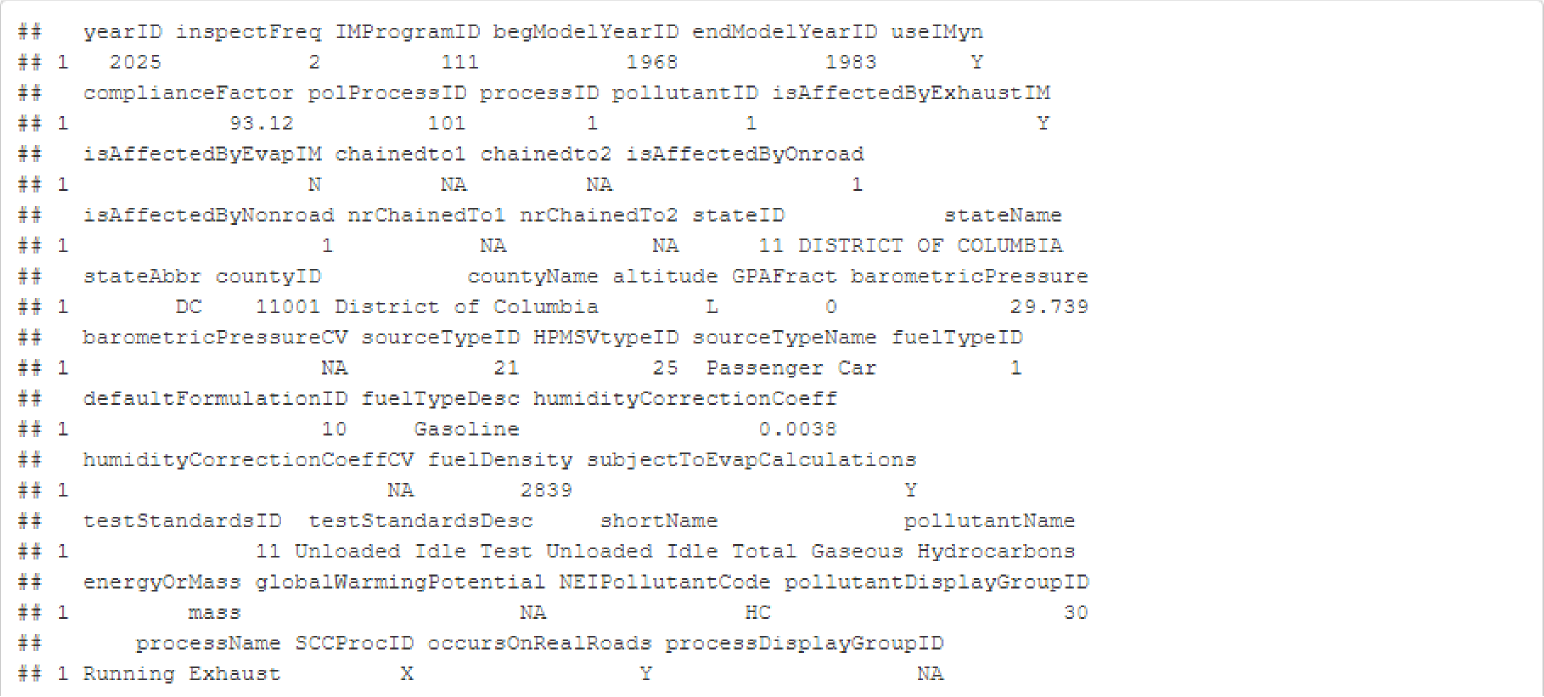
- Over the next 6 Slides will be example code
- I picked some of the more interesting examples from the I/M 110(I) project
- I skipped many steps!
  - If you want to actually be able to run something like this I have more detailed slides and can send them along

Example: Accessing Input Table

We are going to look at the imcoverage table. Notice the additional, joined information.



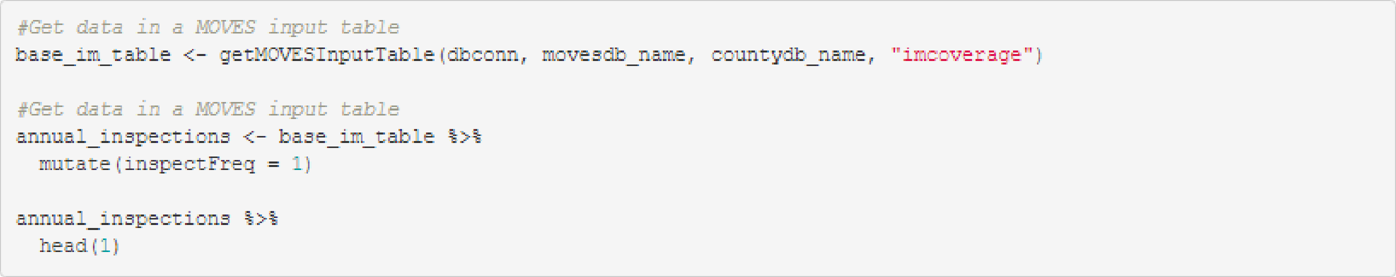
```
#Get data in a MOVES input table
getMOVESInputTable(dbconn, movesdb_name, countydb_name, "imcoverage") %>%
  head(1)
```



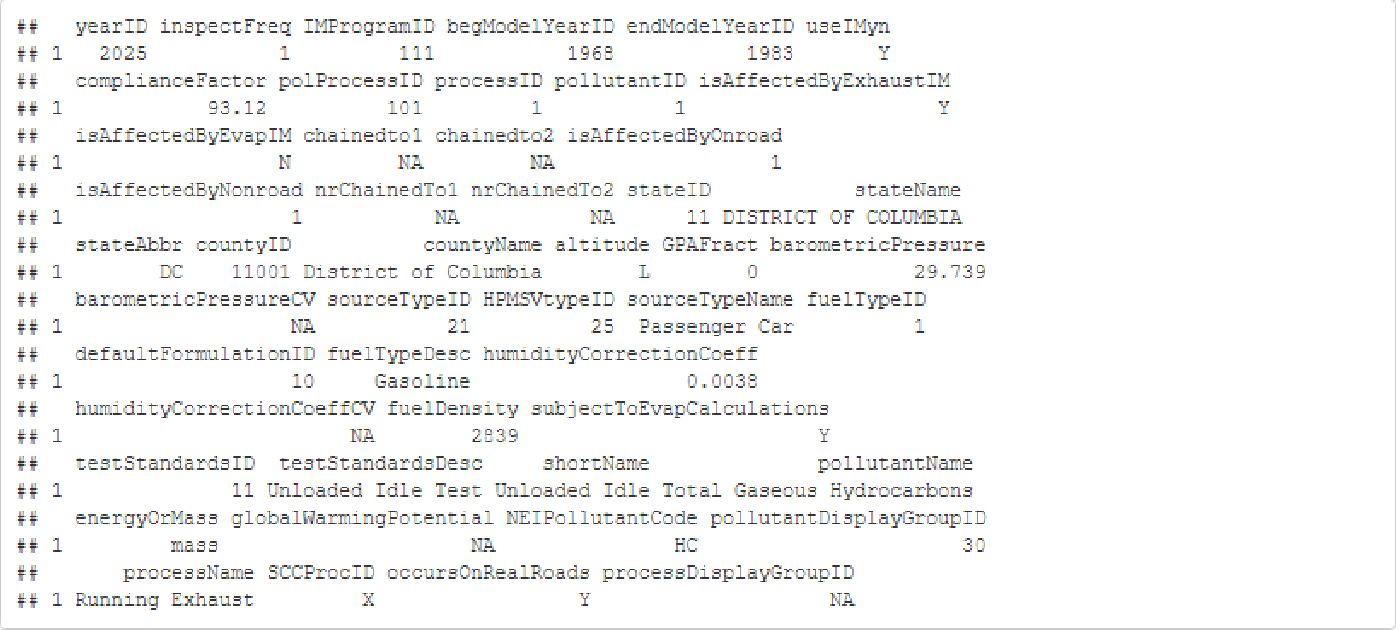
```
##   yearID inspectFreq IMProgramID begModelYearID endModelYearID useIMyn
## 1   2025           2         111           1968           1983       Y
##   complianceFactor polProcessID processID pollutantID isAffectedByExhaustIM
## 1            93.12          101         1           1                     Y
##   isAffectedByEvapIM chainedto1 chainedto2 isAffectedByOnroad
## 1                  N         NA         NA                  1
##   isAffectedByNonroad nrChainedTo1 nrChainedTo2 stateID            stateName
## 1                   1           NA           NA      11 DISTRICT OF COLUMBIA
##   stateAbbr countyID           countyName altitude GPAFract barometricPressure
## 1        DC    11001 District of Columbia        L        0             29.739
##   barometricPressureCV sourceTypeID HPMSVtypeID sourceTypeName fuelTypeID
## 1                   NA           21          25  Passenger Car          1
##   defaultFormulationID fuelTypeDesc humidityCorrectionCoeff
## 1                   10     Gasoline                  0.0038
##   humidityCorrectionCoeffCV fuelDensity subjectToEvapCalculations
## 1                        NA        2839                         Y
##   testStandardsID  testStandardsDesc     shortName              pollutantName
## 1              11 Unloaded Idle Test Unloaded Idle Total Gaseous Hydrocarbons
##   energyOrMass globalWarmingPotential NEIPollutantCode pollutantDisplayGroupID
## 1         mass                     NA               HC                      30
##       processName SCCProcID occursOnRealRoads processDisplayGroupID
## 1 Running Exhaust         X                 Y                    NA
```

Example: Manipulating Input Table

Here we are going to increase the frequency of I/M tests in the inputs



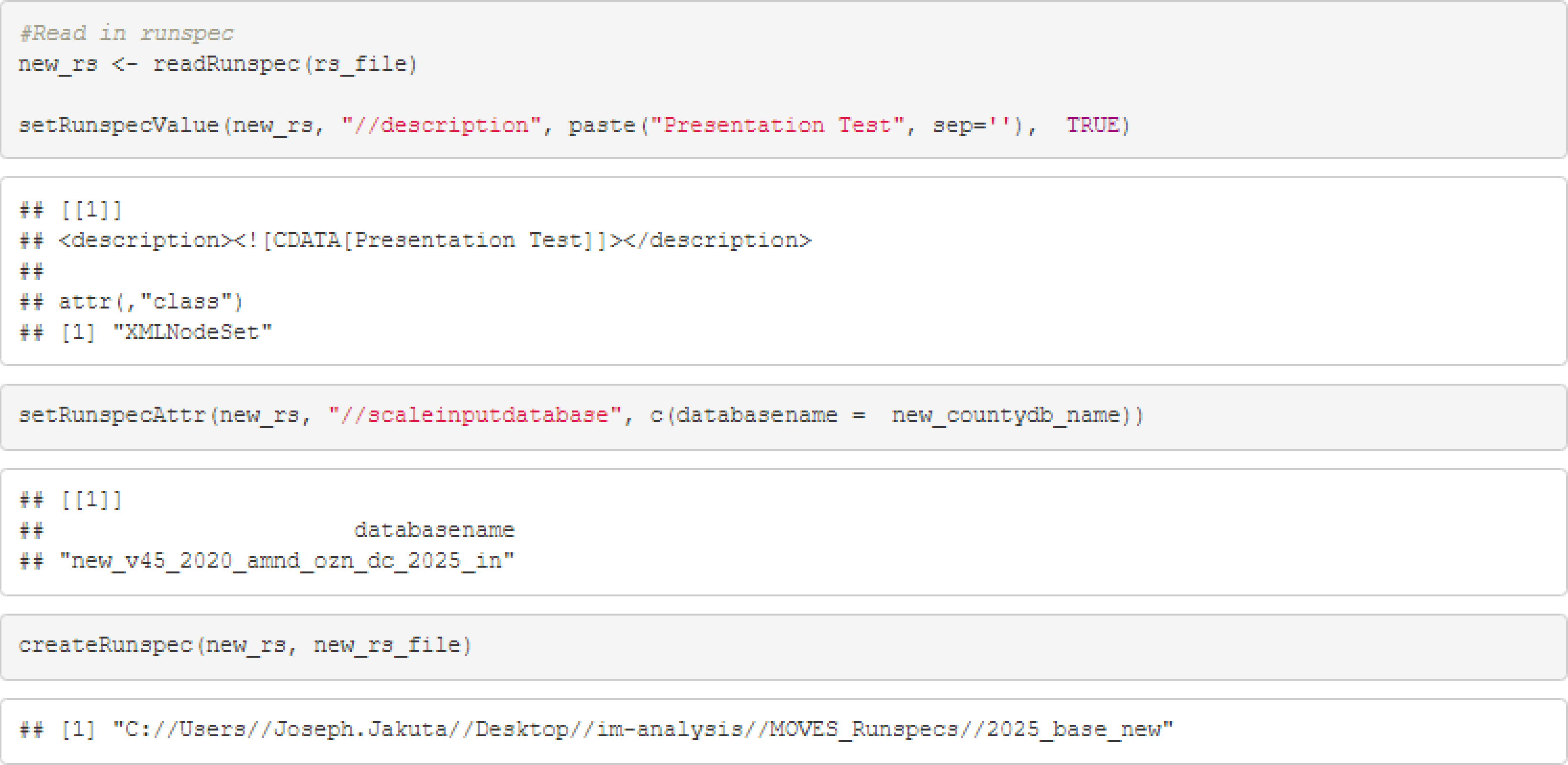
```r
#Get data in a MOVES input table
base_im_table <- getMOVESInputTable(dbconn, movesdb_name, countydb_name, "imcoverage")

#Get data in a MOVES input table
annual_inspections <- base_im_table %>%
  mutate(inspectFreq = 1)

annual_inspections %>%
  head(1)
```



```
##   yearID inspectFreq IMProgramID begModelYearID endModelYearID useIMyn
## 1   2025           1         111           1968           1983       Y
##   complianceFactor polProcessID processID pollutantID isAffectedByExhaustIM
## 1            93.12          101         1           1                     Y
##   isAffectedByEvapIM chainedto1 chainedto2 isAffectedByOnroad
## 1                  N         NA         NA                  1
##   isAffectedByNonroad nrChainedTo1 nrChainedTo2 stateID            stateName
## 1                   1           NA           NA      11 DISTRICT OF COLUMBIA
##   stateAbbr countyID           countyName altitude GPAFract barometricPressure
## 1        DC    11001 District of Columbia        L        0             29.739
##   barometricPressureCV sourceTypeID HPMSVtypeID sourceTypeName fuelTypeID
## 1                   NA           21          25  Passenger Car          1
##   defaultFormulationID fuelTypeDesc humidityCorrectionCoeff
## 1                   10     Gasoline                  0.0038
##   humidityCorrectionCoeffCV fuelDensity subjectToEvapCalculations
## 1                        NA        2839                         Y
##   testStandardsID  testStandardsDesc     shortName              pollutantName
## 1              11 Unloaded Idle Test Unloaded Idle Total Gaseous Hydrocarbons
##   energyOrMass globalWarmingPotential NEIPollutantCode pollutantDisplayGroupID
## 1         mass                     NA               HC                      30
##       processName SCCProcID occursOnRealRoads processDisplayGroupID
## 1 Running Exhaust         X                 Y                    NA
```

Example: Manipulating Runspec

Now we are going to make a new runspec that calls a new County DB, created in a missing step to include the new I/M inputs



```
#Read in runspec
new_rs <- readRunspec(rs_file)

setRunspecValue(new_rs, "//description", paste("Presentation Test", sep=''),  TRUE)
```

```
## [[1]]
## <description><![CDATA[Presentation Test]]></description>
##
## attr(,"class")
## [1] "XMLNodeSet"
```

```
setRunspecAttr(new_rs, "//scaleinputdatabase", c(databasename =  new_countydb_name))
```

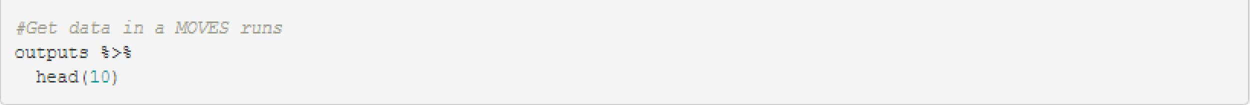
```
## [[1]]
##                          databasename
## "new_v45_2020_amnd_ozn_dc_2025_in"
```

```
createRunspec(new_rs, new_rs_file)
```

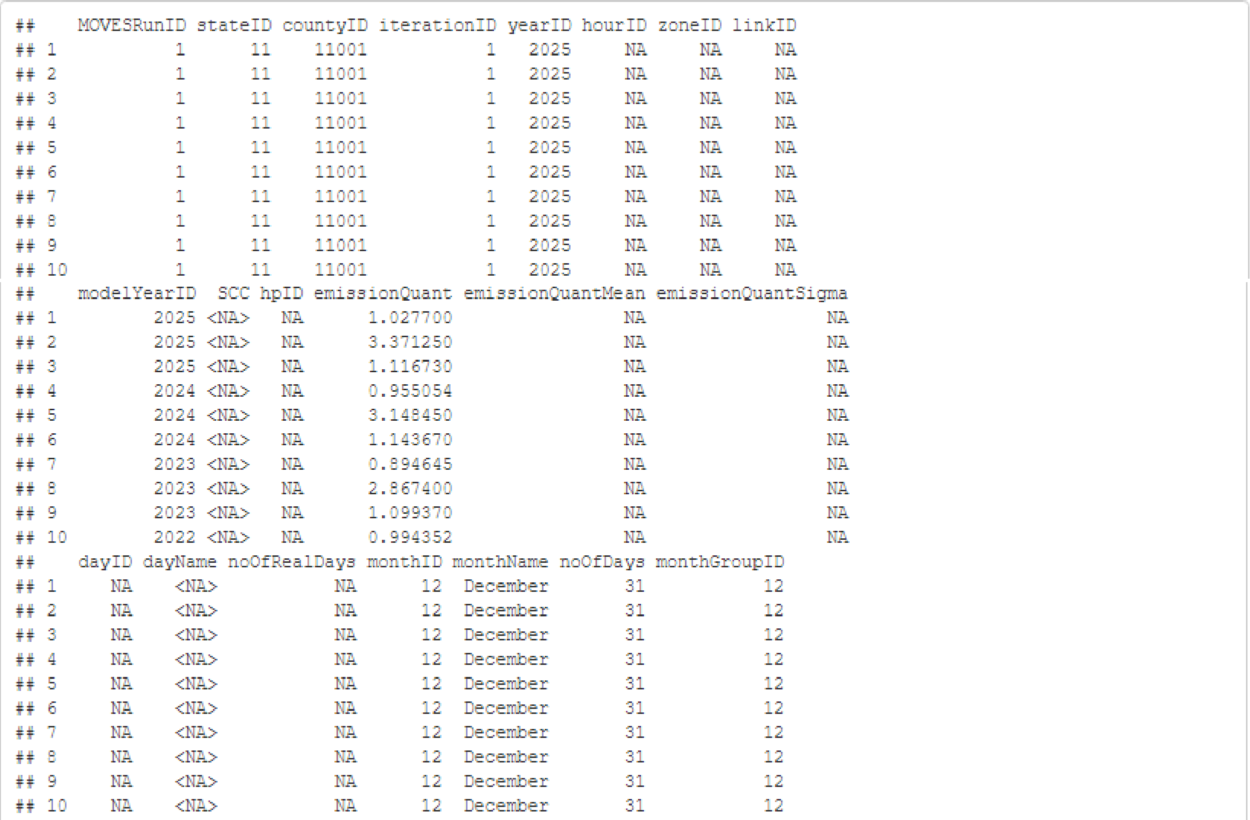
```
## [1] "C://Users//Joseph.Jakuta//Desktop//im-analysis//MOVES_Runspecs//2025_base_new"
```

Example: Output Data

Now let’s look at some output data



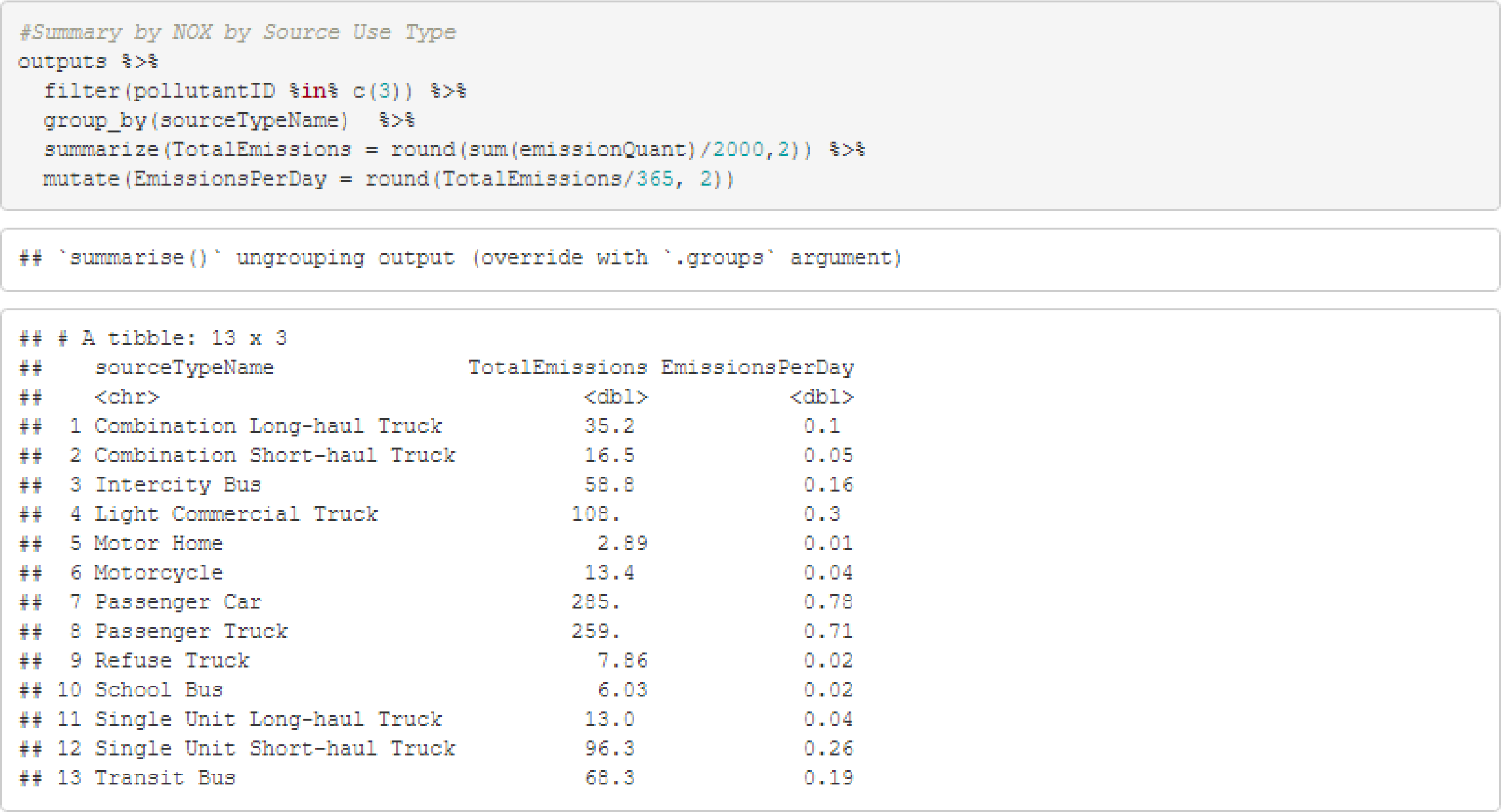
```r
#Get data in a MOVES runs
outputs %>%
  head(10)
```



```
##    MOVESRunID stateID countyID iterationID yearID hourID zoneID linkID
## 1           1      11    11001           1   2025     NA     NA     NA
## 2           1      11    11001           1   2025     NA     NA     NA
## 3           1      11    11001           1   2025     NA     NA     NA
## 4           1      11    11001           1   2025     NA     NA     NA
## 5           1      11    11001           1   2025     NA     NA     NA
## 6           1      11    11001           1   2025     NA     NA     NA
## 7           1      11    11001           1   2025     NA     NA     NA
## 8           1      11    11001           1   2025     NA     NA     NA
## 9           1      11    11001           1   2025     NA     NA     NA
## 10          1      11    11001           1   2025     NA     NA     NA
##    modelYearID  SCC hpID emissionQuant emissionQuantMean emissionQuantSigma
## 1         2025 <NA>   NA      1.027700                NA                 NA
## 2         2025 <NA>   NA      3.371250                NA                 NA
## 3         2025 <NA>   NA      1.116730                NA                 NA
## 4         2024 <NA>   NA      0.955054                NA                 NA
## 5         2024 <NA>   NA      3.148450                NA                 NA
## 6         2024 <NA>   NA      1.143670                NA                 NA
## 7         2023 <NA>   NA      0.894645                NA                 NA
## 8         2023 <NA>   NA      2.867400                NA                 NA
## 9         2023 <NA>   NA      1.099370                NA                 NA
## 10        2022 <NA>   NA      0.994352                NA                 NA
##    dayID dayName noOfRealDays monthID monthName noOfDays monthGroupID
## 1     NA    <NA>           NA      12  December       31           12
## 2     NA    <NA>           NA      12  December       31           12
## 3     NA    <NA>           NA      12  December       31           12
## 4     NA    <NA>           NA      12  December       31           12
## 5     NA    <NA>           NA      12  December       31           12
## 6     NA    <NA>           NA      12  December       31           12
## 7     NA    <NA>           NA      12  December       31           12
## 8     NA    <NA>           NA      12  December       31           12
## 9     NA    <NA>           NA      12  December       31           12
## 10    NA    <NA>           NA      12  December       31           12
```

Example: Making Summaries

Now we use our data frame containing to MOVES data with typical R functions to make summaries



```
#Summary by NOX by Source Use Type
outputs %>%
  filter(pollutantID %in% c(3)) %>%
  group_by(sourceTypeName)  %>%
  summarize(TotalEmissions = round(sum(emissionQuant)/2000,2)) %>%
  mutate(EmissionsPerDay = round(TotalEmissions/365, 2))
```

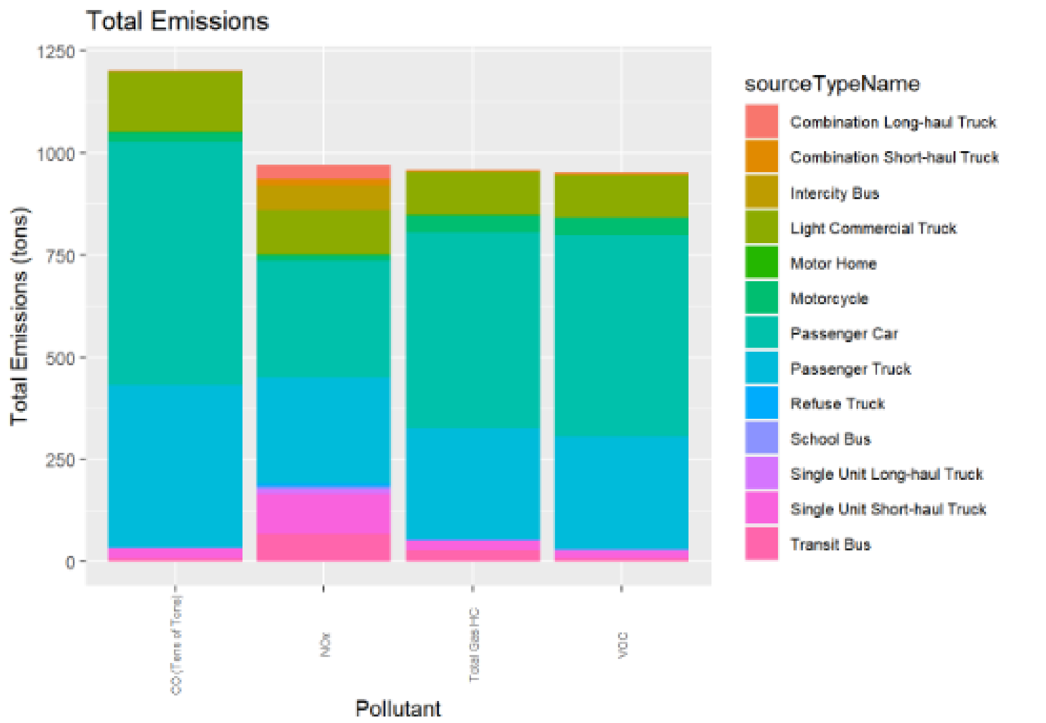
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 13 x 3
##    sourceTypeName               TotalEmissions EmissionsPerDay
##    <chr>                                 <dbl>           <dbl>
##  1 Combination Long-haul Truck           35.2             0.1
##  2 Combination Short-haul Truck          16.5             0.05
##  3 Intercity Bus                         58.8             0.16
##  4 Light Commercial Truck               108.              0.3
##  5 Motor Home                             2.89            0.01
##  6 Motorcycle                            13.4             0.04
##  7 Passenger Car                        285.              0.78
##  8 Passenger Truck                      259.              0.71
##  9 Refuse Truck                           7.86            0.02
## 10 School Bus                             6.03            0.02
## 11 Single Unit Long-haul Truck           13.0             0.04
## 12 Single Unit Short-haul Truck          96.3             0.26
## 13 Transit Bus                           68.3             0.19
```

Example: Making Plots

Now we use our data frame containing to MOVES data with typical R functions to make plots

```
outputs %>%
  filter(pollutantID %in% c(87,1,2,3)) %>%
  group_by(shortName, sourceTypeName)  %>%
  summarize(TotalEmissions = round(sum(emissionQuant)/2000,2),
            pollutantID = max(pollutantID),
            sourceTypeID = max(sourceTypeID)) %>%
  mutate(TotalEmissions = ifelse(pollutantID == 2, TotalEmissions/10, TotalEmissions),
         shortName = ifelse(pollutantID == 2, paste(shortName, "(Tens of Tons)"), shortName)) %>%
  ggplot(aes(x = shortName, y = TotalEmissions , fill=sourceTypeName)) +
  geom_col() +
  labs(title = "Total Emissions",
       x = "Pollutant",
       y = "Total Emissions (tons)") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size=6),
        legend.text = element_text(size=8) )
```



Next Steps (by priority)

Get more people trying it out!

Update to work with MOVES3

- Work with MariaDB – Done!
- Update new DB Structures
  - Input Tables Done!
  - Output Tables, Next
- QA
- Update to new CLI

Put on CRAN

Fix MOVES run bugs

Add NONROAD output functions

Possibly speed up getMOVESOutput on large runs

Considering additional Runspec manipulation functions

Update to work with older versions of MOVES

Questions?

joseph.jakuta@dc.gov