Categories

1. [Advanced Modeling](https://datascienceplus.com/category/advanced-modeling/)

Tags

1. [Data Manipulation](https://datascienceplus.com/tag/data-manipulation/)
2. [Data Visualisation](https://datascienceplus.com/tag/data-visualisation/)
3. [ggplot2](https://datascienceplus.com/tag/ggplot2/)
4. [R Programming](https://datascienceplus.com/tag/rstats/)

The motivation for choosing this data set to explore is straightforward: I wanted a real-world data set for which additional meta data would be easy to find, making NYC data perfect; I wanted something which did not require specialised domain knowledge, and so buses are a good choice; and I wanted a data set that might produce some interesting insights. As an additional bonus, this data set had very few downloads or kernels on Kaggle, so it seemed like mostly untrodden ground.

**Data loading and initial exporation**

Data are available from [the NYC open data site](https://data.cityofnewyork.us/Transportation/Bus-Breakdown-and-Delays/ez4e-fazm). Linked is the main data page, which contains the bus breakdown ID number, as well as the route number, the schools serviced, the borough through which the bus travels, and so on.

The readme file included in this bundle also contains references to linking information. Specifically, the [drivers and attendants](https://data.cityofnewyork.us/Transportation/Drivers-and-Attendants/4tqt-y424), [routes](https://data.cityofnewyork.us/Transportation/Routes/8yac-vygm), [transportation sites](https://data.cityofnewyork.us/Transportation/Transportation-Sites/hg3c-2jsy), [vehicles](https://data.cityofnewyork.us/Transportation/Vehicles/28rh-vpvr), [pre-k riders by transportation site](https://data.cityofnewyork.us/Transportation/PreK-Riders-by-Transportation-Site/kjgh-ywbx), [routes by transportation site](https://data.cityofnewyork.us/Transportation/Routes-by-Transportation-Sites/fgqh-vfzz), and [pre-k vendors by transportation site](https://data.cityofnewyork.us/Transportation/PreK-Vendors-by-Transportation-Site/8wau-idzf). All of this data can be combined (see later) to provide route- and vendor- level data to the main delay data set. For example, to quanitfy the total number of drivers employed by each company, the total number of students or schools that they service, and so on.

Our goal is to be able to make some kind of useful prediction about how long a bus is going to be delayed, at the time its delay is called into the operations centre. So we take a look at our primary 'ii\_' data set.

in\_csv <- "../data/II\_Bus\_Breakdown\_and\_Delays.csv"

ii\_breakdowns <- read\_csv(in\_csv)

ii\_breakdowns %>% glimpse

*## Observations: 259,637*

*## Variables: 21*

*## $ School\_Year "2015-2016", "2015-2016", "201...*

*## $ Busbreakdown\_ID 1212699, 1212700, 1212701, 121...*

*## $ Run\_Type "Special Ed AM Run", "Special ...*

*## $ Bus\_No "48186", "2518", "235", "2102"...*

*## $ Route\_Number "N758", "L530", "K168", "K216"...*

*## $ Reason "Other", "Mechanical Problem",...*

*## $ Schools\_Serviced "75485", "21854", "18366", "21...*

*## $ Occurred\_On 2015-09-02 06:27:00, 2015-09-...*

*## $ Created\_On 2015-09-02 06:29:00, 2015-09-...*

*## $ Boro "Nassau County", "Brooklyn", "...*

*## $ Bus\_Company\_Name "BORO TRANSIT, INC.", "RELIANT...*

*## $ How\_Long\_Delayed "25 minutes", NA, "30MINS", "2...*

*## $ Number\_Of\_Students\_On\_The\_Bus 0, 0, 0, 1, 0, 0, 0, 9, 0, 2, ...*

*## $ Has\_Contractor\_Notified\_Schools "Yes", "Yes", "Yes", "Yes", "Y...*

*## $ Has\_Contractor\_Notified\_Parents "No", "Yes", "Yes", "Yes", "Ye...*

*## $ Have\_You\_Alerted\_OPT "No", "Yes", "No", "No", "No",...*

*## $ Informed\_On 2015-09-02 06:29:00, 2015-09-...*

*## $ Incident\_Number NA, NA, NA, NA, NA, NA, NA, NA...*

*## $ Last\_Updated\_On 2015-09-02 06:29:16, 2015-09-...*

*## $ Breakdown\_or\_Running\_Late "Running Late", "Breakdown", "...*

*## $ School\_Age\_or\_PreK "School-Age", "School-Age", "S...*

ii\_breakdowns %>% distinct(How\_Long\_Delayed) %>% head(n=20)

*## # A tibble: 20 x 1*

*## How\_Long\_Delayed*

*##*

*## 1 25 minutes*

*## 2*

*## 3 30MINS*

*## 4 20 min*

*## 5 30 min*

*## 6 90 MIN*

*## 7 20 MINS*

*## 8 20*

*## 9 30 mins*

*## 10 15 MINS*

*## 11 20 MIN*

*## 12 15 mins*

*## 13 30 MINS*

*## 14 25*

*## 15 0*

*## 16 10 mins*

*## 17 25 min*

*## 18 1hr*

*## 19 20 mins*

*## 20 45 mins*

There is an obvious problem: these delays are 'human recorded', meaning that the format is inconsistent. Sometimes we have minutes, min, mts, hours, hrs; sometimes just two numbers separated by ':', and so on. Additionally, the times are not accurately recorded, as we will verify in a moment.

Wrangling the delay times into something machine readable takes some doing. I don't doubt that there is a more general path available, but I went through sequentially, case by case, as generally as I could; then cleaned up the rest at the end.

#--sequential clean

ii\_times <- ii\_breakdowns %>%

mutate(units\_delayed = str\_extract(How\_Long\_Delayed, "[:alpha:]+"),

time\_delayed = as.numeric(str\_extract(How\_Long\_Delayed, "[:digit:]+")),

units\_delayed = ifelse(str\_detect(units\_delayed, "^[mM]"), "minutes", "other"),

units\_delayed = ifelse([is.na](http://is.na)(units\_delayed), "other", units\_delayed)

) %>%

#remove "in" case, exclude any flagged with "[hH}"

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "in") & !str\_detect(How\_Long\_Delayed, "[hH]"),"minutes", units\_delayed)

) %>%

#deal with "to" case with minutes by taking only the first number. There are apparently no 'to' strings with hours in them.

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "to") & str\_detect(How\_Long\_Delayed, "[mM]"),"minutes", units\_delayed)

) %>%

#deal with the - cases which don't involve hours at all

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "-") & !str\_detect(How\_Long\_Delayed, "[hH]"),"minutes", units\_delayed)

) %>%

#deal with the double digit minute-1hr:

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "-") & str\_detect(How\_Long\_Delayed, "^[1-5][0-9]"),"minutes", units\_delayed)

) %>%

#everything left with a hyphen is a 1 hour or more delay

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "-"), 60, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "-"), "minutes", units\_delayed)

) %>%

#-all hyphens dealt with at this point

#Now work on the "/": first take the minute timers: all those with double digits first

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "/") & str\_detect(How\_Long\_Delayed, "^[1-9][0-9]"), "minutes", units\_delayed)

) %>%

#All those starting with 1/2 go to 30 minutes

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1/2"), 30, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1/2"), "minutes", units\_delayed)

) %>%

#All those starting with 1 1/2 go to 90 minutes

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1 1/2"), 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1 1/2"), "minutes", units\_delayed)

) %>%

#---specific cases

#"1 HOUR1/2"

mutate(

time\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "1 HOUR1/2", 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "1 HOUR1/2", "minutes", units\_delayed)

) %>%

#"@least 1/2"

mutate(

time\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "@least 1/2", 30, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "@least 1/2", "minutes", units\_delayed)

) %>%

#"1HR/20MIN"

mutate(

time\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "1HR/20MIN", 80, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & How\_Long\_Delayed == "1HR/20MIN", "minutes", units\_delayed)

)%>%

#--end specific cases

#1hr/x case

#This is modifying things it shouldn't. Why is this even here?

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "1hr"), 60, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "1hr"), "minutes", units\_delayed)

)%>%

#---cases with ":"

#remove cases which erroneously report a time instead of a duration, since these are not reliable

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, ":") & str\_detect(How\_Long\_Delayed, "[aA][mM]"), NA, time\_delayed)

)%>%

#2:hr cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^2:[hH][rR]"), 120, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^2:[hH][rR]"), "minutes", units\_delayed)

)%>%

#1:45 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:45"), 105, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:45"), "minutes", units\_delayed)

)%>%

#1:30 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:30"), 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:30"), "minutes", units\_delayed)

)%>%

#1: cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:"), 60, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1:"), "minutes", units\_delayed)

)%>%

#--end cases with ":"

#--cases with "^2"

#2 hr cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^2") & str\_detect(How\_Long\_Delayed, "[hH]"), 120, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^2") & str\_detect(How\_Long\_Delayed, "[hH]"), "minutes", units\_delayed)

)%>%

#double digit minute cases:

mutate(

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^2[0-9]"), "minutes", units\_delayed) #times are already correct

)%>%

#---"^1"

#1 hour cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1.\*[rR]$") & !(str\_detect(How\_Long\_Delayed, "^1.5.\*[rR]$")), 60, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1.\*[rR]$") & !(str\_detect(How\_Long\_Delayed, "^1.5.\*[rR]$")), "minutes", units\_delayed)

)%>%

#1.5 hour cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1.5"), 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1.5"), "minutes", units\_delayed)

)%>%

#NR: some of the ^1 x min cases going in here are already tagged as minutes.

#1 hr 30 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "30"), 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "30"), "minutes", units\_delayed)

)%>%

#1 hr 15 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "15"), 75, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "15"), "minutes", units\_delayed)

)%>%

#1 hr 20 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "20"), 80, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "20"), "minutes", units\_delayed)

)%>%

#1 hr 45 cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "45"), 105, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "45"), "minutes", units\_delayed)

)%>%

#1 hr half

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "half"), 90, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "half"), "minutes", units\_delayed)

)%>%

#1 h cases

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "[hH]"), 60, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^1") & str\_detect(How\_Long\_Delayed, "[hH]"), "minutes", units\_delayed)

)%>%

#---has pattern "^[1-9][oO]", that is, o replaces 0

mutate(

time\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^[1-9][oO]"), time\_delayed\*10, time\_delayed),

units\_delayed = ifelse(units\_delayed=="other" & str\_detect(How\_Long\_Delayed, "^[1-9][oO]") , "minutes", units\_delayed)

) %>%

#---final outliers and rejects

#just switch to minutes

mutate(

units\_delayed = ifelse(How\_Long\_Delayed == "35 SM", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "30am", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "30INS", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "45 late", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "40n", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "45 am", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "45NIN", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "45 INUTES", "minutes", units\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "45IN", "minutes", units\_delayed)

) %>%

#needs time modifications

mutate(

time\_delayed = ifelse(How\_Long\_Delayed == "!0 mins", 1, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "0-15 Min", 5, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "0-15 Min", 5, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "MAYBE 1/2", 30, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "half hour", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "Half hour", 30, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "Half hour", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "one hour", 60, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "one hour", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "ONE HOUR", 60, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "ONE HOUR", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "HOUR", 60, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "HOUR", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "hour", 60, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "hour", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "HR1", 60, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "HR1", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1/ 15 min", 75, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1:20MIN", 80, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1 20 min", 80, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1:30 min", 90, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1:30?mins", 90, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1 30mnts", 90, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1 30 min", 90, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "1:40 MINS", 90, time\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "IHR40MIN", 100, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "IHR40MIN", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "3 HRS", 180, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "3 HRS", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "3 hr", 180, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "3 hr", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "3 HOUR", 180, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "3 HOUR", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "4 hours", 240, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "4 hours", "minutes", units\_delayed),

time\_delayed = ifelse(How\_Long\_Delayed == "4 hrs", 240, time\_delayed),

units\_delayed = ifelse(How\_Long\_Delayed == "4 hrs", "minutes", units\_delayed)

) %>%

#sometimes the record just reads 'mins' or equivalent. My 'units\_delayed' picks this up as 'minutes'. We take a guess and set this to 5 minutes.

mutate(

time\_delayed = ifelse(units\_delayed == "minutes" & str\_detect(How\_Long\_Delayed, "[mM][iI][nN]") & [is.na](http://is.na)(time\_delayed), 5, time\_delayed)

) %>%

filter(!([is.na](http://is.na" \t "_blank)(time\_delayed))) %>%

filter(time\_delayed <= 300) %>% #Any delays beyond 5 hours are probably typos, but also problematic if real.

filter(units\_delayed == "minutes") %>% #222 820 / 259 637. The rest are garbage or NA.

select(

Busbreakdown\_ID, School\_Year, Route\_Number, Schools\_Serviced, Bus\_Company\_Name,

Has\_Contractor\_Notified\_Schools, Has\_Contractor\_Notified\_Parents, Have\_You\_Alerted\_OPT, Reason,

Boro, Number\_Of\_Students\_On\_The\_Bus, School\_Age\_or\_PreK,

Created\_On, Occurred\_On,

time\_delayed

)

#---plot of the time\_delayed

library(RColorBrewer)

pal <- brewer.pal(12, "Paired")

reason\_levels <- ii\_times %>%

group\_by(Reason) %>%

mutate(

total\_of\_reason = n()

) %>% distinct(Reason, total\_of\_reason) %>%

arrange(total\_of\_reason) %>%

ungroup() %>%

mutate(total\_of\_reason = row\_number())

ii\_times %>%

mutate(Reason = factor(Reason, levels = reason\_levels[[1]])) %>%

mutate(time\_delayed = time\_delayed) %>%

ggplot(aes(x = time\_delayed, fill = Reason)) +

geom\_bar() +

theme\_classic() +

scale\_fill\_manual(values = pal) +

coord\_cartesian(expand = c(0,0), xlim = c(0, 100)) +

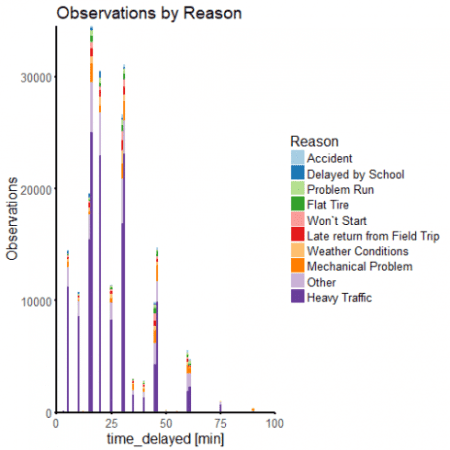
labs(x="time\_delayed [min]",y="Observations") +

theme(text=element\_text(family="Arial", size=16)) +

theme(axis.line = element\_line(colour = 'black', size = 1)) +

theme(axis.ticks = element\_line(colour = "black", size = 1)) +

ggtitle("Observations by Reason")

[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-2-1.png?ssl=1)

A bar graph broken up Reason shows that, as observed when cleaning up the delay data, these are 'human approximated delays'. The driver or attendant (or perhaps someone at operations) has approximated the delay to the nearest convenient time package: 5 minutes, 10 minutes, 15 minutes, 20, 25, 30, 40, 45, 50, 60, 90 minutes, and so on. Some delays have been figured out more accurately by comparing time differences, but most have not. One of the consequences is that if we decide to run a regression model for this data, we can never expect it to be more precise than 5-10 minutes of RMS error. Because that is how the data have been recorded.

Note that there are a few data with delays beyond 100 minutes, and these points will show up later. The graph is presented this way for clarity. Traffic constitutes the major source of delay in the data set; accident is the most rare problem. 'Other', meaning a missing reason or undisclosed reason, is the second most common value in this field.

We can have a look at some of the structure via density plots:

ii\_times %>%

mutate(Reason = factor(Reason, levels = reason\_levels[[1]])) %>%

mutate(time\_delayed = time\_delayed) %>%

ggplot(aes(x = time\_delayed, fill = Reason)) +

geom\_density(aes(y=..count..), size = 0, adjust = 4, position = "stack", color = NA) +

theme\_classic() +

scale\_fill\_manual(values = pal) +

coord\_cartesian(expand = c(0,0), xlim = c(0, 100)) +

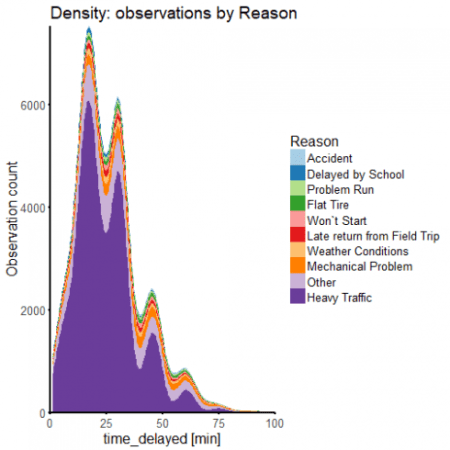
labs(x="time\_delayed [min]",y="Observation count") +

theme(text=element\_text(family="Arial", size=16)) +

theme(axis.line = element\_line(colour = 'black', size = 1)) +

theme(axis.ticks = element\_line(colour = "black", size = 1)) +

ggtitle("Density: observations by Reason")

[](https://i0.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-3-1-12.png?ssl=1)

boro\_levels <- ii\_times %>%

mutate(

Boro = ifelse([is.na](http://is.na)(Boro), "All Boroughs", Boro),

) %>%

group\_by(Boro) %>%

mutate(

total\_of\_boro = n()

) %>% distinct(Boro, total\_of\_boro) %>%

arrange(total\_of\_boro) %>%

ungroup() %>%

mutate(total\_of\_boro = row\_number())

ii\_times %>%

mutate(

Boro = ifelse([is.na](http://is.na)(Boro), "All Boroughs", Boro),

) %>%

mutate(Boro = factor(Boro, levels = boro\_levels[[1]])) %>%

ggplot(aes(x = time\_delayed, fill = Boro)) +

geom\_density(aes(y=..count..), size = 0, adjust = 4, position = "stack", color = NA) +

theme\_classic() +

scale\_fill\_manual(values = pal) +

coord\_cartesian(expand = c(0,0), xlim = c(0, 100)) +

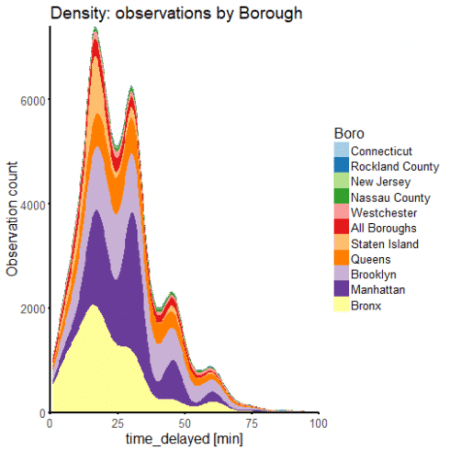
labs(x="time\_delayed [min]",y="Observation count") +

theme(text=element\_text(family="Arial", size=16)) +

theme(axis.line = element\_line(colour = 'black', size = 1)) +

theme(axis.ticks = element\_line(colour = "black", size = 1)) +

ggtitle("Density: observations by Borough")

[](https://i1.wp.com/datascienceplus.com/wp-content/uploads/2018/11/unnamed-chunk-3-2.png?ssl=1)

Here we must be careful with interpretation, since the density plot presents counts convolved with a gaussian density kernel. This helps us to see structure, but we must be careful interpreting the counts. There are still very few observations with delay time of 22 minutes, for example.

But we can see some structure a little more clearly. For example, Manhattan produces more long-time delays than the Bronx or Staten Island.

In the next set of notes we will try to fit a regression, based only on this data. However, we can already see the need to join more data on as we have few predictors. Before building a model or joining to the other data sets, there is a little more cleaning and dummifying of categorical variables to do:

ii\_spread <- ii\_times %>%

mutate(

time\_diff\_report = abs(as.numeric(difftime(Created\_On, Occurred\_On, units = "mins")))

) %>%

mutate(

reported\_before\_resolved = (time\_delayed > time\_diff\_report)\*1L,

Created\_On = NULL,

Has\_Contractor\_Notified\_Schools = (Has\_Contractor\_Notified\_Schools == "Yes")\*1L,

Has\_Contractor\_Notified\_Parents = (Has\_Contractor\_Notified\_Parents == "Yes")\*1L,

Have\_You\_Alerted\_OPT = (Have\_You\_Alerted\_OPT == "Yes")\*1L,

School\_Age = (School\_Age\_or\_PreK == "School-Age")\*1L,

School\_Age\_or\_PreK = NULL

) %>%

mutate(dummy = 1L,

Reason = str\_replace\_all(Reason, " ", ""),

Reason = str\_replace\_all(Reason, "\`", "")

) %>%

spread(key = Reason, value = dummy, fill = 0L, sep = "\_") %>%

mutate(Reason\_Other = NULL) %>% #default case is Other

mutate(dummy = 1L,

Boro = ifelse([is.na](http://is.na)(Boro), "All Boroughs",Boro),

Boro = str\_replace\_all(Boro, " ", "")

) %>%

spread(key = Boro, value = dummy, fill = 0L, sep = "\_") %>%

mutate(Boro\_AllBoroughs = NULL) %>% #default case is all

mutate(

Number\_Of\_Students\_On\_The\_Bus = as.numeric(Number\_Of\_Students\_On\_The\_Bus) #This field has some problems we will deal with later

) %>%

filter(Number\_Of\_Students\_On\_The\_Bus <= 80) #Although this cut is arbitrary, we choose not to believe the records with more than 80 people on the bus.