# Exploring Single Discrete Variable Plots

ACTIVITY 26

We will use the incidents_217_2009 data set to introduce the topics discussed in Keen's Chapter 2 on plotting techniques for a single discrete variable. You should think broadly when creating these plots: what do these discrete variables tell us?

## Purpose

This activity helps you to explore R's ability to review large data sets quickly and make assessments about data types.

## Learning Objective

- Recognize the difference between discrete and continuous variables in plots and data types.
- Recognize that counting records are equivalent to counting events (because many databases are event-level)

## Required Resources

- R, R Studio, PostgreSQL ODBC driver, Direct connection to PostgreSQL with R
- Brief overview of discrete and continuous variables

## Time Allocated

65 minutes in class

## Tasks

### A. Orienting Yourself to the Incident Data Set

Let's browse one of the class databases available for exploration: the *incidents* data set. The instructor will give a brief overview of how the *incidents* data set. Browse the metadata for incidents tables. Use phpPgadmin to browse through the *incidents.incidents_217_2009* table. For all of the *incidents* questions, exclude all of the Boolean error checking variables.

### B. Connect to the Class Database

Connect to the database and read the *incidents* data into R. The SQL query is (don't forget the R syntax to connect and query).

```
library(RODBC)

channel<-odbcConnect(“DATASOURCE”,”ce510”)

qry <- “SELECT * FROM incidents.incidents_217_2009 ORDER BY incidentid”

incidents <-sqlQuery(channel, qry)
```

Let's select one column and issue a simple plot command for each the variables in question 1. R is pretty smart about plotting. Let's just pick the impact type column (*impacttypeid*), which certainly appears discrete.

```
plot (incidents$incidenttypeid)
```

In R, we can use a simple plot to see all of the data plotted

```
#--------------------------------------------------------

par(mfrow=c(4,4), mar=c(2,2,3,1))
```

```
for (i in 1:58) {
plot (incidents[,i], main=names(incidents[i]))
}
```

Note the use of a helpful function to extract the names of a column to use on the plot is:

```
names(incidents)[i]
```

Remember: when *i* is replaced with an integer, the call returns the index or column name corresponding to the number. The first column of the incidents dataframe is *incidentid*. The first plot frames look like:

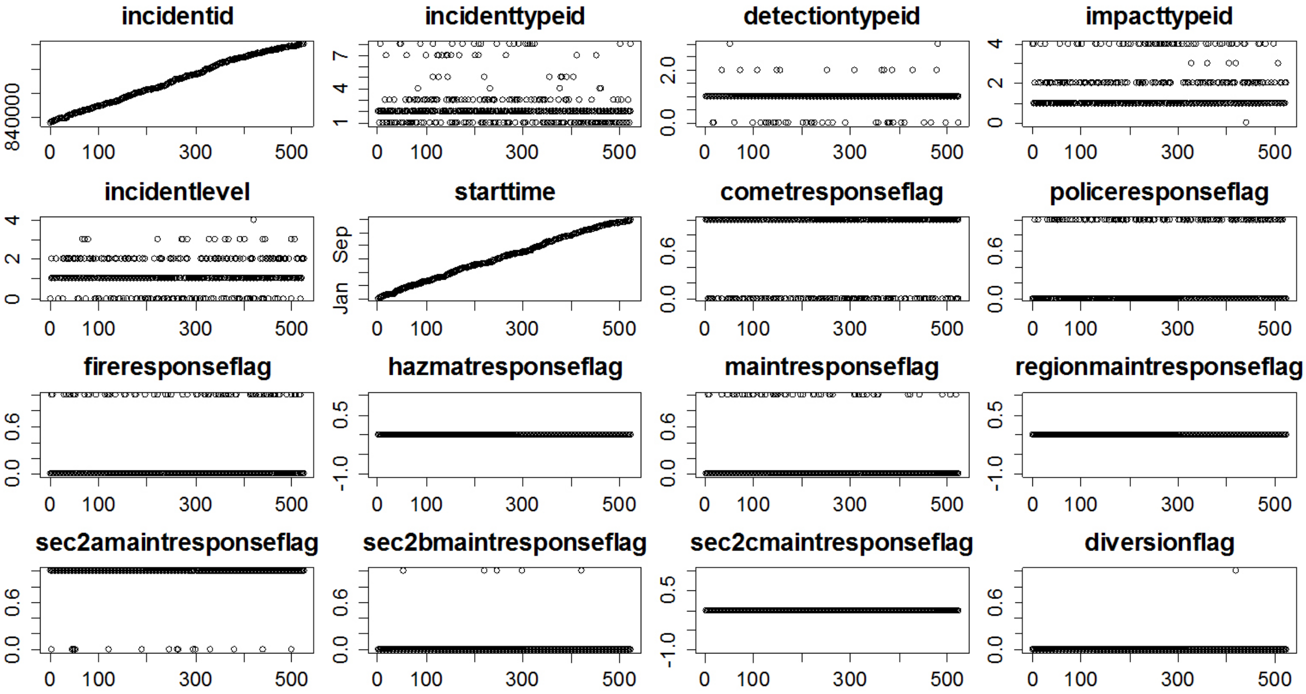

**Figure 47**

We will use the column *incidentypeid* in the rest of the activity. Incident type has the following values:

- 0 Unknown;
- 1 Accident;
- 2 Stall;
- 3 Debris;
- 4 Tow;
- 5 Construction;
- 6 Congestion;
- 7 Other Closure;
- 8 Other Incident;
- 9 Tag

You should select from the plots 4 variables which look interesting and are discrete variables to make your plots. Do not use any of the flag variables. Figure 48 shows a jittered stripplot of the incident type

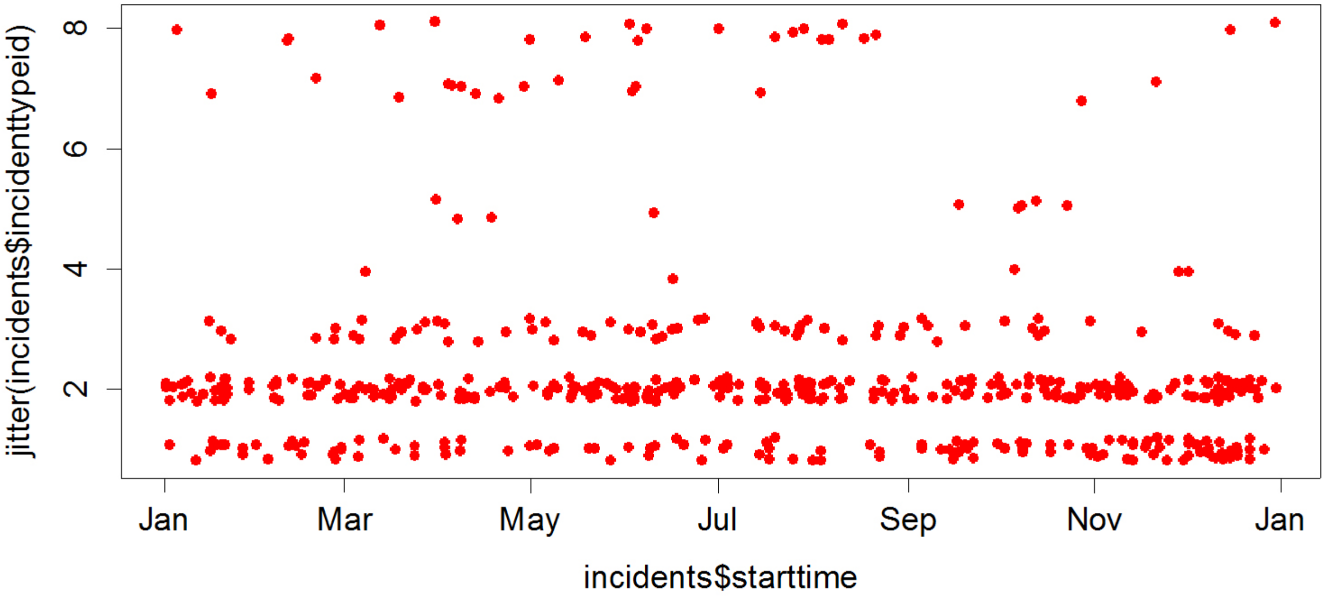

**Figure 48**

## C. Counting Row Events

Before we move to the types of discrete plots, let's start by having R count the number of rows that are in the data frame *incidents* for each discrete element of `incidents[,4]`

```
table (incidents$incidenttypeid)
```

which returns

```
   1    2    3    4    5    7    8
126  283   65    5    9   15   21
```

What do these counts represent? Think about this, since so many of the data we will encounter are event-level data where each row is "something." Counting how many rows have each element of a discrete variable is akin to counting the number of "somethings."

It is also easy to do a 2-way table. Let's extract, extracting month from startime with the lubridate package

```
table ( incidents$incidenttypeid, month (incidents$starttime))
```

#Calculating the col percentages using margins. Change the 2 to 1 to calculate row %

#Columns sum to 1

```
prop.table(           table(incidents$incidenttypeid,           month
(incidents$starttime)),2)
```

#Rows sum to 1

```
prop.table(           table(incidents$incidenttypeid,           month
(incidents$starttime)),1)
```

### D. Type of Plots

Run the R scripts to see how to create

- Stripchart
- Bar Chart
- Pie Chart
- Dot Plot

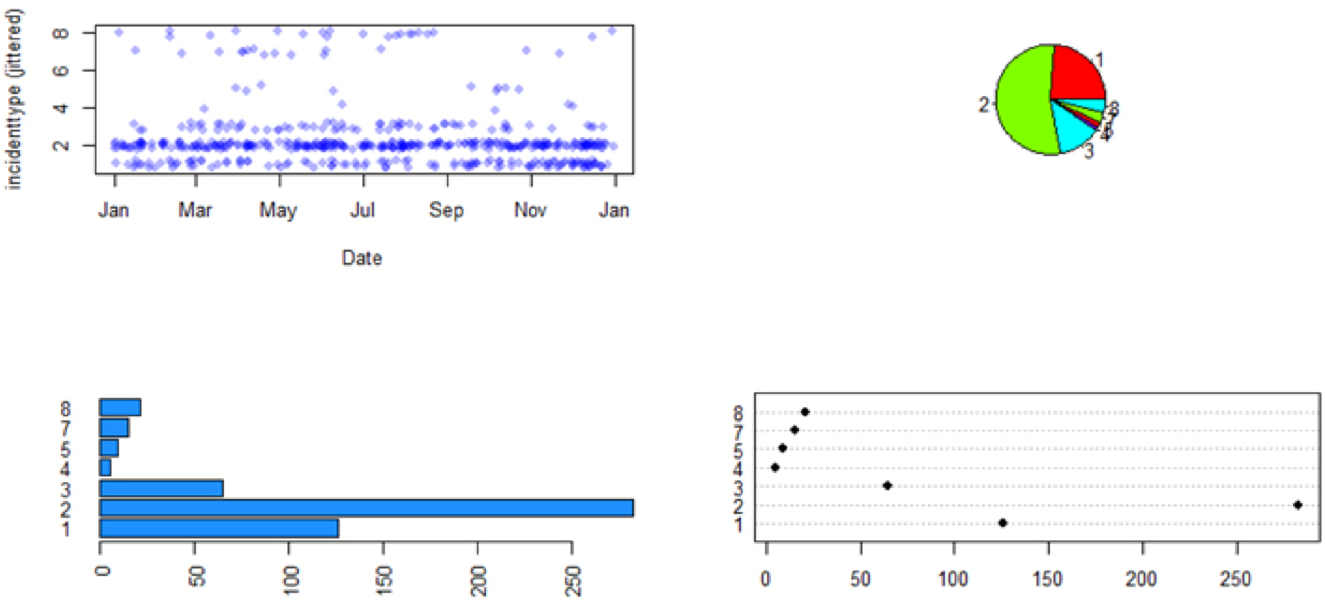

**Figure 49** Sample Plots of Incident Type

### E. On Your Own

Using the existing code, make plots of the four variables you identified. Try making a two-way table by month or day of the week to see possible trends in the data.

## Deliverable

Submit a pdf document with a short write up to the course management dropbox.

## Assessment

This activity only requires you to complete and discuss the plots, you will not be assessed on the graphical presentation quality (though you are free to make them as "pretty" as you like).

| All plots complete | 1 – 2 mlssing | 3 – 4 mlssing | > than 4 missing |
|---|---|---|---|
| 10 points | 9 points | 8 points | 5 points |