Tutorial: Big data with R

Deadline: None.

1. You need to install packages *ff* and *ffbase*. Also download the data for 2007 and 2008 on your computer from `http://stat-computing.org/dataexpo/2009/the-data.html`.

   (a) Import the data files for *2007* and *2008* as ff data frames. Make sure *Month*, *DayOfWeek* and *Year* are imported as factors.

   (b) Save both data frames on your hard drive.

   (c) Take a subset of the 2008 data for December; drop unused levels of *Month*. This is just practice, we are not going to use the subset.

2. In this exercise we are going to create the data we are going to use to plot all airports on a map for USA with predicted delays for 2007 and 2008.

   (a) Run two separate regressions for each one of the two data sets where *ArrDelay* is dependent variable (Y) and *Origin* and *Distance* are explanatory variables.

      i. Obtain predicted ArrDelays for each Airport in Origin (for each year separately) while holding *Distance* fixed at the sample average (for each year).

      In the package *biglm* there is a predict function, you could try this if you are running the regression with biglm(). If you are running the regression with the Chunk_lm.R function, then a predict function for this purpose is given here: **Predicted_airports.R**.

      ii. Put the predicted values, for both years, into one data.frame with the airport abbreviations in the first column, predicted

values in the second and a year variable in the third column. Call the variables "Origin", "pred" and "year".

(b) To plot the airports we need coordinates for all airports in the states. Download and import the *airports.csv* data into R.

(c) For plotting we will use the coordinates in *airports.csv*. Now use the function merge() to merge the airports data frame and the data frame with predicted values. Variables *iata* and *Origin* can be used as merging variables (by.x and by.y). Call the data.frame "plot_data".

3. Now we are going to do some plotting based on the "plot_data" data frame constructed in 2c.

(a) Install and load the package **ggplot2**.

(b) Use the following code to plot USA:

```
#coordinates US boarders (included in the ggplot2 package)
map.us <- map_data(map = "state")
p1 <- ggplot()
p1 <- p1 + geom_polygon(data=map.us,
aes(x = long, y = lat,group=group),fill = grey(0.5))
p1
```

(c) Now plot the airports with the following code and the data.frame plot_data:

```
p1<-p1+geom_point (data = plot_data,
aes (x = long, y = lat),pch = 16)
p1
#Split in to two plots based on year
p1<-p1 + facet_grid(. ~ year)
p1
```

(d) There are some points outside mainland USA, make a subset of "plot_data" where the coordinates only include mainland USA. Use the code above on this subset. Run the code with the subset data from the beginning again.

(e) One can add colour to the points conditioning on predicted delay like this

```
p1<-p1+geom_point (
data = plot_sub,
```

```
aes (
x = long,
y = lat,
colour =pred #ifelse(pred>0,"Delay","Ahead")
),pch = 16)+
 theme(legend.position=c(.5, .175))+labs(colour="Color")+
scale_colour_gradient(low = "#56B1F7", high = "#132B43")
p1
```

(f) One can add text of predicted delays like this:

```
#Add text to points with predicted delays
p1<-p1+geom_text(data = plot_sub,aes (
x = long,
y = lat,label=round(pred,0)
),size=3.2,vjust=0.6)
p1
```

and for the top 1 % like this

```
plot_sub2<-subset(plot_sub,pred>=quantile(pred, probs = 0.99))
p1<-p1+geom_text(data = plot_sub2,aes (
x = long,
y = lat,label=round(pred,0)
),size=3.2,vjust=0.6)
p1
```

but then you have to run from the beginning again and without plotting the predictions for each airport.

(g) One can also add the airport abbreviations in similar way for the top 1 %. Try to do this. Place the abbreviation above the predicted values.

(h) We have now picked out the airports with longest predicted delays. However, these airports might not be the important ones. With the code below I create a size variable including the number of flights for each airport and year.

```
#Table number of flights from each airport in the data
numb07<-table(sub_07$Origin[])
numb08<-table(sub_08$Origin[])
#Make a data.frame with the information from table()
flights<-data.frame(c(dimnames(numb07)[[1]],
```

```
dimnames(numb08)[[1]]),
rbind(cbind(as.numeric(numb07),2007),
cbind(as.numeric(numb08),2008)))
names(flights)<-c("Origin","numb","year")
#Merge with plot_sub
plot_sub2<-merge(plot_sub,flights,by.x=c("iata","year"),
by.y=c("Origin","year"))
```

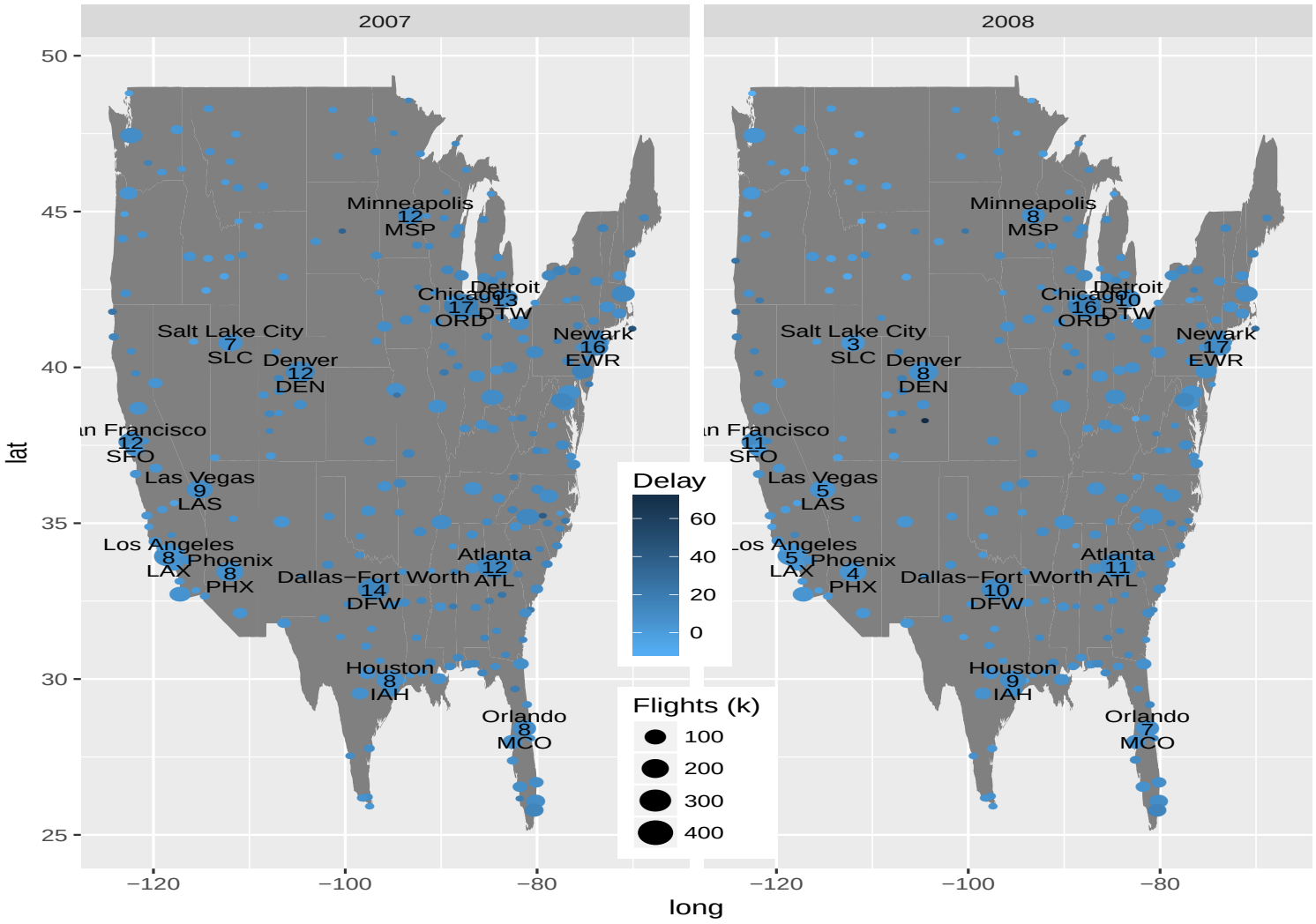
(i) I now use the information of number of flights (in plot_sub2) to set the size on each point.

```
rm(p1)
p1 <- ggplot()
p1 <- p1 + geom_polygon(data=map.us,
aes(x = long, y = lat,group=group),fill = grey(0.5))
p1<-p1+geom_point (
data = plot_sub2,
aes (x = long, y = lat,colour = pred,size =numb/1000),
pch = 16)+labs(size = "Flights (k)",colour="Delay")+
theme(legend.position=c(0.5, .25))+
scale_colour_gradient(low = "#56B1F7", high = "#132B43")
p1<-p1 + facet_grid(. ~ year)
p1
```

4. The airports with the longest delays are very small airports. Try now to plot the delays of the 5% largest airports (the delays printed at the correct points).

   If you want to save the plot you have made, you can do it like this

   ```
   ggsave("~/map.pdf", plot =p1)
   ```


2007
2008
lat
long
50
45
40
35
30
25
−120
−100
−80
Minneapolis
12
MSP
Detroit
13
DTW
Chicago
17
ORD
Salt Lake City
7
SLC
Denver
12
DEN
Newark
16
EWR
San Francisco
12
SFO
Las Vegas
9
LAS
Los Angeles
8
LAX
Phoenix
8
PHX
Dallas−Fort Worth
14
DFW
Atlanta
12
ATL
Houston
8
IAH
Orlando
8
MCO
Delay
60
40
20
0
Flights (k)
100
200
300
400

ffx08