

Un Caso Práctico Usando Leftraru y Visualizando los Resultados con R

Andrew Hart

El Centro de Modelamiento Matemático,
Universidad de Chile

8 de Septiembre de 2016

1 Estudio de Caso

2 Introducción a R

3 Gráficos en R

El problema

Modelar el flujo de tráfico en algunas calles de Santiago Centro y averiguar en que manera cambios en los programas de los semáforos influyen en el flujo de tráfico.

- Mapa de Ave. Matta entre Vicuña Mackenna y Blanco Encalada.
- Investigar el efecto que el largo de las fases rojos y verdes tiene en el tiempo de recorrer por Matta.

- SUMO: Simulator of Urban MObility.
- Patrocinio: Instituto de Sistemas de Transporte (DLR), Alemania.

<http://sumo.dlr.de>.

- Inputs: Archivos xml que contiene el mapa y vehículos/rutas para simular.
- El mapa es de Matta y sus alrededores importado de www.openstreetmap.org.

Flujos de vehículos por rutas fijas

- por Matta: poniente-orienté
- por Matta: orienté-poniente
- 7 flujos de vehículos por calles que cruzan Matta.

Tráfico aleatorio de fondo

- Vehículos generados con orígenes y destinos aleatorios a lo largo de la simulación.

de Jobs

1728 -10 13286

172800 -10 21128

...

172800 -9 18202

...

172800 -5 22032

...

172800 10 26563

```
#!/bin/bash
#Set up a simulation
#Get simulation end time, tll program and
#random seed from job index
line=`sed -n "$1p" jobparams`
end=`echo -e "$line" | cut -f 1 -d " "`
tll=`echo -e "$line" | cut -f 2 -d " "`
seed=`echo -e "$line" | cut -f 3 -d " "`

#Make results dir
resultsdir=$HOME/simresults
[ ! -d $resultsdir ] && mkdir $resultsdir
```

```
#Create temporary working directory
tempdir=`mktemp -d`
#Copy necessary files
...
cp -t $tempdir t1l${t1l}.xml e2.xml
cp -t $tempdir *.py
cd $tempdir

#Set environment variables needed by sumo
export sumo_home=$HOME/src/sumo-0.27.1
export PYTHONPATH=$PYTHONPATH:$sumo_home/tools
export PATH=$PATH:$sumo_home/bin:$sumo_home
/tools:$sumo_home/tools/output:$sumo_home/tools
/xml:$sumo_home/tools/route
#Generate random background traffic
python randomTrips.py -n matta.net.xml -b 0 -e\
$end -p 4.5 --binomial=1 --fringe-factor 20\
--vehicle-class passenger -r background.rou.xml
```

```
#Generate Matta and crossing street traffic
python setFlowEnd.py -t temp-matta.flo.xml -e \
$Send -o matta.flo.xml
#Merge background route traffic and flows
python mergeXML.py matta.flo.xml \
background.rou.xml -o matta.rou
python sort_routes.py matta.rou
mv matta.rou.sorted matta.rou.xml
rm matta.rou

#Merge tll program and e2 detector defs
python mergeXML.py tll${tll}.xml e2.xml \
-o additional.xml
#Run the simulation
sumo -c matta.sumocfg -a additional.xml -e $Send \
--seed $seed
```

```
#Convert the tripinfo data from xml to csv and
#save it in the results directory
python xml2csv.py trips.out.xml -s , \
-o $resultsdir/tripdata$1_${t11}.csv

#Convert the e2 detector data from xml to csv
#and save it in the results directory
python xml2csv.py det.xml -s , \
-o $resultsdir/detdata$1_${t11}.csv

#Remove the temporary working directory
rm -rf $tempdir
```

- Creado por Ross Ihaka y Robert Gentleman por combinar dos lenguajes existentes:: S y Scheme.
- Es software open source.
- Es gratis.
- Incluye los dos características de lenguajes orientados objetos y lenguajes funcionales.
- Comparable con software estadístico comercial.
- Mucha gente contribuyen a R y hay muchas paquetes disponibles para extender su funcionalidad (más que 8000 en CRAN).
- Tiene amplias facilidades para explorar y visualizar datos y resultados de análisis.

tipos Básicos de Variables

Variables en R son objetos de los siguientes tipos y es posible crear nuevas clases de objetos.

[
Tipos de Variables]

- `character` información textual.
- `numeric` valores numéricos.
- `logical` valores de TRUE y FALSE.
- `factor` valores categóricos.
- `list` contenedor de variables de tipos diferentes.
- `data.frame` Similar a lists, pero juntan datos de tipos diferentes en una tabla.

Vectores, Matrices y Arreglos

Se puede tener vectores, matrices y arreglos de los tipos.

R tiene funciones para importar datos de varios formatos de archivos que incluyen:

- tsv: tab-separated values
- csv: comma-separated values
- netcdf y hdf
- xls y xlsx: libros de trabajo de Microsoft Excel
- Archivos de otros softwares de estadísticos.

```
#Lee los datos de Matta poniente-oriente
mattaWE <- read.csv("matta-we.csv")

#Extrae información de interés y ponerla en
#un data frame
we <- data.frame(
  id=as.character(mattaWE$stripinfo_id),
  duration=as.numeric(mattaWE$stripinfo_duration),
  loss=as.numeric(mattaWE$stripinfo_timeLoss),
  length=as.numeric(mattaWE$stripinfo_routeLength),
  depart=as.numeric(mattaWE$stripinfo_depart),
  arrival=as.numeric(mattaWE$stripinfo_arrival),
  adjust=as.numeric(mattaWE$adjust),
  job=factor(mattaWE$job),
  dep.int=floor(mattaWE$stripinfo_duration/900),
  stringsAsFactors=FALSE
)
```

Sistemas de Gráficos en R

R posee tres sistemas de gráficos que son incompatibles en general:

- Clásico
- Lattice
- ggplot2

Cada uno tiene sus ventajas y desventajas.

Scatter Plots

```
#Calcular promedios para los ajustes
m.we <- tapply(we$duration, we$adjust, mean)
s.we <- tapply(we$duration, we$adjust, sd)
df <- data.frame(Promedio=m.we, Desv=s.we)

#Generar un ploteo
plot(as.numeric(names(m.we)), m.we,
     xlab="ajuste", ylab="Promedio",
     main="Duración por Matta Poniente-Oriente")
```

Estadísticos de Duración de Tráfico Poniente-Oriente

ajuste	Promedio	Desv.
-10	376.24	29.01
-9	389.72	29.43
-8	401.27	32.41
-7	407.42	33.89
-6	407.23	36.78
-5	408.87	40.05
-4	405.51	39.87
-3	406.30	40.58
-2	403.66	39.94
-1	407.23	42.22
0	412.01	45.54

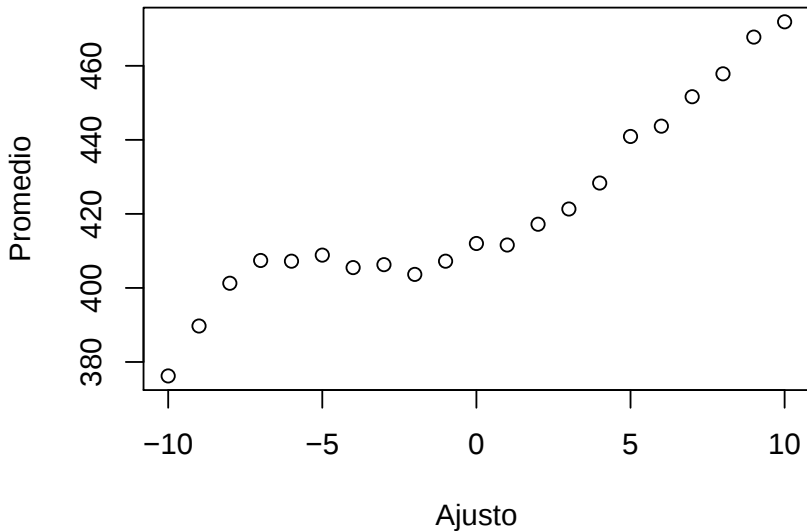
ajuste	Promedio	Desv.
1	411.60	38.87
2	417.21	38.83
3	421.32	35.48
4	428.34	39.01
5	440.93	47.59
6	443.71	42.70
7	451.66	45.12
8	457.84	43.09
9	467.77	48.10
10	471.90	43.32

Estadísticos de Duración de Tráfico Oriente-Poniente

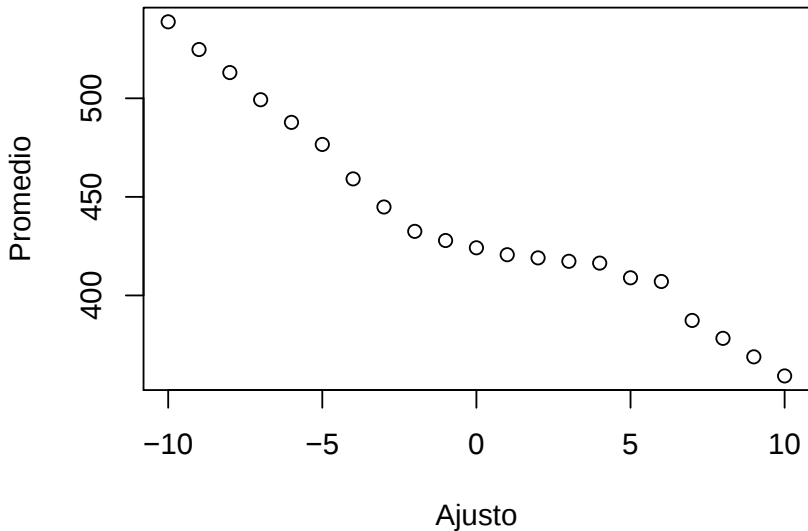
ajuste	Promedio	Desv.
-10	538.82	37.46
-9	524.75	39.34
-8	513.06	36.73
-7	499.29	39.91
-6	487.80	42.06
-5	476.61	44.11
-4	459.13	43.18
-3	444.88	41.12
-2	432.51	41.31
-1	427.88	31.03
0	424.19	30.84

ajuste	Promedio	Desv.
1	420.67	33.15
2	419.07	34.53
3	417.34	36.47
4	416.41	37.98
5	408.96	38.60
6	407.09	93.66
7	387.34	40.03
8	378.25	42.19
9	368.88	42.81
10	359.19	44.34

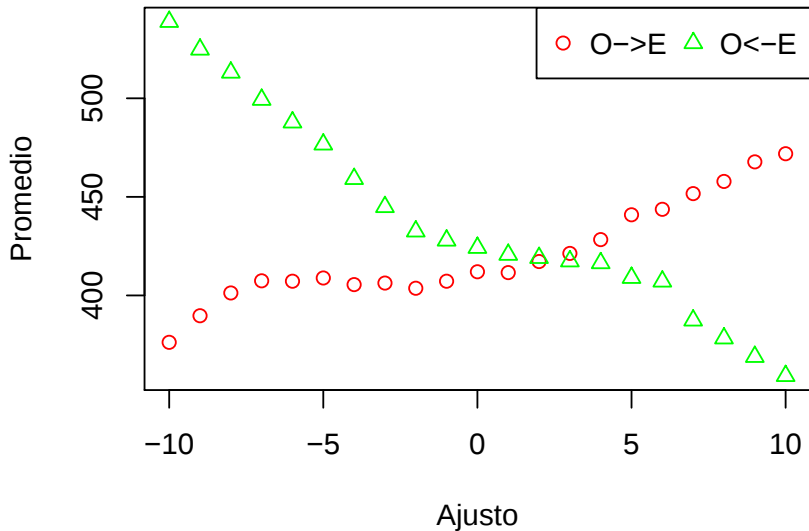
Duración por Matta Poniente–Oriente



Duración por Matta Oriente–Poniente



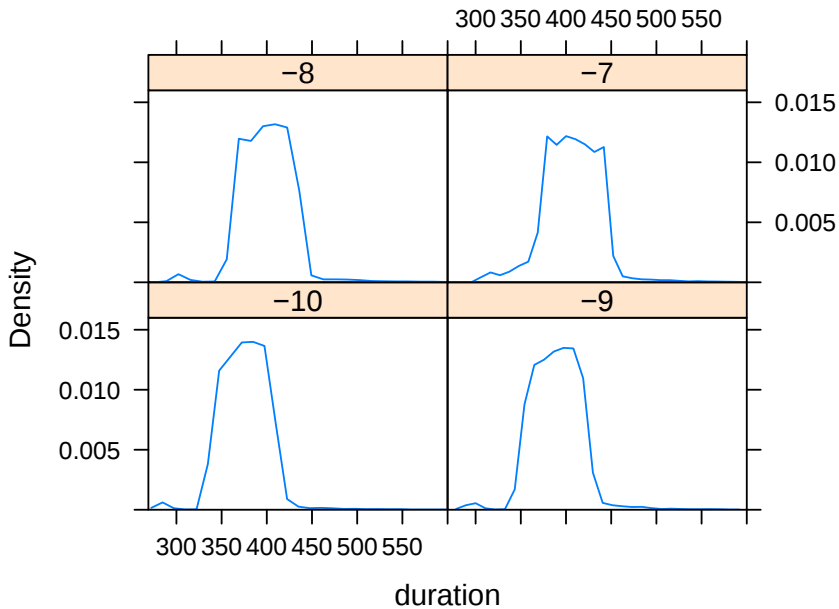
Duración por Matta



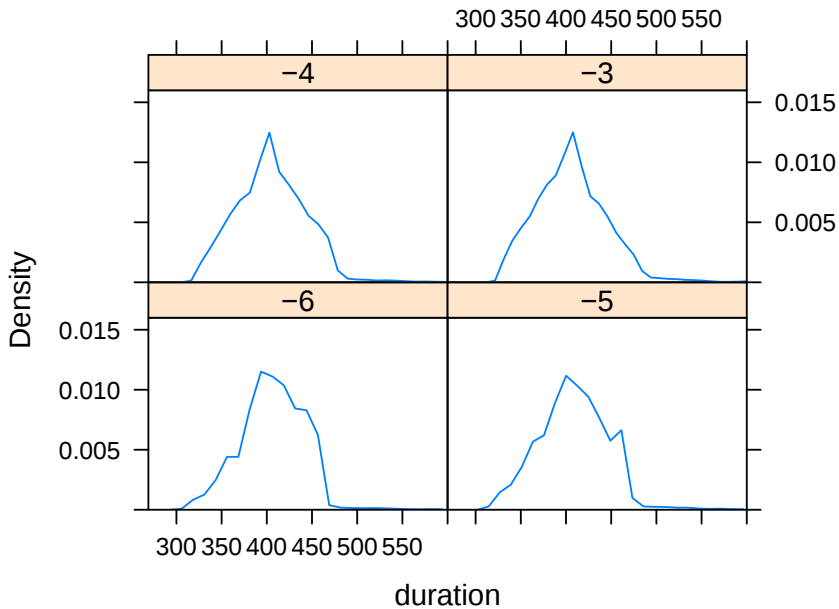
```
#Mande gráficos a un archivo pdf
pdf("jointscatter.pdf", width=5, height=4)
#Crear vectores x y y
x <- as.numeric(c(names(m.we), names(m.ew)))
y <- c(m.we, m.ew)
#Clasificar puntos por dirección de flujo
dir <- c(rep(1, length(m.we)), rep(2,
  length(m.ew)))
col <- c("red", "green") #colores

#Generar pplotteo
plot(x, y, pch=dir, col=col[dir],
xlab="ajuste", ylab="Promedio",
main="Duración por Matta")
#Agregar leyenda al ploteo
legend("topright", legend=c("O->E", "O<-E"),
pch=1:2, col=col, horiz=TRUE)
dev.off() #cierre archivo
```

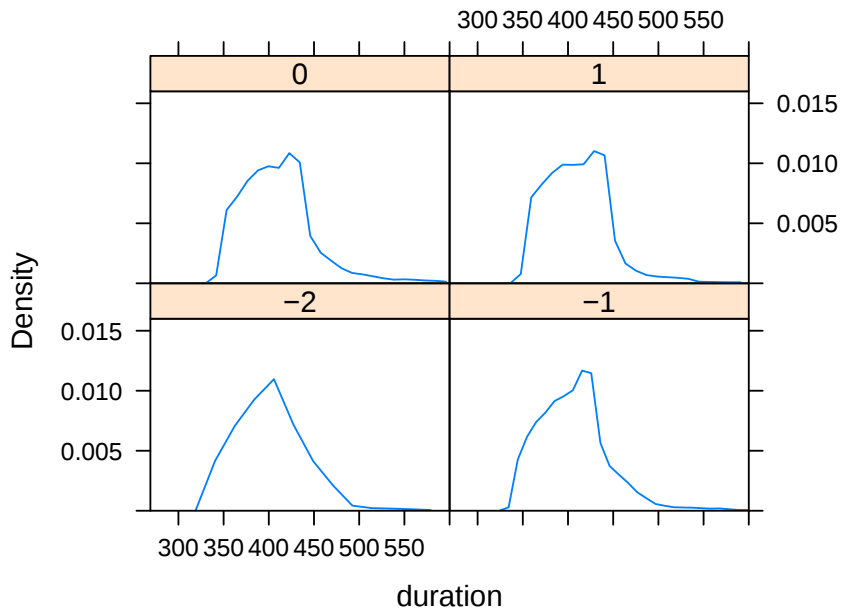
Densidades por Matta Poniente–Oriente



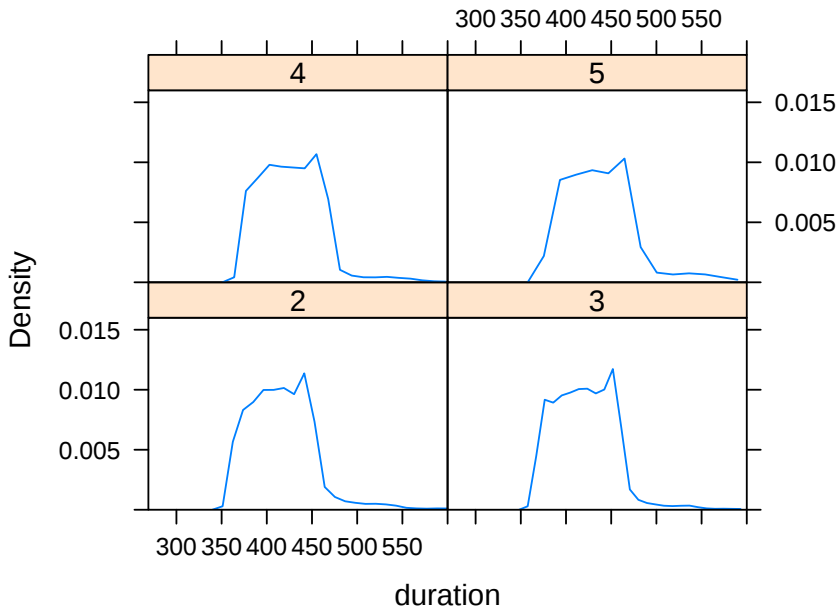
Densidades por Matta Poniente–Oriente



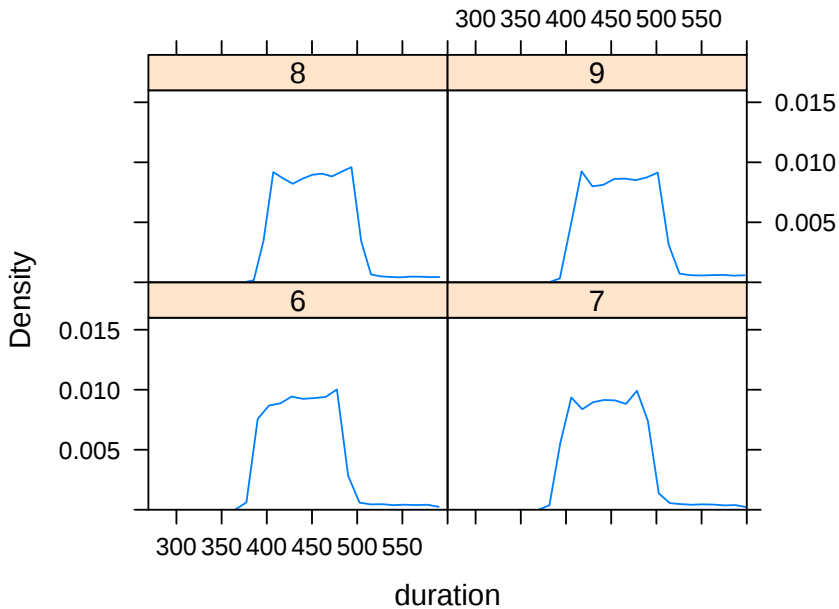
Densidades por Matta Poniente–Oriente



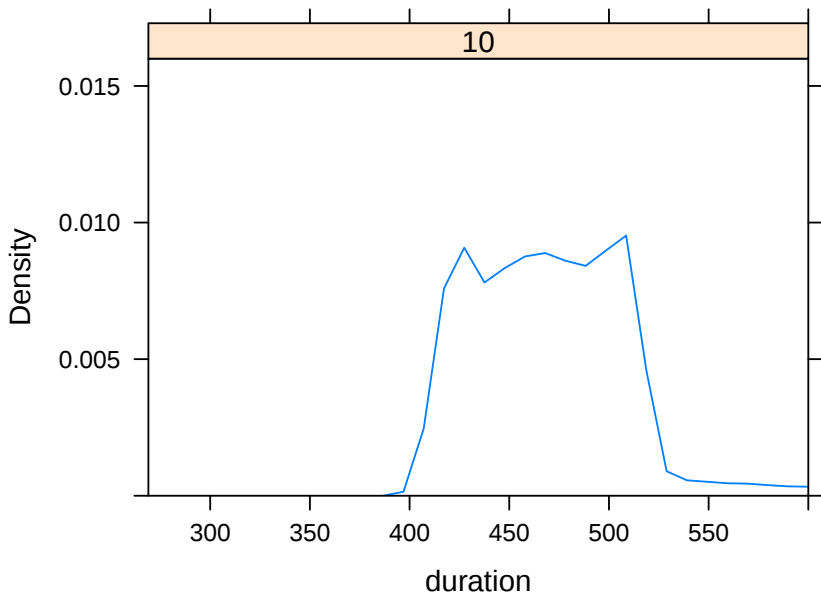
Densidades por Matta Poniente–Oriente



Densidades por Matta Poniente–Oriente



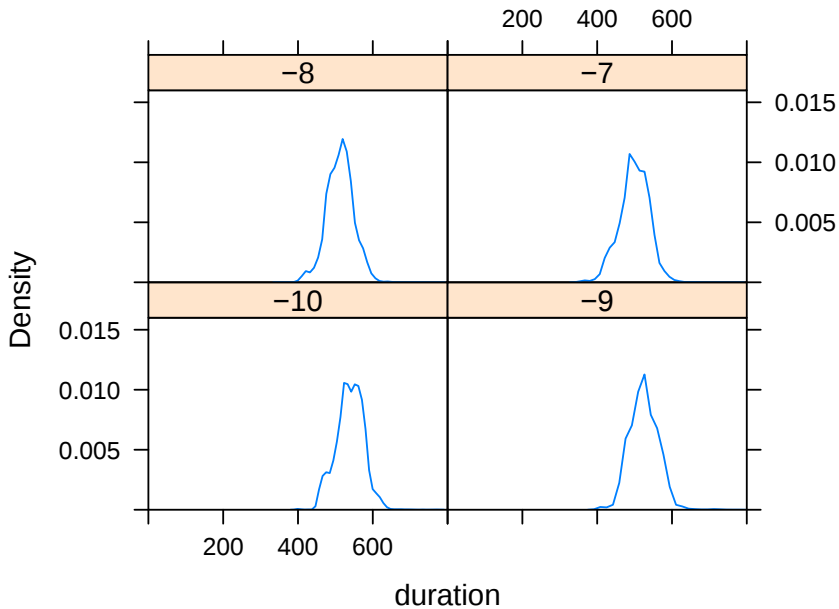
Densidades por Matta Poniente–Oriente



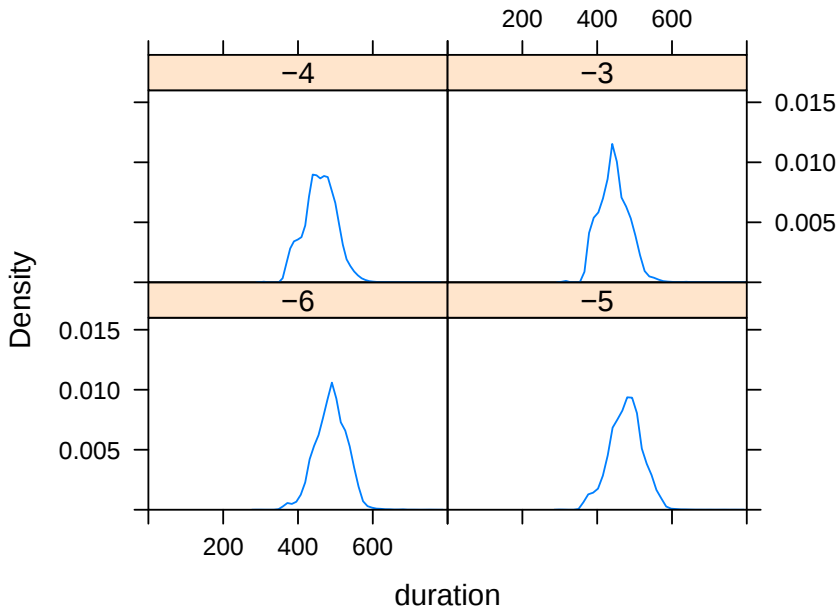
```
library(lattice)

#Densidades de duración poniente-oriente
xrange <- c(min(we$duration), 600)
yrange <- c(0, 0.016)
adjs <- levels(we$adjust)
for (i in 1:ceiling(length(adjs)/4)) {
  a <- ads[((i-1)*4+1):min(i*4, length(ads))]
  pdf(paste0("densWE", i, ".pdf"), width=5,
      height=4)
  dp <- densityplot(~ duration | adjust,
    data=we, subset=adjust %in% a, xlim=xrange,
    ylim=yrange, plot.points=FALSE,
    main="Densidades por Matta Poniente-Oriente")
  print(dp)
  dev.off()
}
```

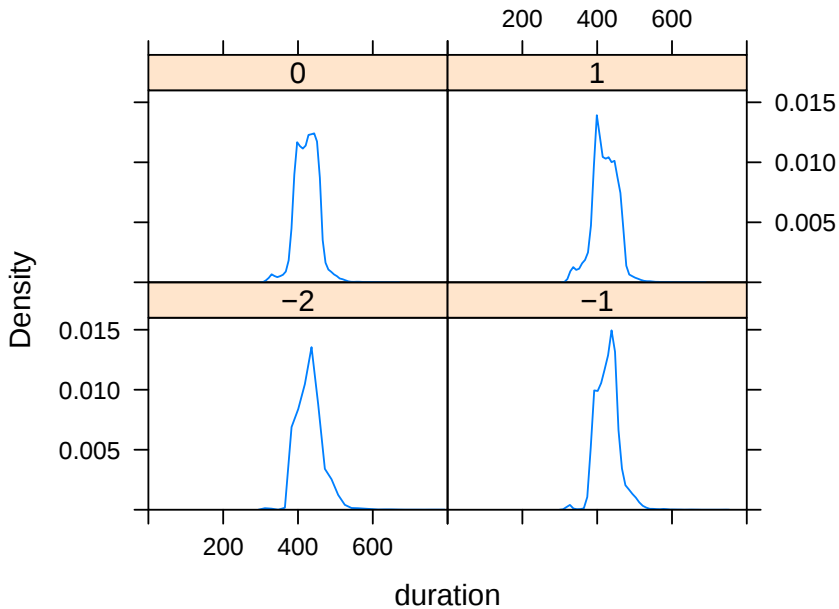
Densidades por Matta Oriente–Poniente



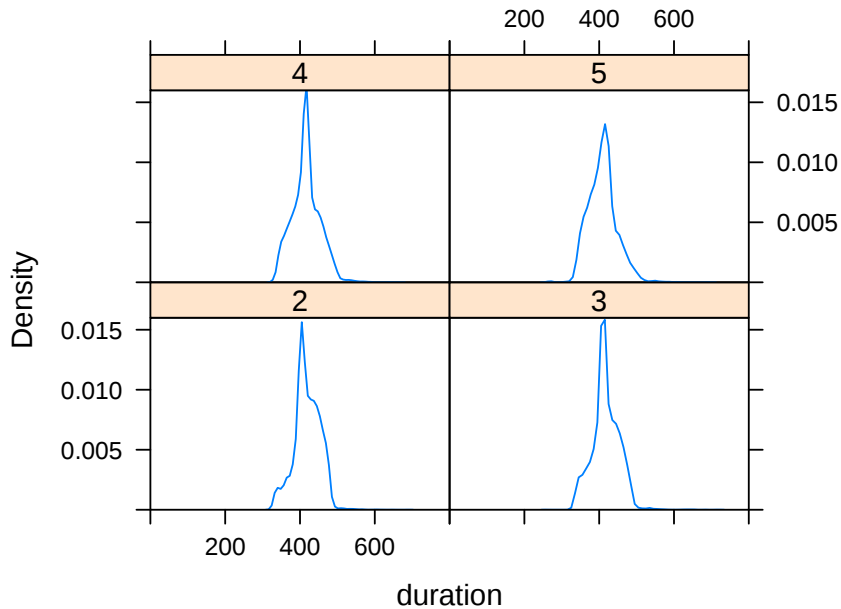
Densidades por Matta Oriente–Poniente



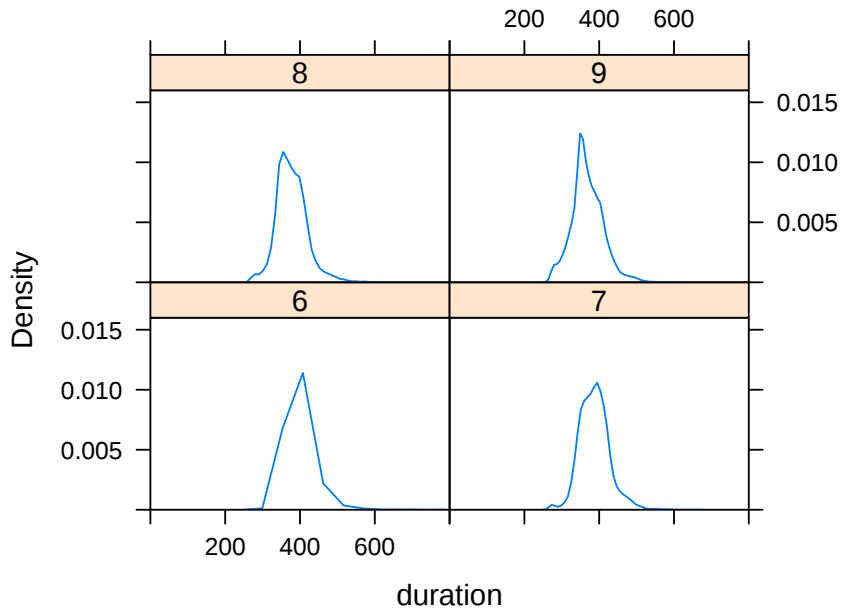
Densidades por Matta Oriente–Poniente



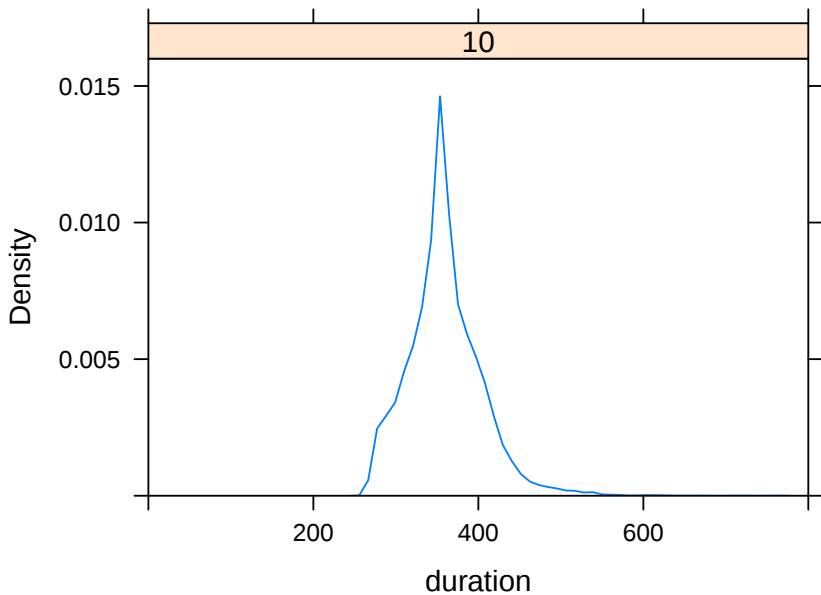
Densidades por Matta Oriente–Poniente



Densidades por Matta Oriente–Poniente



Densidades por Matta Oriente-Poniente



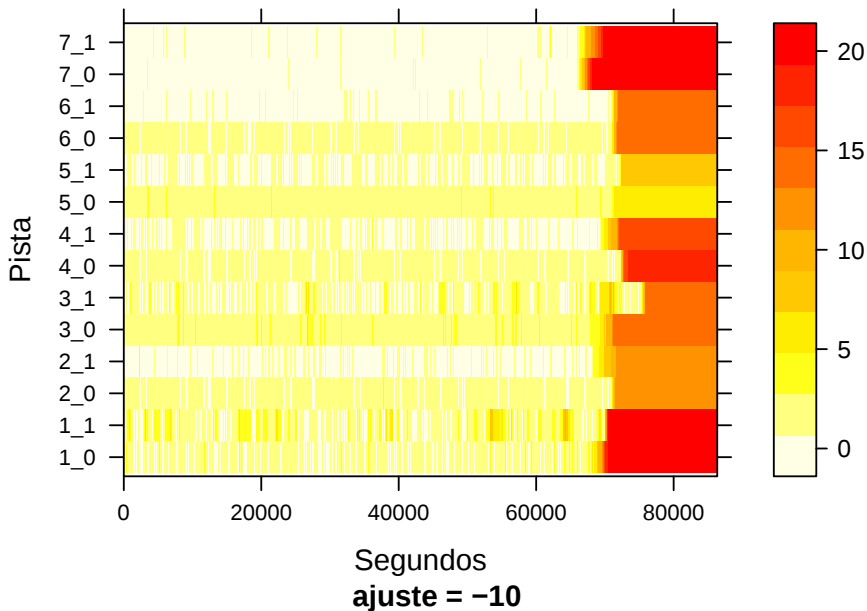
Datos de Tacos

```
library(reshape2)

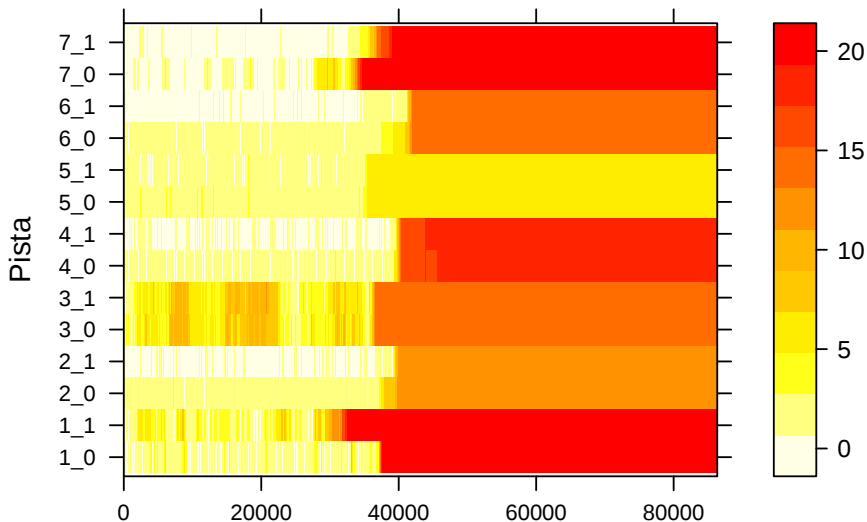
getJamData <- function(file) {
  dat <- read.csv(file)
  dat <- data.frame(
    time=as.numeric(dat$interval_begin),
    id=dat$interval_id,
    length=as.numeric(dat
      $interval_maxJamLengthInVehicles)
  )
  dcast(dat, time~id,
    mean, value.var="length") } #function
```

```
jams <- getJamData(file)
x <- jams[, 1]
y <- 1:14
z <- as.matrix(jams[, -1])
lp <- levelplot(z, row.values=x,
  column.values=y,
  col.regions=rev(heat.colors(max(z))),
  xlab="Segundos", ylab="Pista", cuts=10,
  colorkey=TRUE, aspect="fill",
  scales=list(cex=0.75,
    x=list(at=seq(0,86400, 20000),
      labels=as.character(seq(0, 86400, 20000))),
    y=list(at=1:14,
      labels=paste0(rep(1:7, each=2), "_",
        rep(0:1, 7)))))
main="Tacos en Calles que Cruzan Matta",
sub=paste0("ajuste = ", adj)
)
```

Tacos en Calles que Cruzan Matta

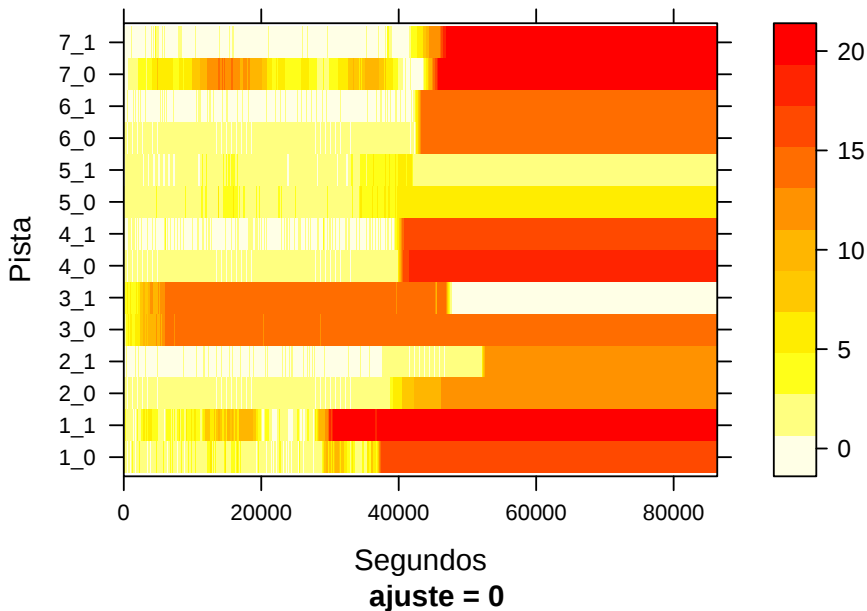


Tacos en Calles que Cruzan Matta

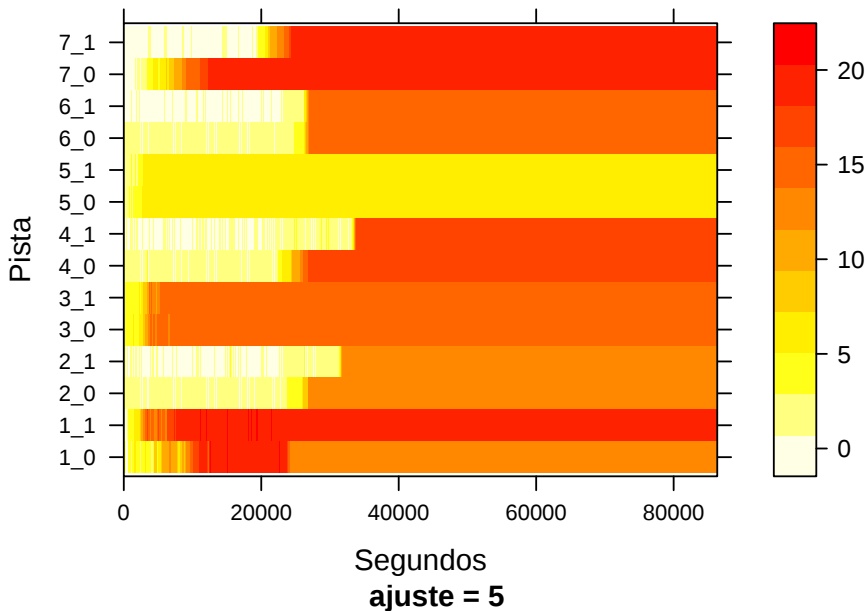


Segundos
ajuste = -5

Tacos en Calles que Cruzan Matta



Tacos en Calles que Cruzan Matta



Tacos en Calles que Cruzan Matta

