

# Lennuliiklus

*Risto Hinno*

*Friday, June 26, 2015*

## Tutvumine andmestikuga

Laadi alla andmestikud airports.csv ja flights.csv ning loe need töökeskkonda.

Andmestikus airports.csv on informatsioon lennujaamade kohta. Selles on järgnevad tunnused:

- id - lennujaama id
- name - lennujaama nimi
- city - millises linnas asub lennujaam
- country - millises riigis asub lennujaam
- lat - lennujaama asukoha laiuskraad
- lon - lennujaama asukoha pikkuskraad

Andmestik flights.csv sisaldab infot lendude kohta. Selles on järgnevad tunnused:

- airline - lennukompanii lühend
- airline\_id - lennukompanii id
- source\_airport - lähtekoha lennujaama lühend
- source\_airport\_id - lähtekoha lennujaama id
- dest\_airport - sihtkoha lennujaama lühend
- dest\_airport\_id - sihtkoha lennujaama id

```
airports=read.csv("../data/airports.csv")
airports_alg=airports
flights=read.csv("../data/flights.csv")
```

## Suurringjoon

Paul Butler kasutas suurringjooni, et visualiseerida ühendusi sõprade vahel. Kahe punkti lühim vahemaa keral avaldub suurringjoone kaudu. Kui tuult ja muid sarnaseid tegureid mitte arvestada, lendavad lennukid pikki vahemaid suurringjoonte marsruutide kaudu.

Näide, kuidas joonistada suurringjoont R-is. Ühendame Liivi 2 maja Facebooki peakontoriga.

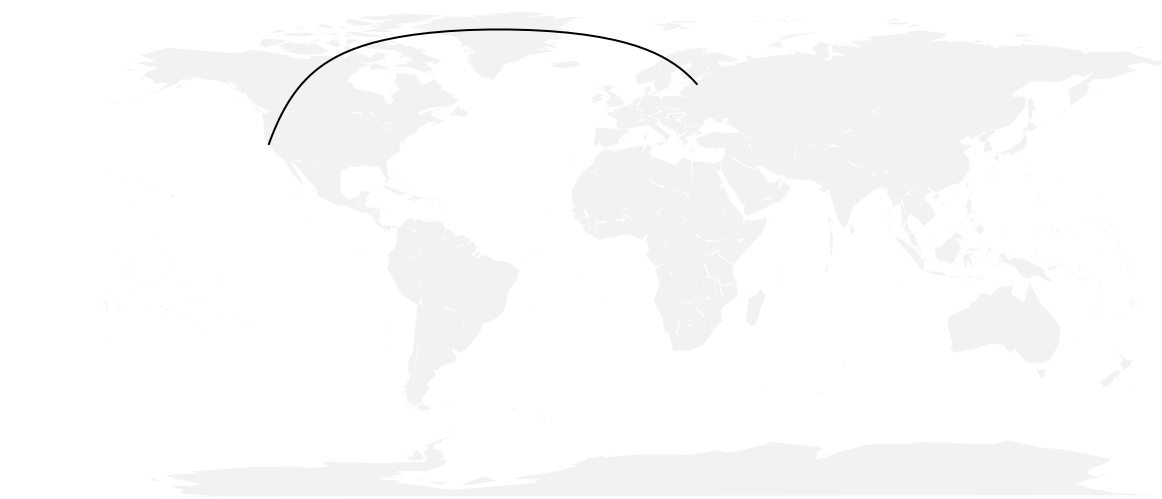
```
library(maps)
library(geosphere)
```

```
## Loading required package: sp
```

```
map("world", col="#f2f2f2", fill=TRUE, border=NA)

lat_liivi = 58.379491
lon_liivi = 26.713199
lat_fb = 37.485245
lon_fb = -122.148807

gc_points = gcIntermediate(c(lon_liivi, lat_liivi), c(lon_fb, lat_fb), n=50, addStartEnd=TRUE)
lines(gc_points)
```



## Ülesanne 1 (5 punkti)

Visualiseeri suurringjoonte abil ühendusi, kuhu saab Tallinna lennujaamast otselennuga.

Näpunäited:

- Kuna otselennuga saab Tallinnast vaid Euroopasse, siis kanna joonisele vaid Euroopa kaart. Selleks uuri, kuidas map funktsioonile ette anda koordinaate. Kasuta xlim = c(-10, 40), ylim = c(30, 70).
- Tallinna lennujaama kood on 415 (mille võid leida, kui uurid andmestikust airports Eesti lennujaamu).
- Sihtkohtade lennujaamad leiad andmestikust flights.csv.
- Lennujaamade koordinaadid leiad andmestikus airports.csv.

```

allikas=airports[, c("id", "lat", "lon")]
names(allikas)=c("id", "source_airport_lat", "source_airport_lon")
data=merge(flights, allikas, by.x="source_airport_id", by.y="id", all.x=T, all.y=F)
names(allikas)=c("id", "dest_airport_lat", "dest_airport_lon")
data=merge(data, allikas, by.x="dest_airport_id", by.y="id", all.x=T, all.y=F)
#teeme subseti ainult neist andmetest, kus alguspunkt on tallinn
data2=subset(data, source_airport_id==415)
#teeme kaardi
map(xlim = c(-10, 40), ylim = c(30, 70), col="#f2f2f2", fill=TRUE, border=NA)
#itereerime punktid peale
for (i in 1:nrow(data2)) {
  points = gcIntermediate(c(data2$source_airport_lon[i],
                             data2$source_airport_lat[i]),
                           c(data2$dest_airport_lon[i],
                             data2$dest_airport_lat[i]), n=50,
                           addStartEnd=TRUE)

  lines(points)
}

```



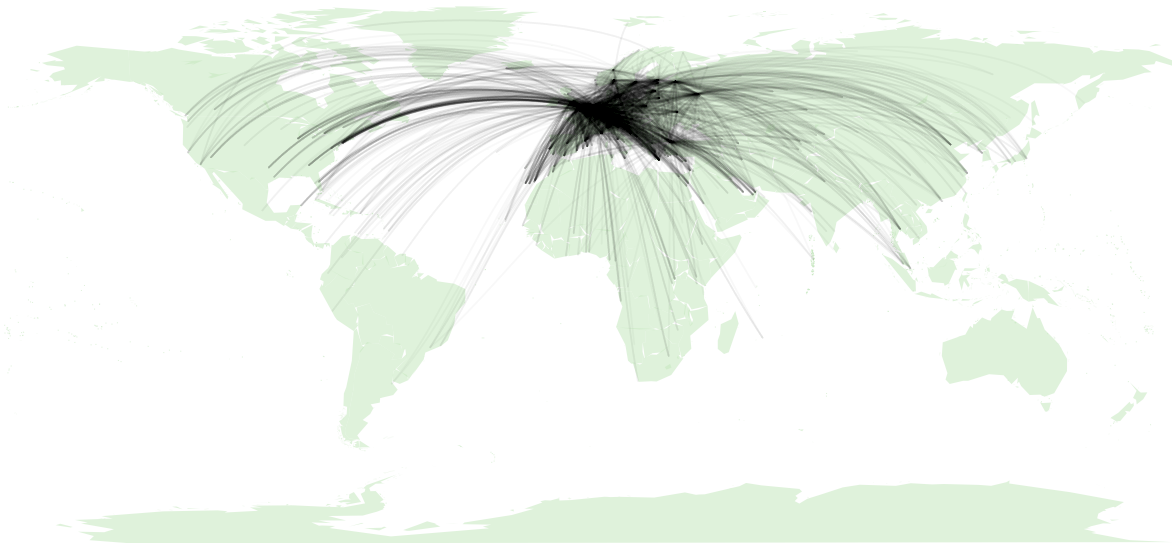
## Ülesanne 2 (9 punkti)

Visualiseeri ühendusi, kuhu saab Tallinna lennujaamast otselennu või ühe vahepeatusega.

Näpunäited:

- Et värve muuta läbipaistvaks, lisa värvikoodile lõppu läbipaistvuse intensiivsus. Näiteks kasuta funktsiooni `lines` korral argumenti `col="#00000010"`, et muuta mustade (värvikood `#000000`) joonte läbipaistvus 10%-ni.
- Visualiseeri ühendusi Tallinn - vahelennujaam - lõpp-punkt. Tingimata pole tarvis mitu korda joonistada Tallinn - vahelennujaam ühendusi, need võid joonistada ka ühe korra.

```
airports=unique(data2$dest_airport_id)
data3=subset(data, source_airport_id==415| source_airport_id %in% airports)
#teeme mapi
map("world",col="#c7e9c090", fill=TRUE, border=NA,mar = c(1, 1, par("mar")[3], 0.1))
#itereerime punktid peale
#Et värve muuta läbipaistvaks, lisa värvikoodile lõppu läbipaistvuse intensiivsus.
#Näiteks kasuta funktsiooni lines korral argumenti col="#00000010", et muuta
#mustade (värvikood #000000) joonte läbipaistvus 10%-ni.
for (i in 1:nrow(data3)) {
  points = gcIntermediate(c(data3$source_airport_lon[i],
                             data3$source_airport_lat[i]),
                           c(data3$dest_airport_lon[i],
                             data3$dest_airport_lat[i]), n=50,
                           addStartEnd=TRUE)
  lines(points, col="#00000008")
}
```



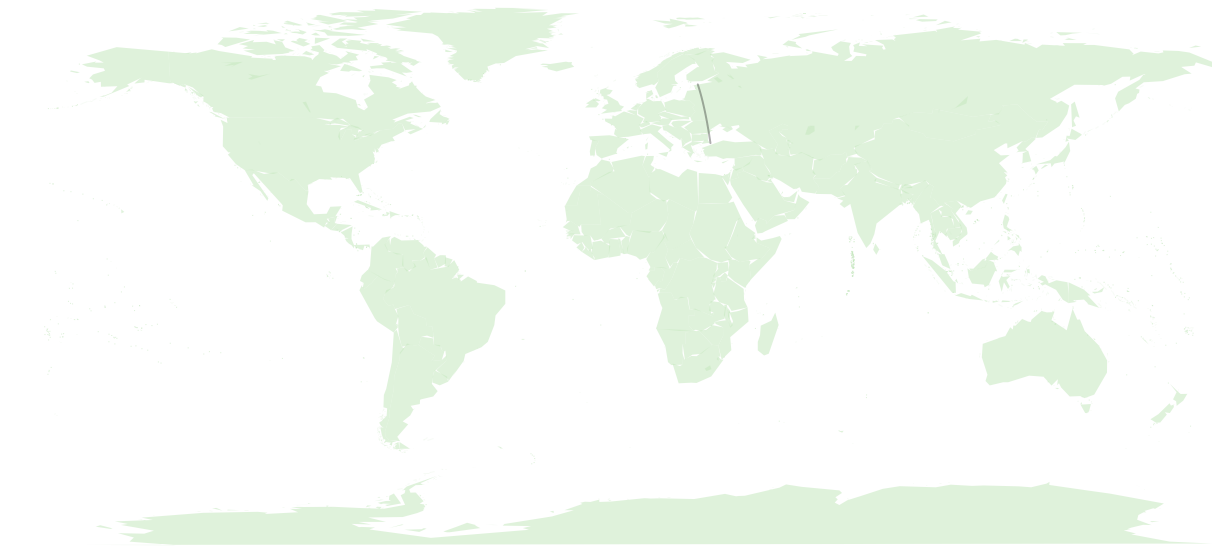
## Boonusülesanne 1 (1 punkt) - Juhuslik ekslemine eralennukiga

Alusta Tallinna lennujaamast. Vali kõikvõimalikest lennujaamadest juhuslikult üks ja lenda sinna. Vali nüüd juhuslikult järgmine sihtpunkt ning lenda sinna. Jätka seda protsessi 100 korral ja visualiseeri läbitud teekonda.

```
map("world",col="#c7e9c090", fill=TRUE, border=NA,mar = c(1, 1, par("mar")[3], 0.1))

#teeme alguse ära, tallinnast esimene liin
data_subset=subset(data, source_airport_id==415)
destination=sample(data_subset$dest_airport_id,1)
data_subset=subset(data_subset, source_airport_id==415&
                    dest_airport_id==destination ) [1,] #võtame esimese rea
#kuna võib olla mitu liini ühte kohta samast lennujaamast

point=gcIntermediate(c(data_subset$source_airport_lon,
                      data_subset$source_airport_lat),
                    c(data_subset$dest_airport_lon,
                      data_subset$dest_airport_lat), n=50,
                    addStartEnd=TRUE)
lines(point, col="#00000050")
```



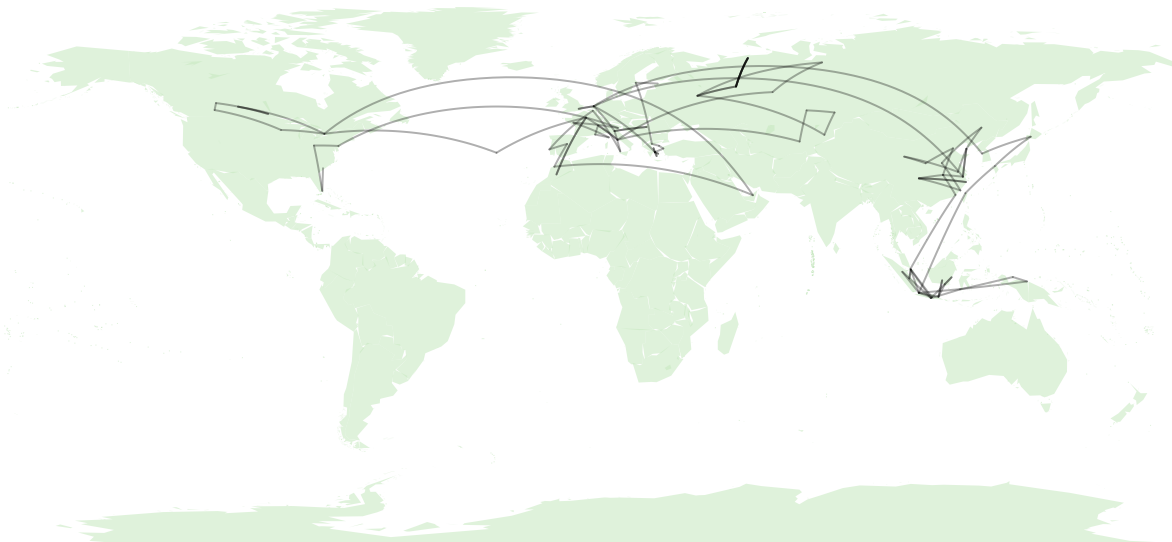
```
#kordame 100 korda
library(maps)
library(geosphere)
```

```

map("world",col="#c7e9c090", fill=TRUE, border=NA,mar = c(1, 1, par("mar")[3], 0.1))
#teeme alguse ära, tallinnast esimene liin
data_subset=subset(data, source_airport_id==415)
destination=sample(data_subset$dest_airport_id,1)
data_subset=subset(data_subset, source_airport_id==415&
                    dest_airport_id==destination )[1,] #võtame esimese rea
#kuna võib olla mitu liini ühte kohta samast lennujaamast

point=gcIntermediate(c(data_subset$source_airport_lon,
                        data_subset$source_airport_lat),
                    c(data_subset$dest_airport_lon,
                        data_subset$dest_airport_lat), n=50,
                    addStartEnd=TRUE)
lines(point, col="#00000050")
for (i in 1:100) {
  data_subset=subset(data, source_airport_id==destination)
  destination=sample(data_subset$dest_airport_id,1)
  data_subset=subset(data_subset, dest_airport_id==destination )[1,]
  points=gcIntermediate(c(data_subset$source_airport_lon,
                           data_subset$source_airport_lat),
                        c(data_subset$dest_airport_lon,
                           data_subset$dest_airport_lat), n=50,
                        addStartEnd=TRUE)
  lines(points, col="#00000050")
}

```



## Boonusülesanne 2 (2 punkti) - Juhuslik ekslemine kasutades kommertsliine

Alusta Tallinna lennujaamast. Vali võimalikest sihtkohtadest üks ja lenda sinna. Vali selle lennujaama võimalikest sihtkohtadest juhuslikult järgmine ning lenda sinna. Jätka seda protsessi 100 korral ja visualiseeri läbitud teekonda.

```
map("world",col="#c7e9c090", fill=TRUE, border=NA,mar = c(1, 1, par("mar")[3], 0.1))

alg=subset(airports_alg, id==415)
samp=sample(airports_alg$id, 1)
dest=subset(airports_alg, id==samp)
point=gcIntermediate(c(alg$lon,
                       alg$lat),
                    c(dest$lon,
                      dest$lat), n=50,
                    addStartEnd=TRUE)
lines(point, col="#00000050")

for (i in 1:100) {
  samp1=sample(airports_alg$id,1)
  alg=subset(airports_alg, id==samp1)
  samp2=sample(airports_alg$id,1)
  dest=subset(airports_alg, id==samp2)
  point=gcIntermediate(c(alg$lon,
                        alg$lat),
                      c(dest$lon,
                        dest$lat), n=50,
                      addStartEnd=TRUE)
  lines(point, col="#00000050")
}
```

