Does the movement of students to a different region of England for education associate with increased lonliness in their county of origin?

Research question

In addition to our general visualsiation of the datasets provided by the JGI Data Competition and others sourced online, we wanted to explore the hypothesis that students leaving their domicile to travel to a different region of the control for higher education might be associated with the lonliness score in the domicile.

We wanted to examine this question because we believe the effect of student migration on those left behind is often overlooked in terms of analysis/interventions.

Analysis

Creating the domiciles/towns, counties and regions datasets

```
# Import town list
# Sourced from here: https://www.paulstenning.com/wp-content/uploads/2015/05/Towns_List.csv
town.county <- read.csv("data/Towns_List.csv")
town.county <- town.county[which(town.county$Country=="England"),]
town.county <- town.county[,c(1:2)]

# Import county list
# Sourced from here: https://wiki.freecycle.org/UK_Counties_and_Regions
county.region <- read.csv("data/county.region.csv", stringsAsFactors=FALSE, header = TRUE)
colnames(county.region)[1] <- "County"

# Merge above datasets on county column
town.county.region <- merge(town.county, county.region)
```

Preparation for migration analysis

```
# Read in required data
# Sourced from here: https://www.hesa.ac.uk/data-and-analysis/students/where-from
data.move <- read.csv("data/Movement data.csv", header = FALSE)

# Remove description of dataset
data.move <- data.move[18:nrow(data.move),]

# Add first row to column headers and delete first row
colnames(data.move) = make.names(as.character(unlist(data.move[1, ])))
data.move = data.move[-1, ]

# Convert to tibble
data.move <- as_tibble(data.move)
```

```r
# Remove "City of" from Domicile column to aid merging
data.move <- data.move %>%
  mutate(Domicile = gsub("City of ", "", Domicile))

# Merge on basis on county name
tmp1 <- merge(data.move, county.region, by.x = "Domicile", by.y = "County")
colnames(tmp1)[1] <- "County"

# Merge on basis on town name
tmp2 <- merge(data.move, town.county.region, by.x = "Domicile", by.y = "Town")
tmp2 <- tmp2[,c(2:9)]

# COmbine the two merged datasets created above
countydata <- rbind(tmp1, tmp2)
```

Matching dates

We ran into issues around how to convert the academic year (used in the student migration data) to the calendar year (used in the lonliness index data).

We decided to assign the full effect of a given academic year to it's earliest calendar year, on the basis that the effect of student migration on lonliness is likely to be most acute in the months immediately after their departure.

```r
countydata <- countydata %>%
  mutate(
  Academic.Year = gsub("2014/15", "2014", Academic.Year),
  Academic.Year = gsub("2015/16", "2015", Academic.Year),
  Academic.Year = gsub("2016/17", "2016", Academic.Year),
  Academic.Year = gsub("2017/18", "2017", Academic.Year)
  )
```

Number of students leaving their county to travel to another region

We decided to examine the effect of moving to another region to study, as a proxy for a substantial distance between home town and place of study. We removed students enrolled in the open university, as this is primarily a distance learning insitution, and so students would not have to travel for education.

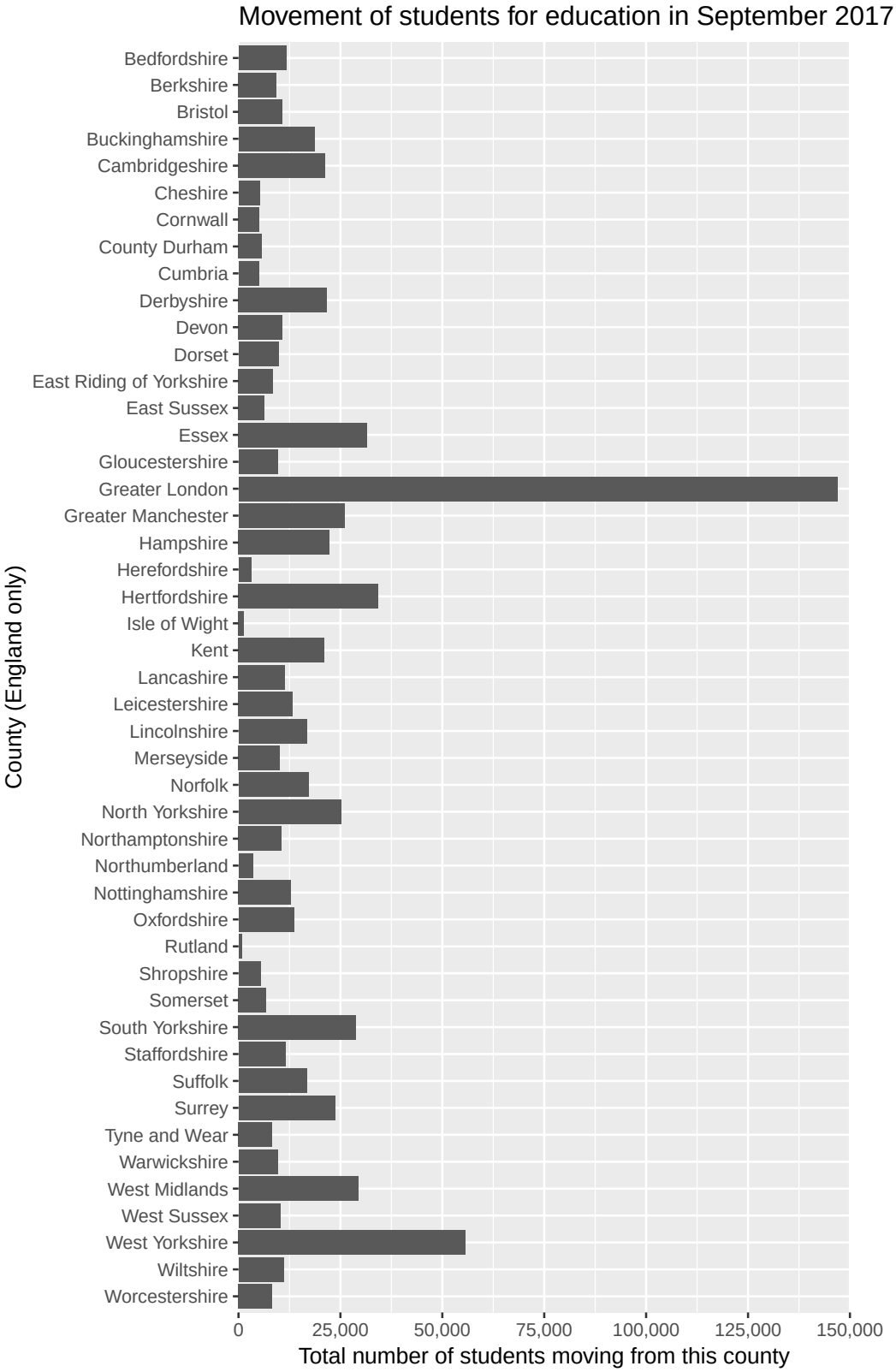
We also removed all those whose destintion was in the same region which they came from. This was used as a proxy for a substantial distance between home town and university.

```r
# Students moving from domicile ######################################################
# Remove rows corresponding to totals to prevent double counting
countydata.trimmed.out <- filter(
  countydata,
  Region.of.HE.provider != "Total England" &
  Region.of.HE.provider != "Total United Kingdom" &
  Region.of.HE.provider != "The Open University in England" &
  Region.of.HE.provider != Region
  )
```

We then summed all those moving to a from a certain area to create an outward migration for education statistic. The plot below shows the results of this analysis for 2017.

Note: the plot below shows 47 of the 48 counties in England. We limited the analysis to England as we only had lonliness data for this country. Further, the City of London is not shown below, as it was included as part of Greater London in the student migration data.

```
# Create sum of people moving from each domicile
data.outof.domicile <- filter(countydata.trimmed.out, X4.way.domicile == "All", Level.of.study == "All"
  group_by(Academic.Year, County) %>%
  summarise(sum.no = sum(as.numeric(as.character(Number))))

fig1.data <- data.outof.domicile[which(data.outof.domicile$Academic.Year=="2017"),]

ggplot(fig1.data, aes(x = fct_rev(County), y=sum.no)) +
  geom_bar(stat="identity") +
  coord_flip() +
  labs(x = "County (England only)",
       y = "Total number of students moving from this county",
       title = "Movement of students for education in September 2017") +
  scale_y_continuous(breaks=seq(0, 175000, 25000),label = scales::comma, expand=c(0,0))+
  expand_limits(y=150000) +
  theme(plot.margin = unit(c(1,1,1,1), "cm"))
```


Movement of students for education in September 2017
Bedfordshire
Berkshire
Bristol
Buckinghamshire
Cambridgeshire
Cheshire
Cornwall
County Durham
Cumbria
Derbyshire
Devon
Dorset
East Riding of Yorkshire
East Sussex
Essex
Gloucestershire
Greater London
Greater Manchester
Hampshire
Herefordshire
Hertfordshire
Isle of Wight
Kent
Lancashire
Leicestershire
Lincolnshire
Merseyside
Norfolk
North Yorkshire
Northamptonshire
Northumberland
Nottinghamshire
Oxfordshire
Rutland
Shropshire
Somerset
South Yorkshire
Staffordshire
Suffolk
Surrey
Tyne and Wear
Warwickshire
West Midlands
West Sussex
West Yorkshire
Wiltshire
Worcestershire
County (England only)
0
25,000
50,000
75,000
100,000
125,000
150,000
Total number of students moving from this county

Association of lonliness with outward migration for education

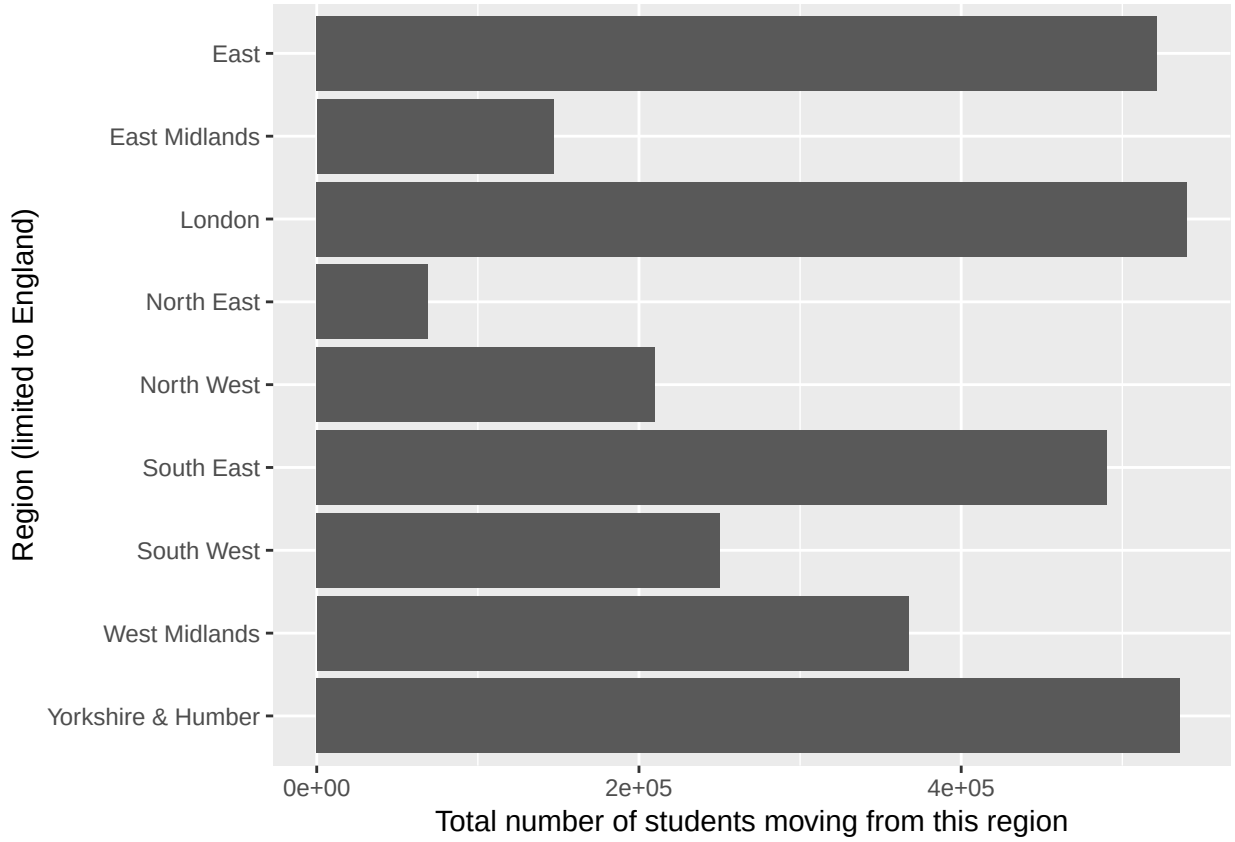
Total change in population due to migration for education

In addition to the above analysis we decide to examine the effect of total migration for education (number of students leaving the region + number of students arriving in the region) on lonliness on that region.

We created a summative rather than net migration for education per region figure on the basis that having students moving into a region is more likely to increase lonliness (as they may be homesick) than reduce it, as they will not keep up the social contacts that the students who left had, plus they might possibly be homesick/lonely themselves.

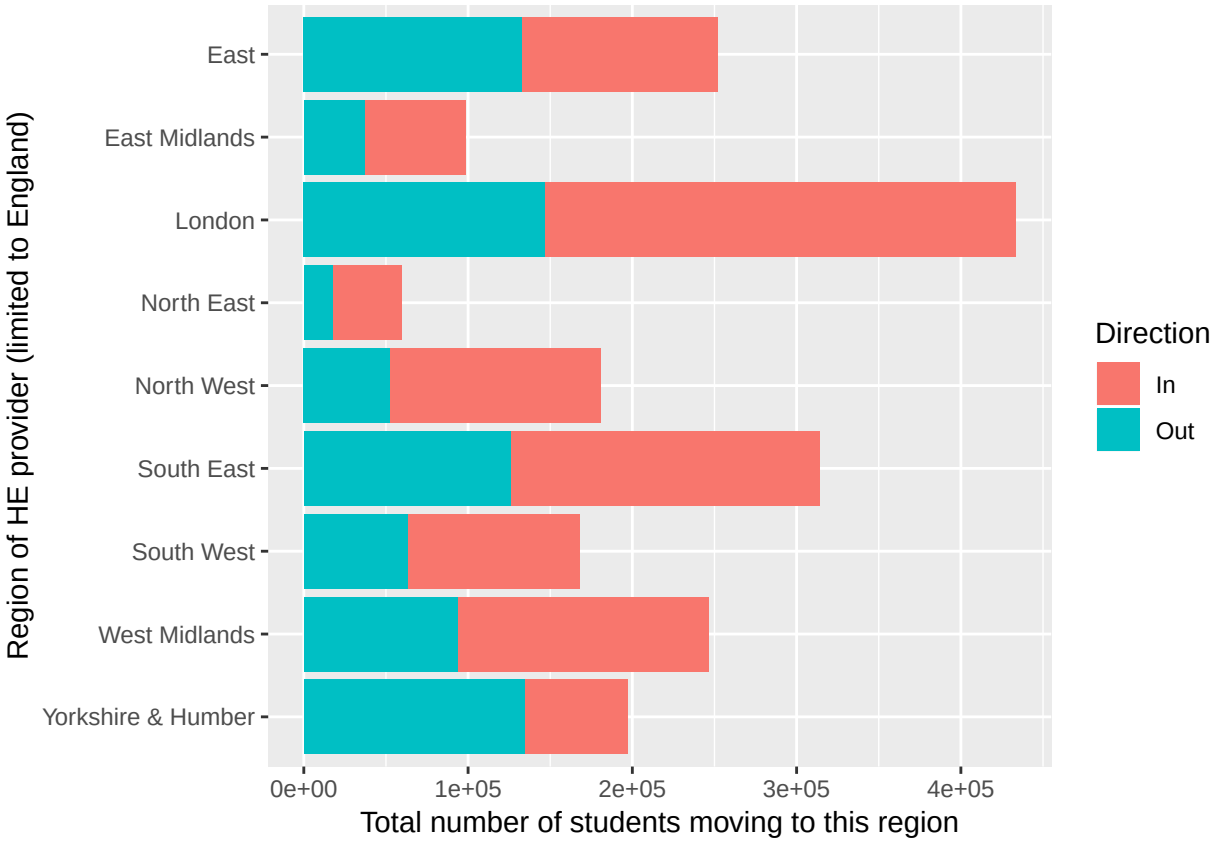
From a region

```
# Building on the intital analysis done above
data.outof.region <- filter(countydata.trimmed.out, X4.way.domicile == "All", Level.of.study == "All", L
  group_by(Academic.Year, Region) %>%
  summarise(no.out = sum(as.numeric(as.character(Number))))

ggplot(data.outof.region, aes(x = fct_rev(Region), y=no.out)) +
  geom_bar(stat="identity") +
  coord_flip() +
  labs(x = "Region (limited to England)",
       y = "Total number of students moving from this region")
```



Into a region

```r
#Limit dataset to those travelling to HE provider within England
countydata.trimmed.in <- filter(
  countydata,
  Region.of.HE.provider == "West Midlands" |
    Region.of.HE.provider == "Yorkshire and the Humber" |
    Region.of.HE.provider == "South West" |
    Region.of.HE.provider == "South East" |
    Region.of.HE.provider == "North West" |
    Region.of.HE.provider == "North East" |
    Region.of.HE.provider == "London" |
    Region.of.HE.provider == "East of England"|
    Region.of.HE.provider == "East Midlands"
)

# Create sum of people moving into each region of England
data.into.region <- filter(countydata.trimmed.in, X4.way.domicile == "All", Level.of.study == "All", Mo
  group_by(Academic.Year, Region) %>%
  summarise(no.in = sum(as.numeric(as.character(Number))))

fig2.data.1 <- data.into.region[which(data.into.region$Academic.Year=="2017"),]
fig2.data.1[4] <- "In"
colnames(fig2.data.1)[4] <- "Direction"
colnames(fig2.data.1)[3] <- "No"
fig2.data.2 <- data.outof.region[which(data.outof.region$Academic.Year=="2017"),]
fig2.data.2[4] <- "Out"
colnames(fig2.data.2)[4] <- "Direction"
colnames(fig2.data.2)[3] <- "No"

fig2.data <- rbind(fig2.data.1,fig2.data.2)


# Plot results
ggplot(fig2.data, aes(x = fct_rev(Region), y= No, fill = Direction)) +
  geom_bar(stat="identity") +
  coord_flip() +
  labs(x = "Region of HE provider (limited to England)",
       y = "Total number of students moving to this region")
```



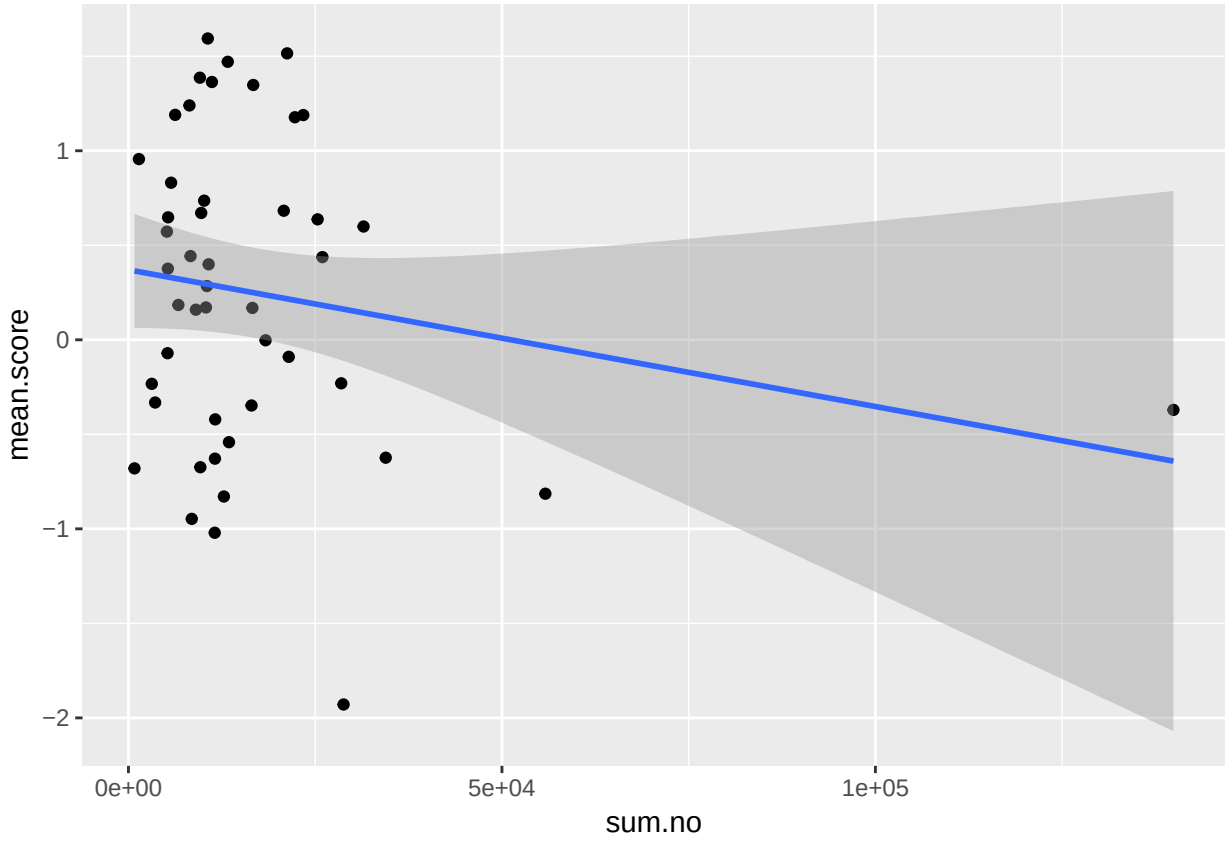
Combine in/out migration numbers

```
data.sum.migration <- merge(data.into.region,data.outof.region)

data.sum.migration <- data.sum.migration %>%
  mutate(no.total = no.in + no.out)
```

```
data.lonliness <- read.csv("data/msoa_loneliness.csv")

data.lonliness <- data.lonliness %>%
  mutate(District = gsub('[[:digit:]]+', '', msoa11nm),
         District = gsub(" $","", District, perl=T))


# Add County Level
district.county <- read.csv("data/district-county.csv")
colnames(district.county)[1] <- "District"

data.lonliness <- merge(data.lonliness,district.county,by="District",all.x=TRUE)

data.lonliness$County <- as.character(data.lonliness$County)

data.lonliness$County <- ifelse(is.na(data.lonliness$County) == TRUE,"0", data.lonliness$County)

data.lonliness$County <- ifelse(data.lonliness$District == "Bournemouth", "Dorset", data.lonliness$Coun
```

```
data.lonliness$County <- ifelse(data.lonliness$District == "Poole", "Dorset", data.lonliness$County)

data.lonliness$County <- ifelse(data.lonliness$District == "Christchurch", "Dorset", data.lonliness$Cou

data.lonliness$County <- ifelse(data.lonliness$District == "East Dorset", "Dorset", data.lonliness$Coun

data.lonliness$County <- ifelse(data.lonliness$District == "West Dorset", "Dorset", data.lonliness$Coun

data.lonliness$County <- ifelse(data.lonliness$District == "North Dorset", "Dorset", data.lonliness$Cou

data.lonliness$County <- ifelse(data.lonliness$District == "Purbeck", "Dorset", data.lonliness$County)

data.lonliness$County <- ifelse(data.lonliness$District == "Shepway", "Kent", data.lonliness$County)

data.lonliness$County <- ifelse(data.lonliness$District == "St Edmundsbury", "West Suffolk", data.lonli

data.lonliness$County <- ifelse(data.lonliness$District == "St. Helens", "Merseyside", data.lonliness$C

data.lonliness$County <- ifelse(data.lonliness$District == "Forest Heath", "West Suffolk", data.lonline

data.lonliness$County <- ifelse(data.lonliness$District == "Suffolk Coastal", "East Suffolk", data.lonl

data.lonliness$County <- ifelse(data.lonliness$District == "Waveney", "East Suffolk", data.lonliness$Co

data.lonliness$County <- ifelse(data.lonliness$District == "Taunton Deane", "Somerset West and Taunton"

data.lonliness$County <- ifelse(data.lonliness$District == "West Somerset", "Somerset West and Taunton"

data.lonliness$County <- ifelse(data.lonliness$District == "Weymouth and Portland", "Dorset", data.lonl

# Add Region
county.region <- read.csv("data/county.region.csv")
colnames(county.region)[1] <- "County"

data.lonliness <- merge(data.lonliness, county.region, by="County",all.x=TRUE)



# Summarise by county
mean.lonliness.county <- data.lonliness %>%
  group_by(County) %>%
  summarise("2016" = mean(loneills_2016),
            "2017" = mean(loneills_2017),
            "2018" = mean(loneills_2018))

mean.lonliness.county <- gather(mean.lonliness.county, Academic.Year, mean.score, "2016","2017","2018",

# Merge with movement data
move.lonliness.county <- merge(mean.lonliness.county,data.outof.domicile)

move.lonliness.county.2016 <- move.lonliness.county[which(move.lonliness.county$Academic.Year=="2016"),]

ggplot(move.lonliness.county.2016, aes(x=sum.no,y=mean.score))+
```

```
  geom_point() +
  geom_smooth(method='lm')
```




```
cor(move.lonliness.county.2016$mean.score, move.lonliness.county.2016$sum.no)
```

```
## [1] -0.1862
```

```
linearMod <- lm(mean.score ~ sum.no, data=move.lonliness.county.2016)
print(linearMod)
```

```
##
## Call:
## lm(formula = mean.score ~ sum.no, data = move.lonliness.county.2016)
##
## Coefficients:
## (Intercept)       sum.no
##    3.70e-01    -7.23e-06
```

```
summary(linearMod)
```

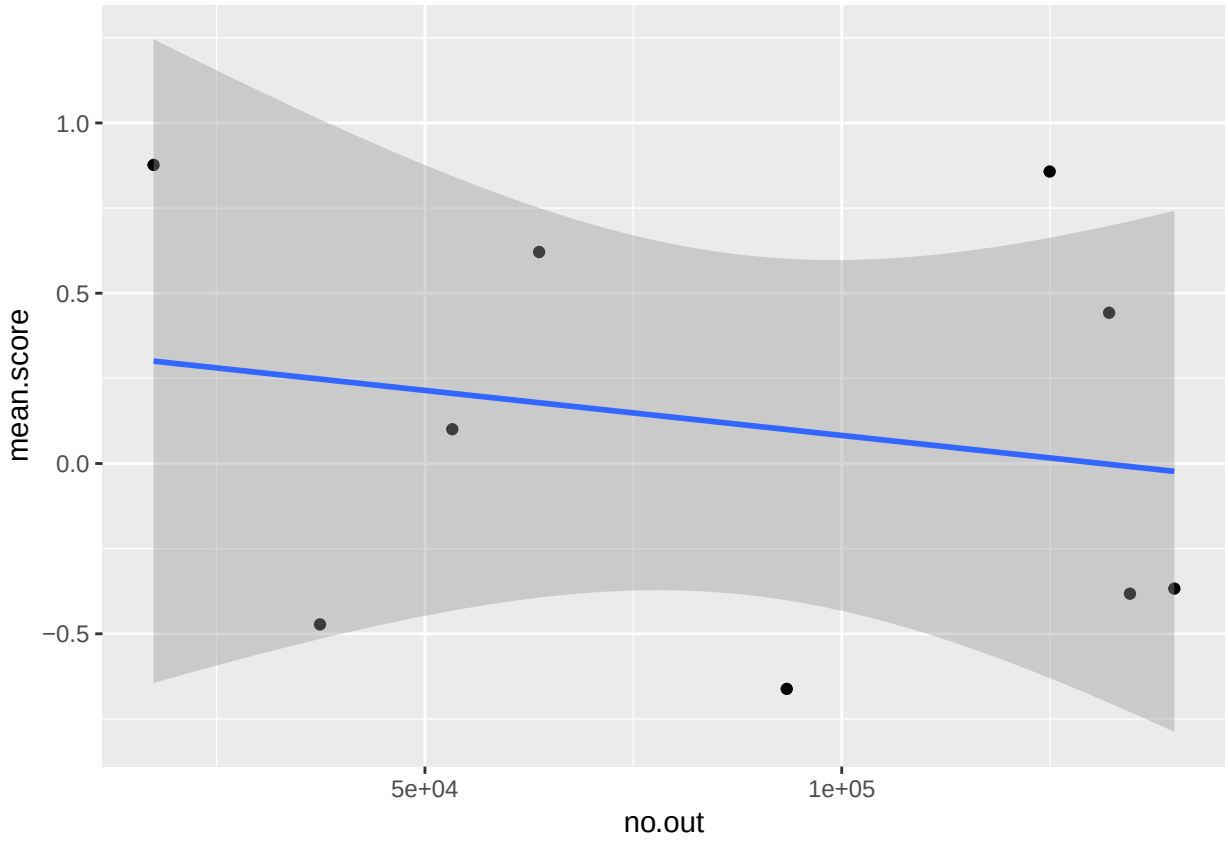
```
##
## Call:
## lm(formula = mean.score ~ sum.no, data = move.lonliness.county.2016)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.091 -0.637  0.044  0.482  1.300
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.70e-01   1.53e-01    2.43    0.019
## sum.no      -7.23e-06   5.69e-06   -1.27    0.210
##
## Residual standard error: 0.81 on 45 degrees of freedom
## Multiple R-squared:  0.0347, Adjusted R-squared:  0.0132
## F-statistic: 1.62 on 1 and 45 DF,  p-value: 0.21
```

```
# Summarise by Region
mean.lonliness.region <- data.lonliness %>%
  group_by(Region) %>%
  summarise("2016" = mean(loneills_2016),
            "2017" = mean(loneills_2017),
            "2018" = mean(loneills_2018))
```

```
## Warning: Factor `Region` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

```
mean.lonliness.region <- gather(mean.lonliness.region, Academic.Year, mean.score, "2016","2017","2018",

# Merge with movement data
move.lonliness.region <- merge(mean.lonliness.region,data.outof.region)

move.lonliness.region.2016 <- move.lonliness.region[which(move.lonliness.region$Academic.Year=="2016"),]

ggplot(move.lonliness.region.2016, aes(x=no.out,y=mean.score))+
  geom_point() +
  geom_smooth(method='lm')
```




```
cor(move.lonliness.region.2016$mean.score, move.lonliness.region.2016$no.out)
```

```
## [1] -0.2047
```

```
linearMod <- lm(mean.score ~ no.out, data=move.lonliness.region.2016)
summary(linearMod)
```

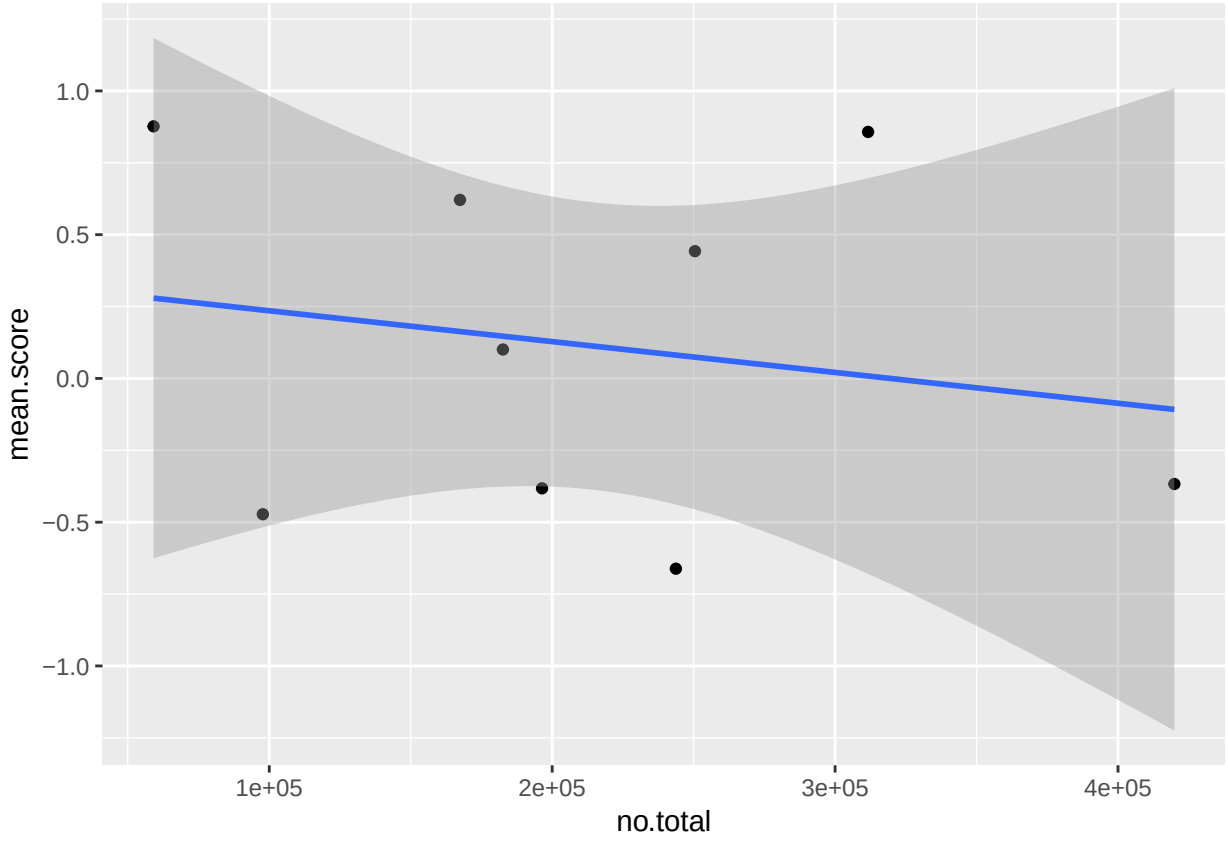
```
##
## Call:
## lm(formula = mean.score ~ no.out, data = move.lonliness.region.2016)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.762 -0.373 -0.105  0.445  0.841
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.47e-01   4.73e-01    0.73     0.49
## no.out      -2.64e-06   4.78e-06   -0.55     0.60
##
## Residual standard error: 0.633 on 7 degrees of freedom
## Multiple R-squared:  0.0419, Adjusted R-squared:  -0.095
## F-statistic: 0.306 on 1 and 7 DF,  p-value: 0.597
```

```
data.sum.migration.2016 <- data.sum.migration[,c(1,2,5)]
data.sum.migration.2016 <- data.sum.migration.2016[which(data.sum.migration.2016$Academic.Year==2016),]
```

```r
move.lonliness.region.2016 <- move.lonliness.region.2016[,c(1:3)]

total.move.lonliness.region.2016 <- merge(data.sum.migration.2016,move.lonliness.region.2016)

cor(total.move.lonliness.region.2016$mean.score, total.move.lonliness.region.2016$no.total)
```

```
## [1] -0.1935
```

```r
ggplot(total.move.lonliness.region.2016, aes(x=no.total,y=mean.score))+
  geom_point() +
  geom_smooth(method='lm')
```




```r
linearMod <- lm(mean.score ~ no.total, data=total.move.lonliness.region.2016)
summary(linearMod)
```

```
##
## Call:
## lm(formula = mean.score ~ no.total, data = total.move.lonliness.region.2016)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7430 -0.5143 -0.0461  0.4579  0.8490
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.43e-01   4.89e-01    0.70     0.51
## no.total    -1.07e-06   2.06e-06   -0.52     0.62
```

```
##
## Residual standard error: 0.634 on 7 degrees of freedom
## Multiple R-squared:  0.0374, Adjusted R-squared:  -0.1
## F-statistic: 0.272 on 1 and 7 DF,  p-value: 0.618
```