PO91Q - Fundamentals in Quantitative Research Methods:

Week 8 - Case Study Solutions

Dr Florian Reiche
F.Reiche@warwick.ac.uk

1 | Core Exercises

1.1 Load Packages and Data

Before starting, we need to load libraries and install packages if not already installed. In these exercises we will be using the following packages:

1. `haven`
2. `ggplot2`
3. `modelsummary`

We will be using the data set from the lecture, but with a different independent variable this time. These will be of particular interest:

Variable	Label	Year
`const`	Parliamentary constituency	n/a
`gcse`	An average score based on a pupil's best eight grades in a group of GCSEs. The maximum a pupil can achieve is 90 points.	2019
`eth_min`	Percent of population composed by ethnic minorities	2011
`idaci`	The Income Deprivation Affecting Children Index rank - how it compares to other constituencies	2015/16
`income`	Mean income by constituency	2017/18

Table 1: Codebook for `london` Data Set

Data are taken from House of Commons Library (n.d.), GOV.UK (2013) and London Data Store (2010). The file is on Moodle. Set your working directory and load the data.

Hint: Note, the data is a .csv file

```
setwd("")

library(tidyverse)
library(haven)
library(modelsummary)

london <- read.csv("london.csv", stringsAsFactors = T)
```

1.2 Inspect your data

Here you can use several basic functions. The dataset does not contain too many variables, so you can start by using `names()`, `str()`, etc.

```
names(london)
## [1] "const"   "gcse"    "income"  "eth_min" "idaci"
dim(london)
## [1] 73  5
```

1.3 Preliminary Analysis

Let's say we want to look at the relationship between income deprivation affecting children and GCSE scores. The two variables are, respectively, `idaci` and `gcse`.

Now, formulate the working (alternative) and the null hypothesis. Write them down.

H_0: Income Deprivation affecting children has no statistical relationship with GCSE scores.

H_1: The higher the level of income deprivation affecting children, the lower the GCSE score in a constituency.

Which is your dependent variable? GCSE Scores

Run a frequency table on the `idaci` variable. Does this distribution make sense? Why/why not?

```
table(london$idaci)
##
##   3   6   8  10  13  14  25  28  30  31  35  37  39  41  42  47  49  53  69  73
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  76  83  90  93  95 102 104 109 117 118 122 134 148 158 161 169 172 175 185 189
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 194 204 208 216 223 224 229 230 237 243 249 252 254 260 262 263 284 286 298 299
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 311 314 326 343 363 376 384 422 430 457 484 495 507
##   1   1   1   1   1   1   1   1   1   1   1   1   1
class(london$idaci)
## [1] "integer"
```

Do the same for the other variable. And guess what is the level of measurement.

```
table(london$gcse)
##
## 39.6375766016713 42.3877005347594 43.5156724611162 43.6321506352087
##                1                1                1                1
## 43.6429299796057 43.7229046242775 43.8195524146054 43.8742537313433
##                1                1                1                1
##  44.545652173913 44.6200495662949 45.0188442211055 45.4591981132075
##                1                1                1                1
## 45.5580415045396 45.6319444444444 45.6929280397022 45.7884141331142
##                1                1                1                1
```

```
## 45.7985537190083 45.8345979899497 46.2872762863535 46.3694331983806
##                1                1                1                1
## 46.4130558722919 46.6391646586345  46.791858500528 47.1616575591985
##                1                1                1                1
## 47.3031669535284 47.7043795620438 48.0539196787149 48.0549388443695
##                1                1                1                1
## 48.0593713620489 48.3177836411609 48.6103896103896  48.622491638796
##                1                1                1                1
## 48.7450781781101 48.9074889867841 49.1402549019608 49.2920081967213
##                1                1                1                1
## 49.3295702479339 49.3976114381833 49.5410915104741 49.7374347258486
##                1                1                1                1
## 49.7733236151604 49.9164532650448 49.9351145038168 50.0100238095238
##                1                1                1                1
## 50.1330469715698 50.2069214437367 50.4787791342952 50.5899585635359
##                1                1                1                1
## 50.6350039215686 50.9429615384615 51.4124698133919  51.720685483871
##                1                1                1                1
## 51.8271096345515 52.1899132111861 52.2372470373747 52.4211532125206
##                1                1                1                1
## 52.6092592592593  52.993068833652 53.2315436241611 53.3844342226311
##                1                1                1                1
## 53.4207509505703 53.5521834061135 54.1110552763819 54.2496577243293
##                1                1                1                1
## 55.1015023041475  55.203789893617 55.7076301806589 55.8317191283293
##                1                1                1                1
## 55.9581151832461  58.168152350081 58.4374216710183 58.5606093432634
##                1                1                1                1
## 60.1115643105446
##                1
class(london$gcse)
## [1] "numeric"
```

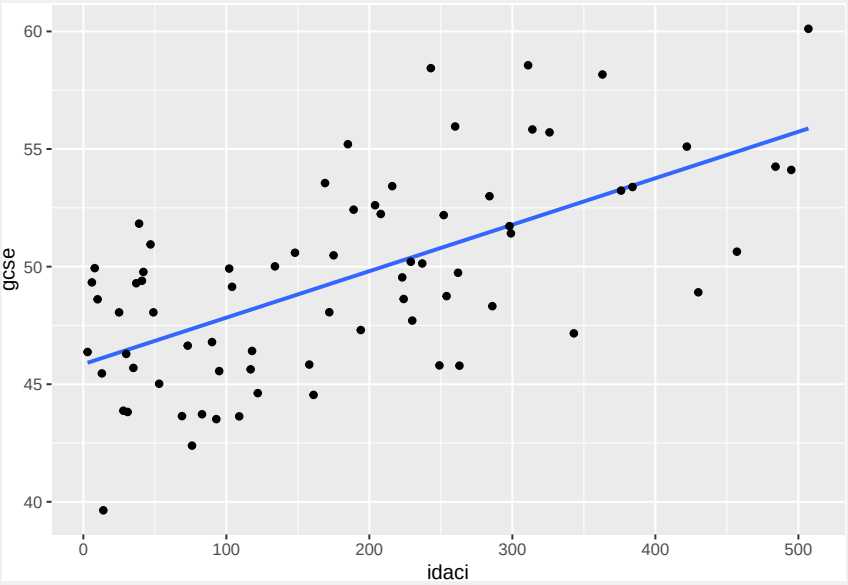
1.4 Visualisation

Let's start with the visualisation of the relationship between the two variables. What is the best way to visualise the relationship considering the level of measurement of our variables?

Hint: Probably a scatterplot, right? So, use a scatterplot to visualise the relationship and add the regression line.

You can use `ggplot`, but also the standard `plot()` function.

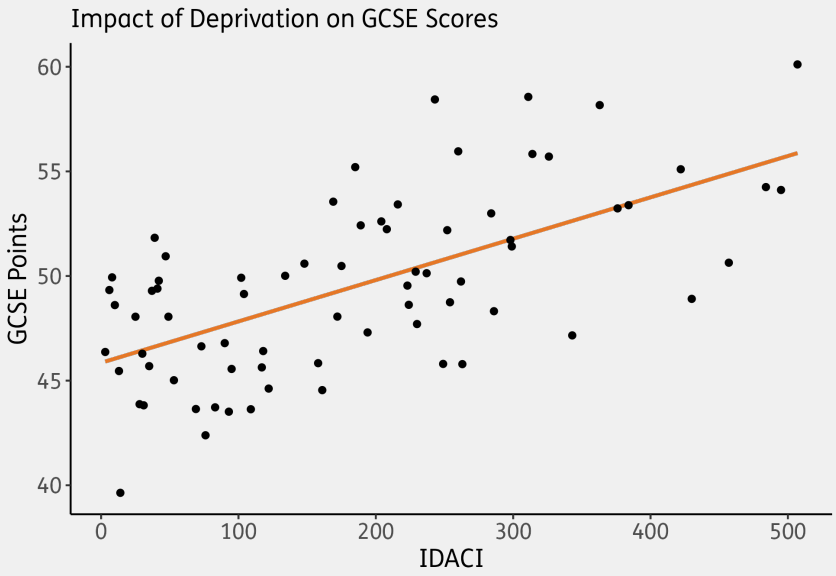
```
ggplot(london, aes(x = idaci, y = gcse)) +
  geom_smooth(method = lm, se=FALSE) +
  geom_point()
```



Improve the graph by:

1. Adding a regression line.
2. Adding up a relevant title, also possibly a subtitle.
3. Adding axes labels and making them readable.

```
ggplot(london, aes(x = idaci, y = gcse)) +
  geom_smooth(method = lm, se=FALSE, colour="orange") +
  geom_point() +
  theme_classic() +
  xlab('IDACI') +
  ylab('GCSE Points') +
  ggtitle("Impact of Deprivation on GCSE Scores") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14)) +
  theme(plot.title = element_text(size = 14))
```

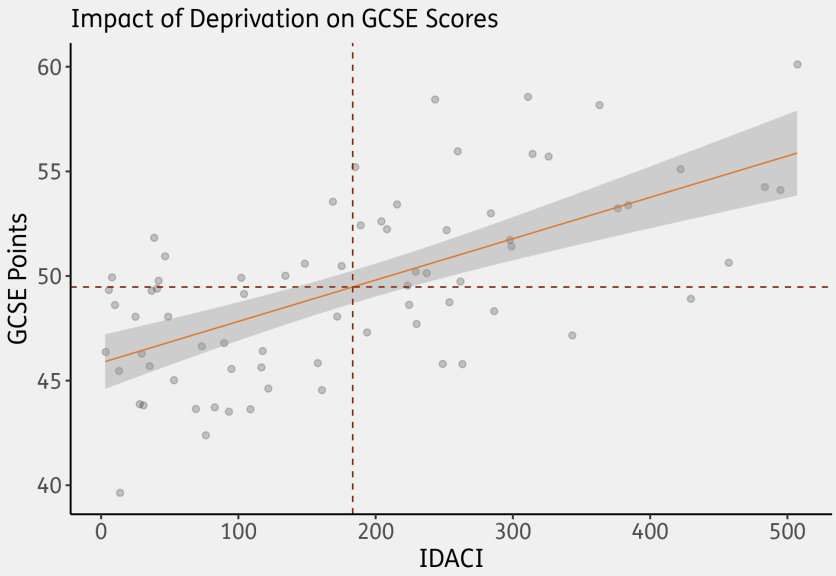


1.5 Visualisation 2.0

Now, draw a vertical and horizontal line corresponding to the mean of your variables using `geom_hline` and `geom_vline`. You can thus check if the regression line passes through the mean of X and Y. (see: https://www.rdocumentation.org/packages/ggplot2/versions/0.9.1/topics/geom_hline).

You can improve the scatterplot using a series of arguments (e.g., alpha) in the `geom_point()` function in ggplot. Try to improve the Aesthetics of the scatterplot playing with alpha, for instance. (see: https: //www.rdocumentation.org/packages/ggplot2/versions/3.4.0/topics/geom_point).

```
ggplot(london, aes(idaci, gcse)) +
  geom_point(position='jitter', alpha = 1/5) +
  xlab('IDACI') +
  ylab('GCSE Points') +
  ggtitle("Impact of Deprivation on GCSE Scores") +
  theme_classic() +
  geom_smooth(method = 'lm', se=T, colour = 'orange', lwd=0.4)+
  geom_hline(yintercept = mean(london$gcse, na.rm=TRUE), color='red', lty='dashed', lwd=0.4)+
  geom_vline(xintercept = mean(london$idaci,na.rm=TRUE), color='red', lty='dashed', lwd=0.4)+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14)) +
  theme(plot.title = element_text(size = 14))
```



1.6 Saving the Scatterplot

You can save a graph as .png, .JPG (even .pdf) that you can then import in a word document. Although there are many way to use your R output, saving a graph might be sometimes useful.

Use the function `ggsave()` to save your scatterplot. Again, there are tons of examples online, google it.

Hint: You first need to store the graph in an object.

Hint 2: The file will end up in your working directory

```
# First, create an object #
scatterplot <- ggplot(london, aes(idaci, gcse)) +
  geom_point(position='jitter', alpha = 1/5) +
  xlab('IDACI') +
  ylab('GCSE Points') +
  ggtitle("Impact of Deprivation on GCSE Scores") +
  theme_classic() +
  geom_smooth(method = 'lm', se=T, colour = 'orange', lwd=0.4)+
  geom_hline(yintercept = mean(london$gcse, na.rm=TRUE), color='red', lty='dashed', lwd=0.4)+
  geom_vline(xintercept = mean(london$idaci,na.rm=TRUE), color='red', lty='dashed', lwd=0.4)+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14)) +
  theme(plot.title = element_text(size = 14))
```

```
# Then save the file #
save_plot("scatterplot_idaci.png", scatterplot)
```

1.7 Regression Analysis (yes, finally)

Now we can finally run a linear regression with `gcse` as the outcome variable and `idaci` as the predictor using the `lm()` function. Store the results in an object called `model` and visualise the regression output using `summary()`.

```
# Store the results in an object called model #
model<-lm(gcse ~ idaci, london)
# Visualise the regression output using summary() #
summary(model)
##
## Call:
## lm(formula = gcse ~ idaci, data = london)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4921 -2.5321 -0.1722  2.7077  7.7815
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45.852958   0.657251  69.765  < 2e-16 ***
## idaci        0.019765   0.002894   6.831  2.4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.317 on 71 degrees of freedom
## Multiple R-squared:  0.3965, Adjusted R-squared:  0.388
## F-statistic: 46.66 on 1 and 71 DF,  p-value: 2.398e-09
```

You can also extract specific blocks of the output table. One way of doing it is to use the brackets [] after the `summary()` function. For example `summary()[8]`. Try to extract the block of Coefficients from the table, like this:

```
summary(model)[4]
## $coefficients
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 45.8529580 0.65725150 69.764707 3.767825e-67
## idaci        0.0197651 0.00289364  6.830532 2.397995e-09
```

1.8 Interpretation

Interpret the results, starting with model evaluation.

1. Is the p-value of the F-statistics statistically significant? We will be discussing this in our following lectures.
    - Yes, highly significant with p-value: 2.398e-09.
2. How much variation in the outcome variable does the model explain? What does this tell us about the model?
    - Income Deprivation Affecting Children explains 39.65% of the variation in GCSE scores in London constituencies. This is not bad for a single variable, but there are probably other factors, as well.
3. What's the value of the slope? What does it mean?
    - 0.019765. For every additional rank in the Income Deprivation Affecting Children Index, the GCSE score will rise by 0.019765, on average.
4. What's the value of the intercept? How do we interpret it? Is it statistically significant? What does it mean in practice?
    - The intercept (45.852958) is highly significant, and indicates that if the Income Deprivation Affecting Children Index was zero, the average GCSE score in London constituencies would be 45.86.
5. Interpret the results (in plain language) referring to the hypothesis you formulated above.
    - The higher the level of income deprivation affecting children, the lower the GCSE score in a London constituency, and therefore we verify our alternative hypothesis.

References

GOV.UK. (2013). *National Statistics: Income and tax by Parliamentary constituency.* available online at https://www.gov.uk/government/statistics/income-and-tax-by-parliamentary-constituency-2010-to-2011.

House of Commons Library. (n.d.). *Data Dashboard.* available online at https://commonslibrary.parliament.uk/type/data-dashboard/.

London Data Store. (2010). *London Parliamentary Constituency Profiles 2010.* available online at https://data.london.gov.uk/dataset/london-parliamentary-constituency-profiles.