

GY7702 Report

Author

23/12/2020

This document was created to meet the requirements of GY7702 R for Data Science at University of Leicester. It was designed and created in R Markdown, a markup language that allows users to create documents that can be formatted to embed code blocks, code outputs and hyperlinks. When the R Markdown file is compiled, the markup language is hidden and the document is displayed in plain text.

This content was created using R, Rstudio, RMarkdown and GitHub

The libraries used in this assignment were

```
library(tidyverse)
library(knitr)
#library(tinytex)
library(kableExtra)
library(gridExtra)
library(psych)
library(magrittr)
```

For further information regarding the source code, data and libraries used in this assignment, please see the projects GitHub page ([link](#)).

References

The Author would like to acknowledge that this document includes teaching materials from Dr Stefano De Sabbato for the module GY7702 R for Data Science. Dr Stefanos teaching materials can be found [here](#)

R for Data Science by Garrett Golemund and Hadley Wickham, O'Reilly Media, 2016. See online book [here](#)

Load in the data

```
# Extract
unzip("Data/GY7702_2020-21_Assignment_2--data_pack.zip", exdir = "Data")

# Load
OAC_2011 <-
  readr::read_csv(
    "Data/GY7702_2020-21_Assignment_2--data_pack/2011_OAC_Raw_kVariables.csv")

OAC_2011_meta <-
  readr::read_csv(
    "Data/GY7702_2020-21_Assignment_2--data_pack/OA11_LSOA11_MSOA11_LAD11_EW_LUv2.csv")
```

Option A

Joining the data sets

```
# Rename the column 'OA11CD' to OA so it matches the same column in OAC_2011
OAC_2011_meta <- OAC_2011_meta %>%
  dplyr::rename(
    OA = OA11CD
  )

# Join OAC_2011 with OAC_2011_meta, based on the OA column for each data frame
complete_OAC <-
  OAC_2011 %>%
  dplyr::left_join(., OAC_2011_meta)
```

Subsetting

- Selecting assigned LAD
- Selecting the appropriate variables
 - k004 Persons aged 45 to 64
 - k009 Persons aged over 16 who are single
 - k010 Persons aged over 16 who are married or in a registered same-sex civil partnership
 - k027 Households who live in a detached house or bungalow
 - k031 Households who own or have shared ownership of property
 - k041 Households with two or more cars or vans
 - k046 Employed persons aged between 16 and 74 who work part-time

```
# Subset data based off LAD11NM = Wellingborough
wellingborough <- complete_OAC %>%
  dplyr::filter(LAD11NM == "Wellingborough")

# Select the appropriate variables, based on their 'VariableCode'
wellingborough_variables <- wellingborough %>%
  dplyr::select(k004, k009, k010, k027, k031, k041, k046)
```

Question A.1

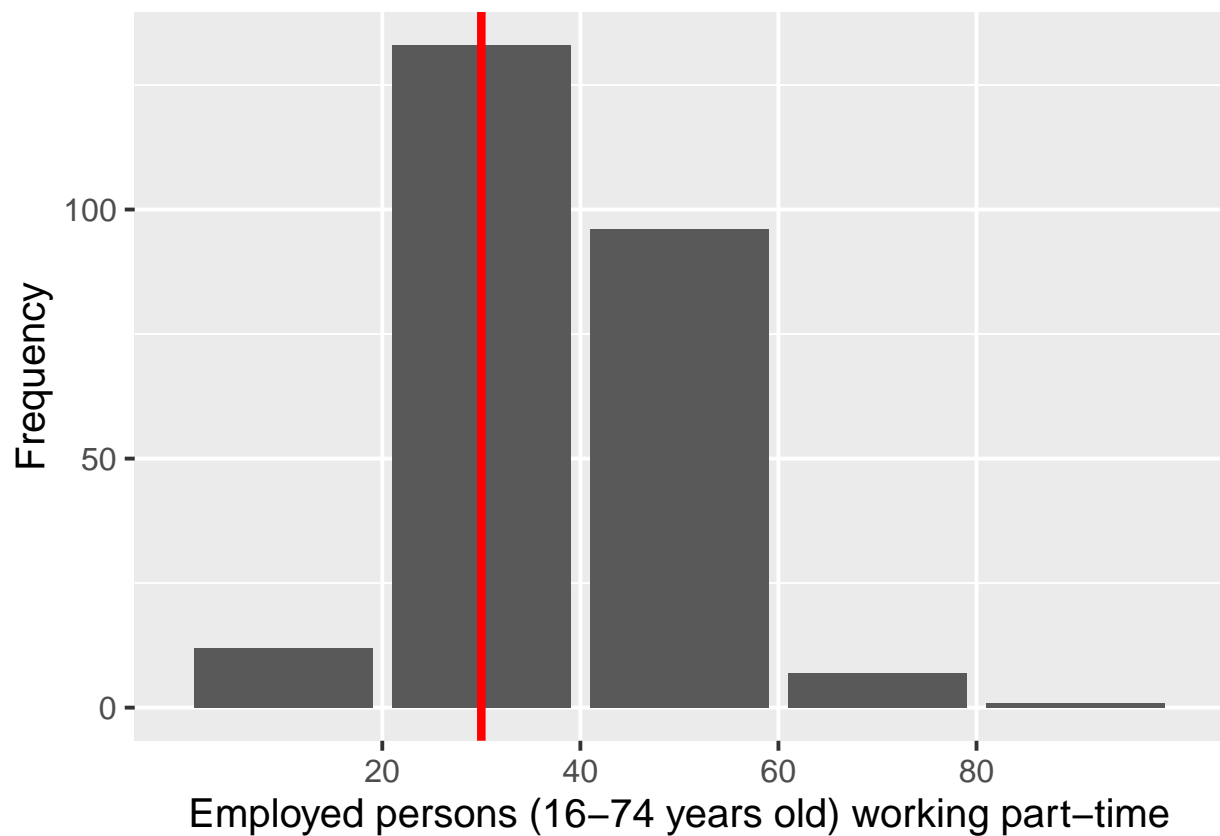
Word count (excluding titles = 497)

Exploring the distribution of Employed persons aged between 16 and 74 who work part-time

The distribution of part-time employed persons (PTEP) throughout 249 output areas (OA's) in the Local Area District (LAD) of Wellingborough was investigated.

Basic histogram for visual interpretation

```
k046_hist <- wellingborough %>%  
  ggplot2::ggplot(aes(  
    x = k046)) +  
  ggplot2::geom_bar() +  
  scale_x_binned(  
    n.breaks = 5) +  
  xlab("Employed persons (16-74 years old) working part-time") +  
  ylab("Frequency") +  
  geom_vline(aes(  
    xintercept = mean(k046),  
    colour = "red",  
    lwd = 1.5)) +  
  theme_gray(  
    base_size = 15)  
  
print(k046_hist)
```

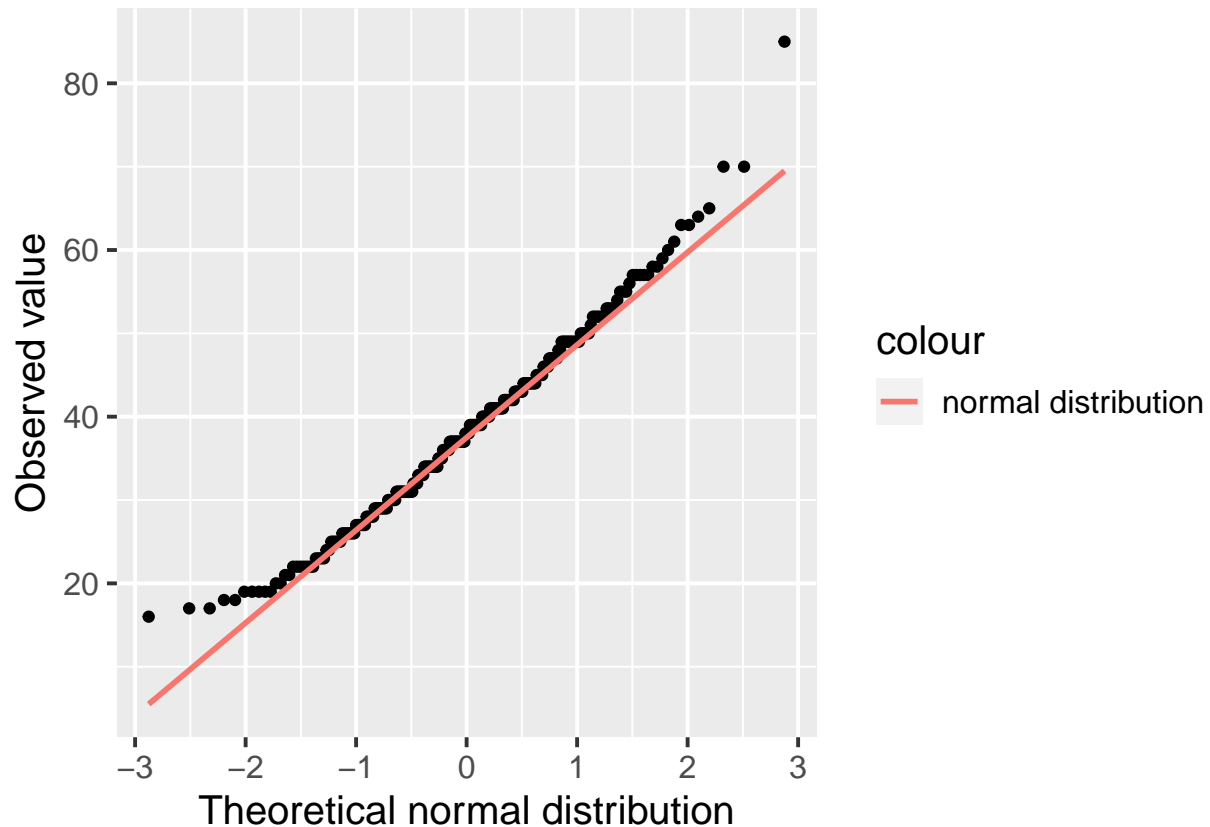


An initial assessment suggests that the distribution of PTEPs throughout Wellingborough is normal. The majority of OA's in Welingborough have 35-45 PTEPs with some having as little as 20 and at least one having >80 PTEPs. A red line also demonstrates the average number of PTEPs per OA (38)

Plotting against normal distribution

The distribution can be further assessed using `stat_qq`. This demonstrates where values may lie in the observed data that are affecting the normal distribution.

```
wellingborough %>%  
  ggplot2::ggplot(aes(  
    sample = k046))+  
  ggplot2::stat_qq()+  
  ggplot2::stat_qq_line(  
    aes(  
      colour = "normal distribution"),  
    show.legend = TRUE,  
    lwd = 1)+  
  xlab("Theoretical normal distribution")+  
  ylab("Observed value")+  
  theme_gray(  
    base_size = 15)
```



There are an elevated number of PTEPs at the tails of the distribution. That is, a considerable number of OA's have a minimum and maximum number of PTEPs.

Shapiro-Wilk test

Using a Shapiro-Wilk test we can investigate further if this a normal distribution.

```
k046 <- wellingborough %>%
  dplyr::pull(k046)%>%
  stats::shapiro.test()
print(k046)
```

```
##
## Shapiro-Wilk normality test
##
## data:  .
## W = 0.98179, p-value = 0.002799
```

In the Shapiro-Wilk test, a null hypothesis that the data is normally distributed is set. This is rejected if the p-value < 0.01. That is, there is a < 1% chance of the data being normally distributed. In this case $p < 0.01$, so the data is not normally distributed.

Exploring skewness and kurtosis

The overall shape of the data can be assessed further using *skewness* and *kurtosis*

```
wellingborough_skew <-
  wellingborough %>%
    dplyr::select(k046)%>%
    pastecs::stat.desc(basic = FALSE,
                      desc = FALSE,
                      norm = TRUE)
```

	k046
skewness	0.4888972
skew.2SE	1.5841824
kurtosis	0.4670775
kurt.2SE	0.7596877
normtest.W	0.9817935
normtest.p	0.0027990

A positive skew means the data leans left. The standard error for the skew `skew.2SE` lies outside of the range $-1 + 1$, meaning the result is significant. A positive kurtosis also means the data has heavy tails. However, this result is not significant as `kurt.2SE` lies within $-1 + 1$

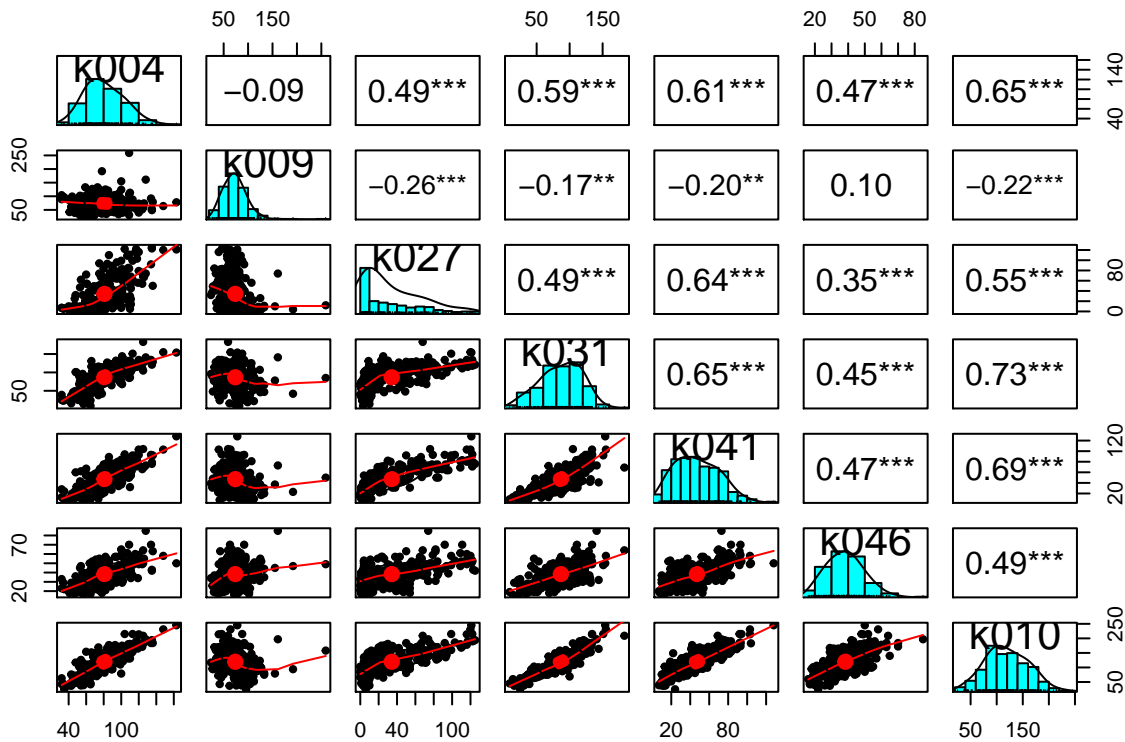
Multi-variate analysis

We have just carried out a uni-variate analysis. Relationships between variables can be assessed through multivariate analysis. The variables used in this study are split between collecting data on *persons* and *households*. We can use a `pairs.panel` to compare these sets of data and discover correlations. The data here is not normally distributed (except k010), hence we will use `kendall_tau` which assumes a non-normal distribution and ties.

```
wellingborough_norm <-
  wellingborough_variables %>%
  dplyr::select(k004, k009, k027, k031, k041, k046, k010) %>%
  pastecs::stat.desc(norm = TRUE)
```

	k004	k009	k027	k031	k041	k046	k010
normtest.W	0.9833013	0.8898289	0.858993	0.9843597	0.9747363	0.9817935	0.9913952
normtest.p	0.0050945	0.0000000	0.000000	0.0078237	0.0002059	0.0027990	0.1521888

```
wellingborough_variables %>%
  select(k004, k009, k027, k031, k041, k046, k010)%>%
  pairs.panels(method = "kendall",
              stars = TRUE)
```



k010 has a strong correlation with k031 and k041. This is understandable. People who have shared ownership of a property will usually (but not always) be in a relationship and may also own a car. k009 does not positively correlate with any other variable, possibly for the opposite reasons.

Normalising data

The amount of people in a relationship (k010) and the amount of households with shared ownership (k031) is highly correlated (0.73). However, k031 gives information on *households* while k010 gives information on *persons*. To aid comparison of the data sets, the z-scores can be taken.

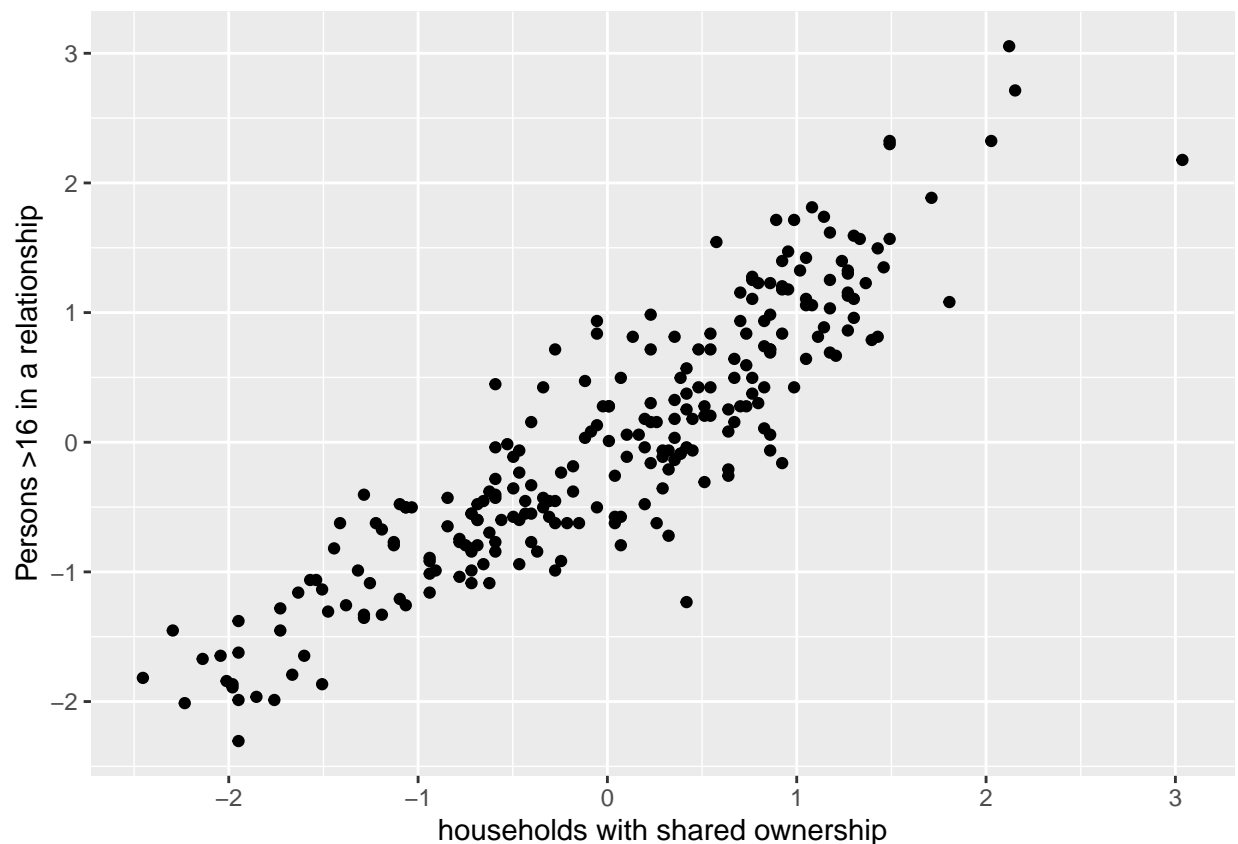
```
wellingborough_variables <-
  wellingborough_variables %>%
  mutate(z_k031 = scale(k031)) %>%
  mutate(z_k010 = scale(k010))
```

Note, though that z scores conserve the distribution of the data sets. This means that a Shapiro-Wilks test will still fail for k031

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  .  
## W = 0.98436, p-value = 0.007824
```

This assessment should not usually be carried out on non-normally distributed data. However, for the purpose of this study the new plot of z-scores offers more exploratory analysis. Values on either axis that are close to 0 are close to the mean. OA's that have the average number of people >16 in a relationship, also have the average number of households with shared ownership.

```
ggplot2::ggplot(  
  wellingborough_variables,  
  aes(z_k031,  
      z_k010)  
)+  
ggplot2::geom_point()+  
xlab("households with shared ownership")+  
ylab("Persons >16 in a relationship")
```



Question A.2

Word count (excluding titles = 480) ## Multiple linear regression

Households with two or more cars or vans (K041) and persons >16 who are married or in a same-sex civil partnership (K010) has a strong correlation with shared property ownership. Due to that, these variables will be selected to predict shared property ownership (k031).

```
wellingborough_variables %$%
stats::lm(k031 ~ k010 + k041) ->
  shared_prop_model

summary(shared_prop_model)
```

```
##
## Call:
## stats::lm(formula = k031 ~ k010 + k041)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.247  -8.094   0.021   8.545  46.628
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.51853    2.81821   1.958  0.0513 .
## k010         0.61210    0.04342  14.096 <2e-16 ***
## k041         0.17026    0.07740   2.200  0.0288 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.67 on 246 degrees of freedom
## Multiple R-squared:  0.8154, Adjusted R-squared:  0.8139
## F-statistic: 543.3 on 2 and 246 DF,  p-value: < 2.2e-16
```

With a $p < 0.01$, the performance of the model is good. A high F-statistic [$f(2, 246) = 543.3$] demonstrates this model is more effective than if the mean for the outcome variable was used in all cases of the independent variables. An adjusted R^2 of 0.81 demonstrates that k010 and k041 explain 81% of the variance in k031.

For 1 increase in k010, k031 increases by 0.61. For 1 increase in k010, k031 increases by 0.17. This suggests that k010 has a larger impact on k031. The 3-dimensional model plane intersects the y-axis at 5.51.

P-values show good results for k010 ($P < 0.01$), however confidence is exceeded for k041 and the y-intercept which suggests the model is faltering.

Standardise coefficients

`lm.beta` standardises the coefficients to aid comparison. 1 standard deviation (SD) increment in k010, leads to 0.79 more shared properties and 1 SD increment in (k041) leads to 0.12 more shared properties. This further suggests that k010 is explaining more of the variance than k041.

```
shared_prop_model %>%
  lm.beta::lm.beta()
```



```
##
## Call:
## stats::lm(formula = k031 ~ k010 + k041)
##
## Standardized Coefficients:
## (Intercept)      k010      k041
##  0.0000000    0.7929166    0.1237340
```

Determine confidence intervals

`stats::confint()` quantifies a range at which the true coefficients would occupy. If the range is small, the model is robust.

```
shared_prop_model %>%
  stats::confint()
```

```
##              2.5 %      97.5 %
## (Intercept) -0.03236216 11.0694228
## k010         0.52657324  0.6976349
## k041         0.01780218  0.3227096
```

These ranges are relatively large. The model cannot be confident on the true value of the coefficients. The range is significantly smaller for k041 though.

Testing for outliers

We can test for outliers by assessing the standard residuals and the cook's distance. A threshold for a cook's distance > 1 will be set.

```
wellingborough_variables %>%
  mutate(
    # Take the standardised residuals, from the shared prop model
    model_stdres = shared_prop_model %>%
      stats::rstandard(),
    # Take the cook distance from the shared prop model
    model_cook_dist = shared_prop_model %>%
      stats::cooks.distance()
  ) ->

# Put those into a new table called shared_prop_output
shared_prop_output

# From shared_prop_output, select the outcome variable with the associate
# residuals and cook distance
shared_prop_output %>%
  dplyr::select(k031, model_stdres, model_cook_dist) %>%
  # Filter by stdres more than 2.58 and cook distance more than 1
  dplyr::filter(abs(model_stdres) > 2.58 | model_cook_dist > 1)
```

```
## # A tibble: 2 x 3
##   k031 model_stdres model_cook_dist
```

```
##      <dbl>      <dbl>      <dbl>
## 1    183      2.82      0.106
## 2    100      3.43      0.0551
```

:filter(abs(model_stdres) >2.58 is specifying any values above the 99% range of the data set. 2 values exceed this threshold. These values are outliers as they lie on the .5% upper and .5% lower limits of the data set.

Testing assumptions

We wil now step through the assumptions made during the building of this model

1. A linear relationship

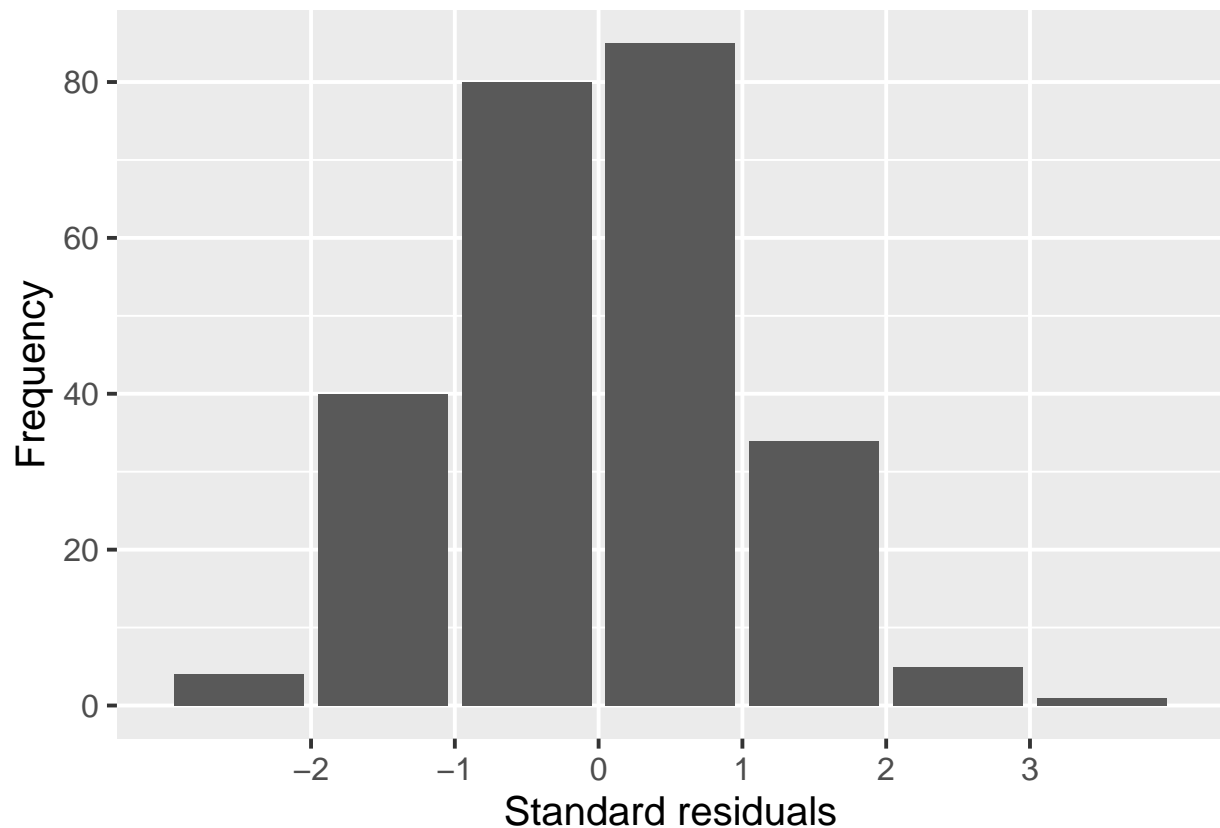
This was previously demonstrated in the univariate analysis

2. Normal Distribution of standard residuals

A $p > 0.01$ states that the distribution of the standard residual is not significant, and therefore normal.

```
##
## Shapiro-Wilk normality test
##
## data:  model_stdres
## W = 0.99494, p-value = 0.5826
```

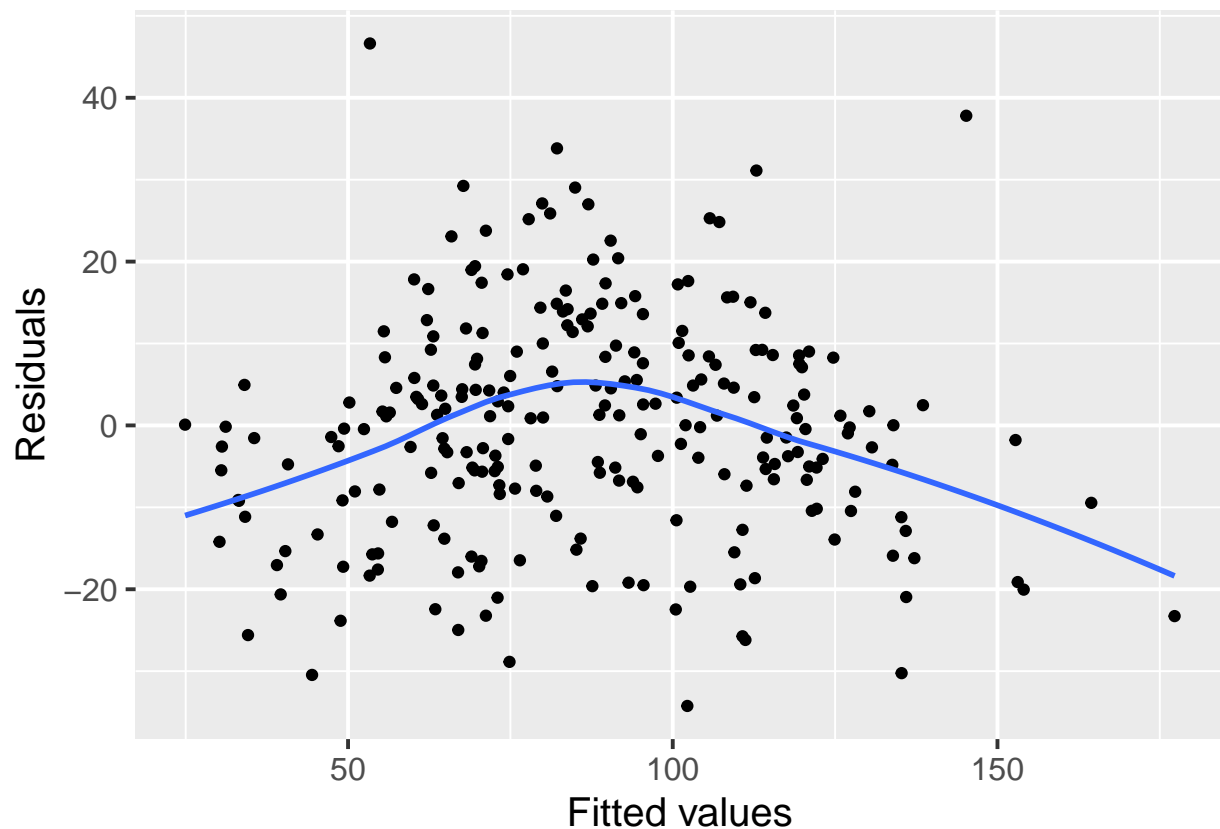
This can be visually assessed with a histogram



3. Standard residuals are homoscedastic and independent

The homoscedasticity can be assessed by plotting the fitted values with the residuals

```
ggplot2::ggplot(shared_prop_model,
  aes(
    x = fitted.values(shared_prop_model),
    y = residuals(shared_prop_model)))+
ggplot2::geom_point()+
ggplot2::geom_smooth(se=FALSE)+
xlab("Fitted values")+
ylab("Residuals")+
theme_gray(
  base_size = 15)
```



A “cloud” of residuals shows that residuals are not correlated with the model. Residuals are similar whether the outcome variable is high or low. However, a curved line demonstrates that these residuals can vary throughout the model.

A Breusch-Pagan test further demonstrates that the residuals are homoscedastic as the result is not significant.

```
shared_prop_model %>%
  lmtest::bptest()
```

```
##
## studentized Breusch-Pagan test
```

```
##
## data:  .
## BP = 0.41851, df = 2, p-value = 0.8112
```

Independence of residuals can be tested with a Durbin-Watson test. The residuals are independent.

```
shared_prop_model %>%
  lmtest::dwtest()
```

```
##
## Durbin-Watson test
##
## data:  .
## DW = 1.744, p-value = 0.01959
## alternative hypothesis: true autocorrelation is greater than 0
```

4. Relationship between independent variables is independent

Finally, we can test the variance inflation factor (ViF). This tests if the variables are independent from one another.

```
shared_prop_model %>%
  car::vif()
```

```
##      k010      k041
## 4.216944 4.216944
```

The results for this test are relatively high. This means that the variables ‘move’ together. A change in k041 may also affect k010. These variables are not independent from one another.