ARI1101 Group Assignment

Kian Parnis and Evangeline Azzopardi

22nd January 2021

Task 1: Understanding the data

The variables for data-rental.csv are classified as follows:

Continuous Quantitative variables:

- house_price
- bedrooms
- surface

Discrete Qualitative variables:

- rental_agency
- city

Statistical Analysis Methods

Pre-Cleaning Analysis Before performing any analysis the distribution of the quantitative variables was calculated to determine the skewness of the data and its effect on the results obtained during the analysis.

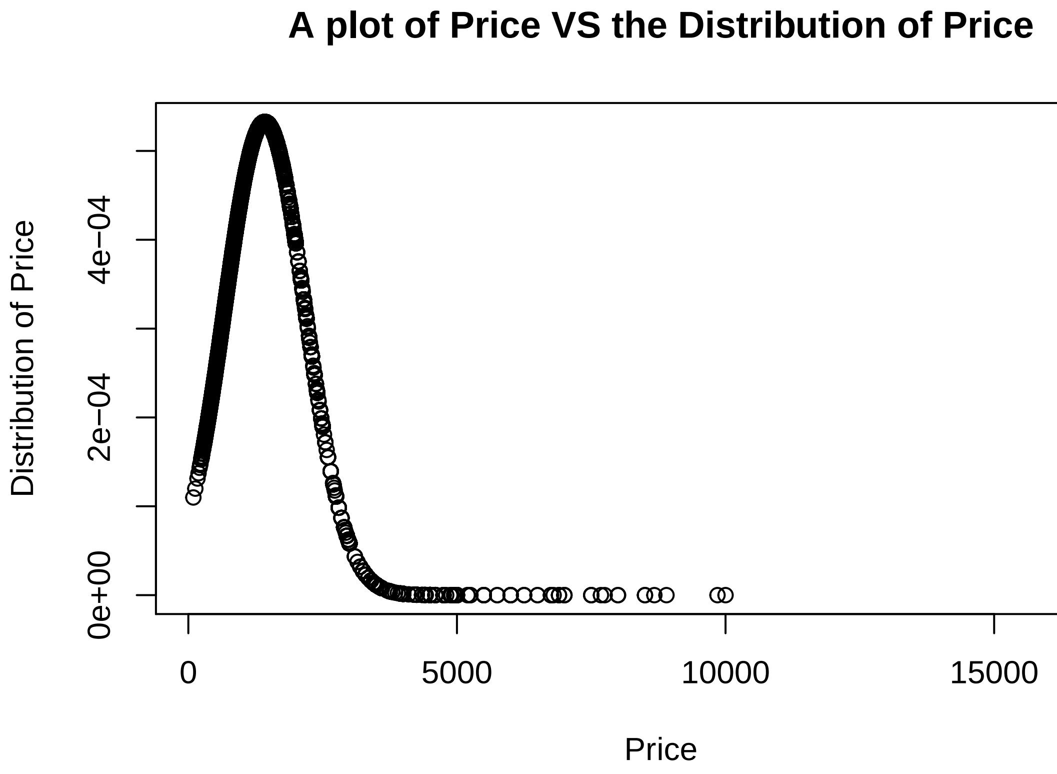
```
preCleanedRent <- read.csv(file = 'data-rental.csv')

priceMean<-mean(preCleanedRent$house_price, na.rm = TRUE)#1423.65
priceMean
```

```
## [1] 1423.65
```

```
priceSD<-sd(preCleanedRent$house_price, na.rm = TRUE)#748.99
priceSD
```

```
## [1] 748.99
```

```
normDistPrice<-dnorm(preCleanedRent$house_price, 1423.65, 748.99)
plot(preCleanedRent$house_price, normDistPrice,
     main = "A plot of Price VS the Distribution of Price",
     xlab = "Price",
     ylab = "Distribution of Price")
```



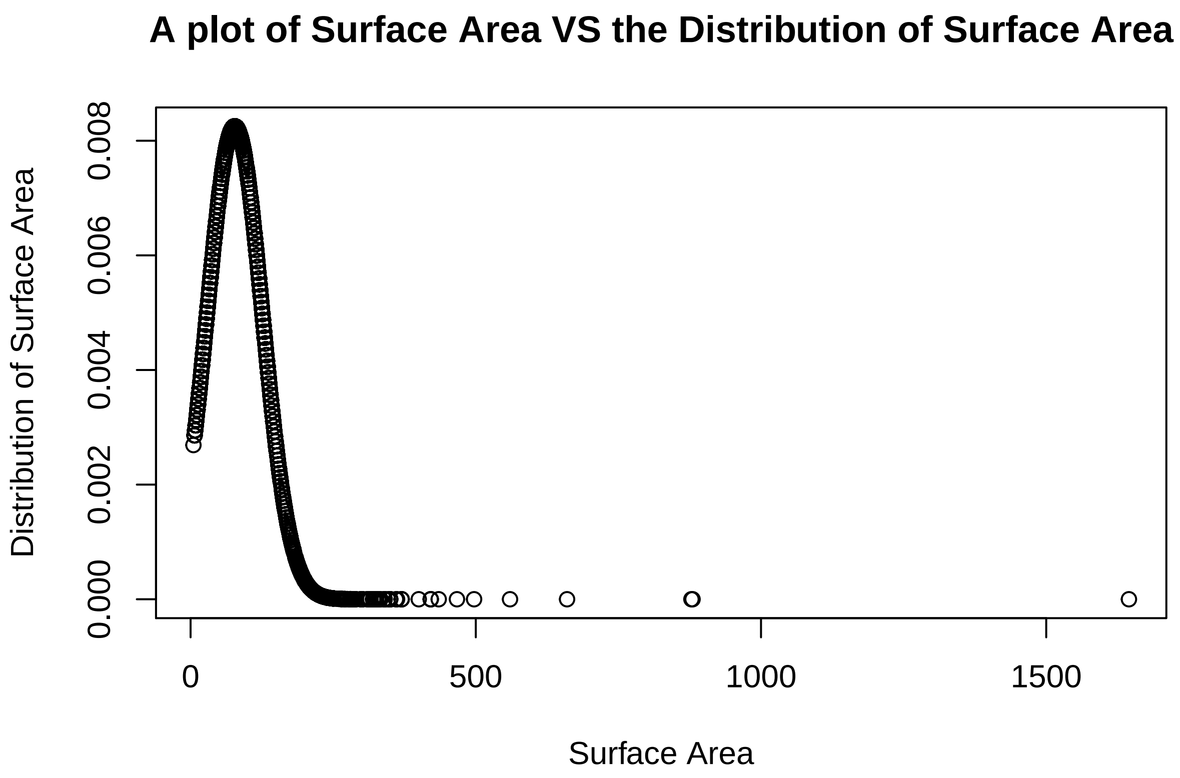
```
areaMean<-mean(preCleanedRent$surface, na.rm = TRUE)#77.38945
areaMean
```

```
## [1] 77.38945
```

```
areaSD<-sd(preCleanedRent$surface, na.rm = TRUE)#48.34938
areaSD
```

```
## [1] 48.34938
```

```
normDistSArea<-dnorm(preCleanedRent$surface, 77.38945, 48.34938)
plot(preCleanedRent$surface, normDistSArea,
     main = "A plot of Surface Area VS the Distribution of Surface Area",
     xlab = "Surface Area",
     ylab = "Distribution of Surface Area")
```



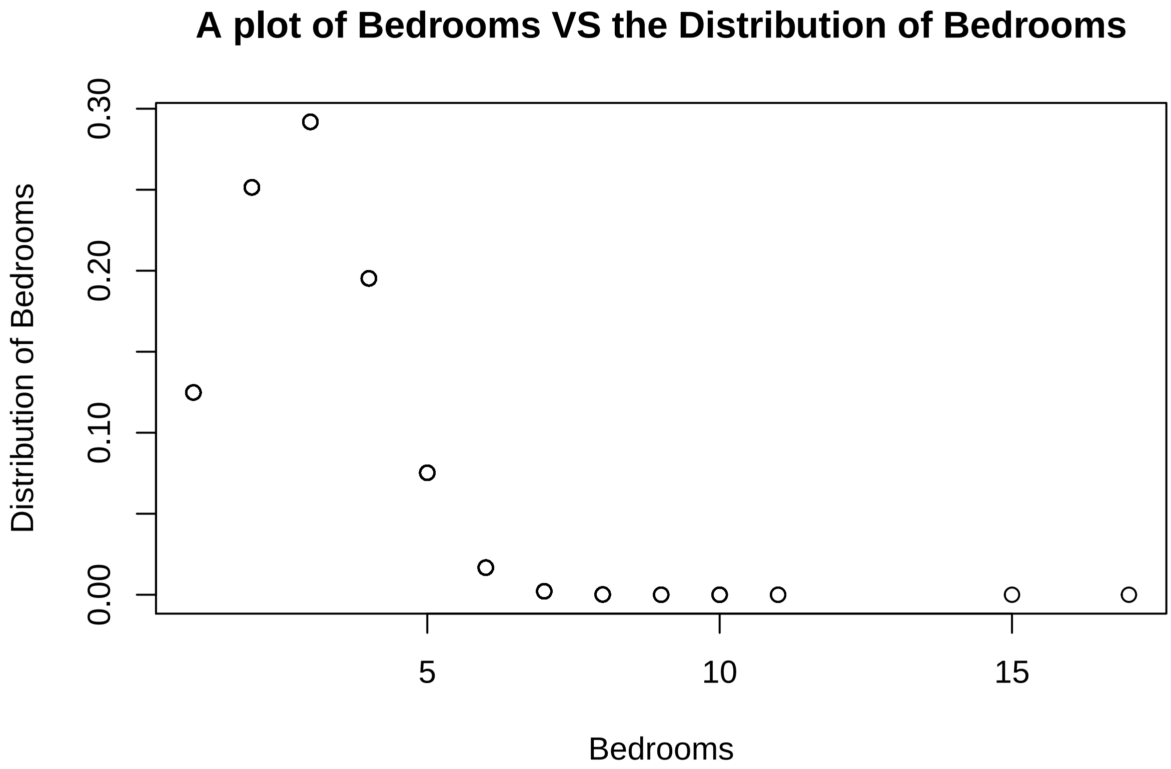
```r
roomsMean<-mean(preCleanedRent$bedrooms, na.rm = TRUE)#2.770552
roomsMean
```

```
## [1] 2.770552
```

```r
roomsSD<-sd(preCleanedRent$bedrooms, na.rm = TRUE)#1.347051
roomsSD
```

```
## [1] 1.347051
```

```r
normDistBRooms<-dnorm(preCleanedRent$bedrooms, 2.770552, 1.347051)
plot(preCleanedRent$bedrooms, normDistBRooms,
     main = "A plot of Bedrooms VS the Distribution of Bedrooms",
     xlab = "Bedrooms",
     ylab = "Distribution of Bedrooms")
```



As can be seen in the plots above, the quantitative data is positively skewed with the majority of the values clustered to the left with a longer right tail. Due to the skewness of the data, for certain parts of Task 3, a sample will be used to minimise the positive skewness of the data.

Correlation The variables which could be considered in measuring the correlations are as follows:

- house_price VS city.
- house_price VS bedrooms VS surface.
- house_price VS surface.
- city VS bedrooms VS surface.
- bedrooms VS surface.
- rental_agency VS city.

The variables chosen for our analysis of the correlation are:

- house_price VS surface.
- house_price VS bedrooms.
- surface VS bedrooms.

Regression Regression, through the creation of a linear model which further measures the relationship between the variables, will be applied to make predictions for said variables.

Sampling Methods The sampling method chosen for our analysis is systematic sampling, as it eliminates any bias when creating the sample. This will applied by using the sample_n() function in the dplyr library.

The population will be divided by 3, after cleaning, to produce a sample as required by Task 2.3 and will be used to find the sample means and to create a heatmap to show the relationship between the location of a property and its price.

The distribution of the quantitative variables in the sample population for Task 3 was re-calculated and resulted in the data being less positively skewed, however the skewness was not eliminated entirely. Hence, it still influenced the results obtained in our analysis.

Task 2: Cleaning the data

Duplicated rows

The library **tidyverse** and **dplyr** were primarily used for cleaning the data, they help to transform the data set with ease and **%>% filter()** was commonly used throughout the cleaning process.

The first part of cleaning involved identify and inspecting any identical rows in the data set, this was achieved with **duplicated()** which displays if any duplicates are present and which rows they occur in. Afterwards **unique()** was used to add every unique row to the new data set, since data was removed the row numbers were reset and **duplicated()** is used again to validate succession.

```
Rent <- read.csv(file = 'data-rental.csv')

which(duplicated(Rent))
```

```
##   [1]   33   65   66   80   98  161  162  193  258  290  314  321
##  [13]  353  354  386  417  418  449  481  482  513  514  545  546
##  [25]  554  577  578  602  609  610  641  642  673  674  690  705
##  [37]  706  737  738  769  770  801  802  833  834  865  866  872
##  [49]  897  898  929  930  961  962  993  994 1025 1026 1039 1057
##  [61] 1058 1089 1090 1121 1122 1153 1154 1185 1186 1195 1217 1218
##  [73] 1231 1249 1250 1281 1282 1303 1313 1314 1322 1345 1346 1358
##  [85] 1377 1378 1380 1381 1409 1410 1441 1442 1473 1474 1505 1506
##  [97] 1513 1525 1537 1538 1569 1570 1601 1602 1633 1634 1636 1665
## [109] 1666 1697 1698 1703 1729 1730 1761 1762 1793 1794 1825 1826
## [121] 1857 1858 1889 1890 1921 1922 1945 1946 1953 1954 1985 1986
## [133] 1989 2017 2018 2049 2050 2081 2082 2103 2113 2114 2145 2146
## [145] 2177 2178 2209 2210 2213 2241 2242 2273 2274 2305 2306 2332
## [157] 2337 2338 2369 2370 2401 2402 2433 2434 2465 2466 2468 2481
## [169] 2488 2489 2497 2498 2529 2530 2536 2561 2562 2579 2585 2586
## [181] 2591 2593 2594 2605 2625 2626 2657 2658 2684 2689 2690 2721
## [193] 2722 2753 2754 2772 2773 2785 2786 2817 2818 2849 2850 2881
## [205] 2882 2913 2914 2945 2946 2977 2978 2982 3009 3010 3041 3042
## [217] 3073 3074 3105 3106 3137 3138 3145 3150 3169 3170 3201 3202
## [229] 3233 3234 3265 3266 3291 3297 3298 3306 3308 3309 3310 3311
## [241] 3312 3313 3318 3329 3330 3361 3362 3393 3394 3425 3426 3439
## [253] 3457 3458 3489 3490 3515 3521 3522 3553 3554 3585 3586 3617
## [265] 3618 3649 3650 3681 3682 3713 3714 3745 3746 3777 3778 3788
## [277] 3809 3810 3841 3842 3873 3874 3905 3906 3937 3938 3947 3955
## [289] 3965 3969 3970 3996 4001 4002 4033 4034 4055 4065 4066 4072
## [301] 4079 4080 4083 4086 4090 4097 4098 4129 4130 4161 4162 4193
## [313] 4194 4225 4226 4232 4257 4258 4265 4271 4289 4290 4296 4321
## [325] 4322 4353 4354 4385 4386 4417 4418 4449 4450 4481 4482 4513
## [337] 4514 4533 4534 4535 4536 4537 4545 4546 4547 4577 4578 4609
## [349] 4610 4637 4641 4642 4647 4673 4674 4705 4706 4737 4738 4753
## [361]  4754  4769  4770  4792  4801  4802  4833  4834  4865  4866  4897  4898
## [373]  4927  4929  4930  4936  4952  4961  4962  4985  4993  4994  5025  5026
## [385]  5052  5057  5058  5087  5089  5090  5121  5122  5153  5154  5185  5186
## [397]  5213  5217  5218  5221  5228  5249  5250  5281  5282  5313  5314  5343
## [409]  5345  5346  5356  5377  5378  5394  5399  5407  5409  5410  5415  5441
## [421]  5442  5473  5474  5496  5505  5506  5537  5538  5569  5570  5581  5583
## [433]  5584  5585  5586  5587  5593  5601  5602  5605  5633  5634  5653  5665
## [445]  5666  5680  5697  5698  5713  5729  5730  5735  5757  5761  5762  5793
## [457]  5794  5825  5826  5848  5857  5858  5889  5890  5904  5921  5922  5925
## [469]  5935  5953  5954  5985  5986  6017  6018  6049  6050  6059  6081  6082
## [481]  6088  6113  6114  6125  6145  6146  6177  6178  6194  6209  6210  6241
## [493]  6242  6268  6273  6274  6305  6306  6330  6337  6338  6346  6369  6370
## [505]  6378  6401  6402  6406  6422  6433  6434  6462  6465  6466  6497  6498
## [517]  6520  6529  6530  6532  6540  6561  6562  6578  6593  6594  6605  6625
## [529]  6626  6638  6647  6657  6658  6661  6689  6690  6707  6720  6721  6722
## [541]  6753  6754  6759  6785  6786  6788  6805  6817  6818  6849  6850  6855
## [553]  6856  6881  6882  6913  6914  6945  6946  6977  6978  6984  7009  7010
## [565]  7041  7042  7056  7057  7073  7074  7105  7106  7115  7137  7138  7149
## [577]  7169  7170  7182  7201  7202  7233  7234  7265  7266  7281  7285  7297
## [589]  7298  7323  7329  7330  7345  7361  7362  7393  7394  7425  7426  7433
## [601]  7457  7458  7489  7490  7507  7521  7522  7553  7554  7564  7583  7585
## [613]  7586  7597  7612  7617  7618  7625  7649  7650  7680  7681  7682  7707
## [625]  7709  7710  7711  7712  7713  7714  7715  7716  7717  7718  7719  7720
## [637]  7721  7722  7723  7724  7725  7726  7733  7745  7746  7775  7777  7778
## [649]  7784  7787  7790  7794  7809  7810  7831  7841  7842  7852  7862  7873
## [661]  7874  7905  7906  7937  7938  7940  7969  7970  8001  8002  8028  8033
## [673]  8034  8050  8053  8059  8065  8066  8088  8097  8098  8129  8130  8161
## [685]  8162  8185  8192  8193  8194  8204  8218  8225  8226  8235  8257  8258
## [697]  8285  8289  8290  8309  8321  8322  8325  8330  8353  8354  8385  8386
## [709]  8410  8417  8418  8423  8449  8450  8467  8478  8481  8482  8504  8513
## [721]  8514  8523  8541  8545  8546  8547  8548  8550  8553  8577  8578  8596
## [733]  8609  8610  8630  8641  8642  8663  8664  8665  8666  8673  8674  8678
## [745]  8705  8706  8711  8720  8733  8735  8737  8738  8769  8770  8771  8795
## [757]  8800  8801  8802  8833  8834  8865  8866  8897  8898  8929  8930  8961
## [769]  8962  8971  8993  8994  9025  9026  9057  9058  9089  9090  9099  9121
## [781]  9122  9153  9154  9185  9186  9194  9195  9205  9217  9218  9244  9249
## [793]  9250  9281  9282  9313  9314  9319  9328  9345  9346  9352  9353  9360
## [805]  9373  9377  9378  9382  9409  9410  9431  9441  9442  9473  9474  9505
## [817]  9506  9537  9538  9568  9569  9570  9600  9601  9602  9621  9633  9634
## [829]  9645  9660  9665  9666  9697  9698  9709  9717  9729  9730  9761  9762
## [841]  9793  9794  9810  9825  9826  9841  9856  9857  9858  9864  9876  9889
## [853]  9890  9921  9922  9927  9928  9953  9954  9976  9985  9986  9988 10017
## [865] 10018 10028 10030 10031 10032 10033 10034 10035 10036 10048 10049 10050
## [877] 10055 10064 10067 10081 10082 10086 10087 10101 10113 10114 10145 10146
## [889] 10166 10168 10177 10178 10198 10202 10209 10210 10237 10238 10240 10241
## [901] 10242 10244 10249 10273 10274 10305 10306 10323 10327 10336 10337 10338
## [913] 10356 10369 10370 10389 10401 10402 10433 10434 10465 10466 10497 10498
## [925] 10529 10530 10561 10562 10571 10572 10578 10583 10593 10594 10625 10626
## [937] 10630
```

```
Rent <- unique(Rent)
row.names(Rent) <- NULL #Reset row numbers
which(duplicated(Rent))
```

```
## integer(0)
```

Numerical Check

Afterwards **is.numeric** was used to go through all the rows.

```
Rent_num <- unlist(lapply(Rent, is.numeric))
Rent_num
```

```
##   house_price rental_agency          city      bedrooms       surface 
##          TRUE         FALSE         FALSE          TRUE          TRUE
```

Missing Values

N/A’s were taken into consideration and **is.na()** was used to locate any missing values in the data set, only house_price was found to contain missing values so imputation was required.

```
x <- which(is.na(Rent))
print(Rent[x, ])
```

```
##      house_price                             rental_agency             city
## 86            NA                          BED’R Apartments        Groningen
## 484           NA                          BED’R Apartments        Groningen
## 589           NA                          BED’R Apartments        Groningen
## 638           NA                          BED’R Apartments        Groningen
## 690           NA                          BED’R Apartments        Groningen
## 1203          NA                          Fransen Vastgoed      Vlaardingen
## 2182          NA                Havos Vastgoedbelegging bv        Uithuizen
## 2429          NA                 Comfortable Staff Housing        Westbroek
## 2728          NA     Makelaarskantoor Paul Schreinemachers            Venlo
## 4360          NA                    vastgoedPROmakelaar.nl            Weert
## 4659          NA                               Expat Group          Tilburg
## 5821          NA                Staffhousing Services B.V. Pernis Rotterdam
## 5833          NA                Staffhousing Services B.V.            Hoorn
## 5879          NA                          Fransen Vastgoed       Sliedrecht
## 5881          NA                          Fransen Vastgoed Pernis Rotterdam
## 6202          NA                Staffhousing Services B.V.         Lelystad
## 7616          NA                               NL en Wonen       IJzendijke
## 7617          NA                               NL en Wonen           Bussum
## 7943          NA              Huijers Vastgoed Makelaardij            Weert
## 8037          NA                        Grand Prix Rentals     Valkenswaard
## 8082          NA                              AB&P Vaassen         Roermond
## 8083          NA                              AB&P Vaassen         Roermond
## 8485          NA                          Fransen Vastgoed        Maassluis
## 8520          NA Gerro de Boer Makelaardij & Taxaties o.g.        Purmerend
## 8837          NA                              AB&P Vaassen         Roermond
## 8852          NA                             Stam Vastgoed       Loosdrecht
## 8919          NA                              AB&P Vaassen         Roermond
## 9167          NA                        123Wonen Flevoland    Biddinghuizen
## 9212          NA                     Short Stay Group B.V.        Amsterdam
## 9215          NA                     Short Stay Group B.V.        Amsterdam
## 9216          NA                     Short Stay Group B.V.        Amsterdam
##      bedrooms surface
## 86          3      50
## 484         2      55
## 589         1      30
## 638         2      30
## 690         2      50
## 1203        6     142
## 2182       10     130
## 2429        8     250
## 2728        2      28
## 4360        4      85
## 4659        6     150
## 5821        5     100
## 5833        5     103
## 5879        5      85
## 5881        5     136
## 6202        5     135
## 7616       15     360
## 7617        5     144
## 7943        4     135
## 8037        9     878
## 8082       10     340
## 8083       10     262
## 8485        4      75
## 8520        6     290
## 8837        4      90
## 8852        1      50
## 8919        6     118
## 9167        2      56
## 9212        3      83
## 9215        2      45
## 9216        2      70
```

When dealing with missing data a general format was followed in order for these values to be filled in. This format's intent was to look at the data set and impute data based on similar records in the set.

Data was organized based off the rental agency column for better understanding on what decisions to take in the process and this is split into three categories:

- *All Rental N/A*, which is when agency is filtered and all the agency's prices aren't present.
- *!All Rental N/A*, which has agencies present with both missing and present prices.
- *Appears once*, Agencies that only appear once with one NA value.

The format starts off with filtering out N/A values, per agency based on city, bedrooms and surface area. In the likelihood that no other data is present with the three variables another filter is done but with city being excluded and again if no other data is found then only the bedrooms are filtered.

When multiple occurrences of similar data was found, **mean()** was used to get an average of them and impute N/A's. If only one other row was found then its price iwas simply copied to replace the N/A value.

```
#All Rental NA
Rent %>% filter(rental_agency=="BED'R Apartments")
```

```
##   house_price    rental_agency      city bedrooms surface
## 1          NA BED’R Apartments Groningen        3      50
## 2          NA BED’R Apartments Groningen        2      55
## 3          NA BED’R Apartments Groningen        1      30
## 4          NA BED’R Apartments Groningen        2      30
## 5          NA BED’R Apartments Groningen        2      50
```

```
Rent %>% filter(city=="Groningen", bedrooms==3, surface==50)
```

```
##   house_price    rental_agency      city bedrooms surface
## 1          NA BED’R Apartments Groningen        3      50
## 2        1000 Tuitman Vastgoed Groningen        3      50
```

```
Rent$house_price[86] <- 1000
v <- Rent %>% filter(city=="Groningen", bedrooms==2, surface==55)
Rent$house_price[484] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(city=="Groningen", bedrooms==1, surface==30)
```

```
##   house_price                rental_agency      city bedrooms surface
## 1          NA             BED’R Apartments Groningen        1      30
## 2         850 Groningse Panden Beheer B.V. Groningen        1      30
```

```
Rent$house_price[589] <- 850
Rent %>% filter(city=="Groningen", bedrooms==2, surface==30)
```

```
##   house_price    rental_agency      city bedrooms surface
## 1          NA BED’R Apartments Groningen        2      30
```

```
v <- Rent %>% filter(bedrooms==2, surface==30)
Rent$house_price[638] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(city=="Groningen", bedrooms==2, surface==50)
```

```
##   house_price                rental_agency      city bedrooms surface
## 1          NA             BED’R Apartments Groningen        2      50
## 2        1350 Groningse Panden Beheer B.V. Groningen        2      50
## 3         850          Gruno Vastgoed B.V. Groningen        2      50
## 4        1175          Gruno Vastgoed B.V. Groningen        2      50
## 5        1020          Gruno Vastgoed B.V. Groningen        2      50
## 6        1000     Van der Meulen Makelaars Groningen        2      50
## 7         995                K&P Makelaars Groningen        2      50
```

```
v <- Rent %>% filter(city=="Groningen", bedrooms==2, surface==50)
Rent$house_price[690] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Short Stay Group B.V.")
```

```
##   house_price         rental_agency      city bedrooms surface
## 1          NA Short Stay Group B.V. Amsterdam        3      83
## 2          NA Short Stay Group B.V. Amsterdam        2      45
## 3          NA Short Stay Group B.V. Amsterdam        2      70
```

```r
v <- Rent %>% filter(city=="Amsterdam", bedrooms==3, surface==83)
Rent$house_price[9212] <- mean(v$house_price, na.rm=TRUE)
v <- Rent %>% filter(city=="Amsterdam", bedrooms==2, surface==45)
Rent$house_price[9215] <- mean(v$house_price, na.rm=TRUE)
v <- Rent %>% filter(city=="Amsterdam", bedrooms==2, surface==70)
Rent$house_price[9216] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Staffhousing Services B.V.")
```

```
##   house_price              rental_agency             city bedrooms surface
## 1          NA Staffhousing Services B.V. Pernis Rotterdam        5     100
## 2          NA Staffhousing Services B.V.            Hoorn        5     103
## 3          NA Staffhousing Services B.V.         Lelystad        5     135
```

```r
Rent %>% filter(city=="Pernis Rotterdam", bedrooms==5, surface==100)
```

```
##   house_price              rental_agency             city bedrooms surface
## 1          NA Staffhousing Services B.V. Pernis Rotterdam        5     100
```

```r
v <- Rent %>% filter(bedrooms==5, surface==100)
Rent$house_price[5821] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(city=="Hoorn", bedrooms==5, surface==103)
```

```
##   house_price              rental_agency  city bedrooms surface
## 1          NA Staffhousing Services B.V. Hoorn        5     103
```

```r
v <- Rent %>% filter(bedrooms==5, surface==103)
Rent$house_price[5833] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(city=="Lelystad", bedrooms==5, surface==135)
```

```
##   house_price              rental_agency     city bedrooms surface
## 1          NA Staffhousing Services B.V. Lelystad        5     135
## 2        1850                  2ndhome4u Lelystad        5     135
```

```r
Rent$house_price[6202] <- 1850

#!All Rental NA
Rent %>% filter(rental_agency=="Fransen Vastgoed")
```

```
##    house_price    rental_agency                     city bedrooms surface
## 1           NA Fransen Vastgoed              Vlaardingen        6     142
## 2          995 Fransen Vastgoed                Maassluis        3      65
## 3         1250 Fransen Vastgoed                Rotterdam        5     110
## 4         1095 Fransen Vastgoed              Schoonhoven        5      96
## 5         1295 Fransen Vastgoed                Rotterdam        2      75
## 6         1495 Fransen Vastgoed Ouderkerk aan den IJssel        5     125
## 7           NA Fransen Vastgoed               Sliedrecht        5      85
## 8           NA Fransen Vastgoed         Pernis Rotterdam        5     136
## 9          985 Fransen Vastgoed                Dordrecht        2      50
## 10        1375 Fransen Vastgoed                Rotterdam        2      56
## 11        1295 Fransen Vastgoed                 Schiedam        3      90
## 12        1095 Fransen Vastgoed                Rotterdam        2      65
## 13        1295 Fransen Vastgoed                Maassluis        4      67
## 14        1495 Fransen Vastgoed              Vlaardingen        5     100
## 15        1350 Fransen Vastgoed              Barendrecht        4      93
## 16          NA Fransen Vastgoed                Maassluis        4      75
## 17         975 Fransen Vastgoed                Rotterdam        2      45
```

```
Rent %>% filter(bedrooms==6, surface==142)
```

```
##   house_price            rental_agency        city bedrooms surface
## 1          NA         Fransen Vastgoed Vlaardingen        6     142
## 2        2450 Makelaarsassociatie B.V.    Den Haag        6     142
```

```
Rent$house_price[1203] <- 2450
v <- Rent %>% filter(bedrooms==5, surface==85)
Rent$house_price[5879] <- mean(v$house_price, na.rm=TRUE)
v <- Rent %>% filter(bedrooms==5, surface==136)
Rent$house_price[5881] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(city=="Maassluis", bedrooms==4, surface==75)
```

```
##   house_price    rental_agency      city bedrooms surface
## 1          NA Fransen Vastgoed Maassluis        4      75
```

```
v <- Rent %>% filter(bedrooms==4, surface==75)
Rent$house_price[8485] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Stam Vastgoed")
```

```
##    house_price rental_agency        city bedrooms surface
## 1         1645 Stam Vastgoed   Amsterdam        4      95
## 2          925 Stam Vastgoed   Hilversum        2      40
## 3         1750 Stam Vastgoed   Hilversum        3     110
## 4         1500 Stam Vastgoed      Almere        4      95
## 5         1550 Stam Vastgoed   Vinkeveen        3      75
## 6         1300 Stam Vastgoed   Hilversum        2      60
## 7         1200 Stam Vastgoed   Hilversum        2      60
## 8          925 Stam Vastgoed   Hilversum        2      55
## 9         1100 Stam Vastgoed   Hilversum        2      67
## 10        1500 Stam Vastgoed      Almere        4      80
## 11        1600 Stam Vastgoed   Hilversum        3     110
## 12        2250 Stam Vastgoed   Hilversum        5     200
## 13         950 Stam Vastgoed       Weesp        1      25
## 14        1050 Stam Vastgoed       Weesp        2      40
## 15        1120 Stam Vastgoed   Hilversum        2      60
## 16         750 Stam Vastgoed  Kortenhoef        2      45
## 17        1825 Stam Vastgoed Soesterberg        6     160
## 18        2000 Stam Vastgoed   Hilversum        2      90
## 19        2750 Stam Vastgoed   Amsterdam        3     130
## 20        1250 Stam Vastgoed   Hilversum        2     100
## 21        2000 Stam Vastgoed   Hilversum        3      90
## 22        1500 Stam Vastgoed   Hilversum        3     100
## 23        2500 Stam Vastgoed   Amsterdam        4     120
## 24        1500 Stam Vastgoed   Amsterdam        3      70
## 25        1500 Stam Vastgoed   Vinkeveen        3      75
## 26        1525 Stam Vastgoed   Hilversum        3     110
## 27        2000 Stam Vastgoed       Weesp        4      98
## 28        1250 Stam Vastgoed       Laren        3      90
## 29        1650 Stam Vastgoed  Loosdrecht        1      60
## 30        3850 Stam Vastgoed  Loosdrecht        5     150
## 31          NA Stam Vastgoed  Loosdrecht        1      50
```

```
v <- Rent %>% filter(bedrooms==1, surface==50)
Rent$house_price[8852] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Havos Vastgoedbelegging bv")
```

```
##   house_price              rental_agency       city bedrooms surface
## 1        2500 Havos Vastgoedbelegging bv Appingedam        6     110
## 2          NA Havos Vastgoedbelegging bv  Uithuizen       10     130
## 3         895 Havos Vastgoedbelegging bv  Groningen        2      55
## 4         950 Havos Vastgoedbelegging bv  Groningen        3      62
## 5         865 Havos Vastgoedbelegging bv   Haren Gn        3      72
```

```
Rent %>% filter(bedrooms==10) ##
```

```
##    house_price                          rental_agency          city bedrooms
## 1         3000                       123Wonen Alkmaar Heerhugowaard       10
## 2           NA             Havos Vastgoedbelegging bv     Uithuizen       10
## 3         2475                     Rotsvast Eindhoven   Westerhoven       10
## 4         2950                       Tameling Verhuur       Katwijk       10
## 5         5000 Best Intermediair Vastgoed Makelaardij      Oirschot       10
## 6         3500                      123Wonen Den Haag    Moordrecht       10
## 7         4000                 ’t Gooi Estate Rentals     Hilversum       10
## 8         4950                        Het Hoofse Huis    Maastricht       10
## 9         5000         Von Poll Real Estate - Centrum     Amsterdam       10
## 10          NA                           AB&P Vaassen      Roermond       10
## 11          NA                           AB&P Vaassen      Roermond       10
## 12        1950                Visschedijk Makelaardij    Maastricht       10
## 13        3500        NumberXII Exclusive Real Estate      Den Haag       10
## 14        1650            Zuyd Makelaardij & Vastgoed    Maastricht       10
## 15        5000                            Vivir Wonen         Spijk       10
##    surface
## 1      435
## 2      130
## 3      400
## 4      235
## 5      362
## 6      346
## 7      255
## 8      288
## 9      200
## 10     340
## 11     262
## 12     161
## 13     189
## 14     110
## 15     350
```

```r
Rent$house_price[2182] <- 1650
Rent %>% filter(rental_agency=="Comfortable Staff Housing")
```

```
##   house_price             rental_agency       city bedrooms surface
## 1        3250 Comfortable Staff Housing     Bussum        5     150
## 2        1650 Comfortable Staff Housing  Amsterdam        3      85
## 3          NA Comfortable Staff Housing  Westbroek        8     250
## 4        3300 Comfortable Staff Housing  Amsterdam        3     120
## 5        5500 Comfortable Staff Housing  Amsterdam        3     181
## 6        2250 Comfortable Staff Housing Amstelveen        3      90
## 7        2250 Comfortable Staff Housing Amstelveen        4     125
```

```r
Rent %>% filter(bedrooms==8) ##
```

```
##    house_price                   rental_agency           city bedrooms surface
## 1         4000                123Wonen Zeeland           Goes        8     298
## 2         2900                  Rotsvast Breda          Breda        8     179
## 3         1250                     Sterckwonen        Sittard        8     143
## 4         3500  Residence Housing & Relocation      Landgraaf        8     332
## 5         1575 Van der Laarse Makelaardij o.g.       Aalsmeer        8     178
## 6         1250              Rotsvast Eindhoven        Helmond        8     180
## 7         3500               VERRA Real Estate       Den Haag        8     230
## 8         4400         The Real Estate Company       Den Haag        8     370
## 9         2450               VERRA Real Estate      Wassenaar        8     150
## 10        3750           Estata Makelaars O.G.       Den Haag        8     232
## 11          NA       Comfortable Staff Housing      Westbroek        8     250
## 12        3560              Grand Prix Rentals   Valkenswaard        8     340
## 13        1500              Expatdesk Nijmegen       Nijmegen        8     220
## 14        2495                123Wonen Tilburg     Moergestel        8     421
## 15        6250            Amstelland Makelaars      Amsterdam        8     265
## 16        3000  The Hague Real Estate Services       Voorburg        8     160
## 17        2750                        Listings        Naarden        8     160
## 18        2500           123Wonen West-Brabant Bergen op Zoom        8     240
## 19        1295              Rotsvast Eindhoven      Eindhoven        8     150
## 20        2500  The Hague Real Estate Services       Den Haag        8     194
## 21        1850                 Wij Makelaardij        Utrecht        8     108
## 22        2800                  Aaiman Rentals      Dreischor        8     130
## 23        3950                Tameling Verhuur        Katwijk        8     212
## 24        3950           Estata Makelaars O.G.       Den Haag        8     325
## 25        2500             The Housing Company         Urmond        8     160
## 26        3750           Estata Makelaars O.G.       Den Haag        8     264
## 27        1750                      EHR Arnhem         Arnhem        8     148
## 28        4250                DSTRCT Amsterdam      Amsterdam        8     165
## 29        3750                 Avenir Vastgoed       Den Haag        8     210
## 30        2250                   EasyMakelaars         Leiden        8     167
## 31        3600             BjÃ¶rnd Makelaardij          Delft        8     240
## 32        4250               VERRA Real Estate       Den Haag        8     265
## 33        7000        Expat & Real Estate B.V.       Den Haag        8     560
## 34        4250            Wunderink & de Lange      Wassenaar        8     190
## 35        3250           Estata Makelaars O.G.       Den Haag        8     180
## 36        3950               VERRA Real Estate       Den Haag        8     245
## 37        1100                Honings Vastgoed       Bocholtz        8      89
## 38        5500            Rappange Makelaardij      Amsterdam        8     240
## 39        4400           Your Home Makelaardij     Amstelveen        8     225
## 40        8675         Dutch Housing Centre BV      Amsterdam        8     240
## 41        2750     Hakkenbroek Housing Company      Hilversum        8     150
## 42        1800                    Domica Venlo        Meterik        8     240
```

```
Rent$house_price[2429] <- 3950
Rent %>% filter(rental_agency=="Expat Group")
```

```
##   house_price rental_agency    city bedrooms surface
## 1          NA   Expat Group Tilburg        6     150
## 2        1050   Expat Group Tilburg        2      75
## 3        1325   Expat Group Tilburg        3      56
```

```
v <- Rent %>% filter(bedrooms==6, surface==150)
Rent$house_price[4659] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="AB&P Vaassen")
```

```
##   house_price rental_agency     city bedrooms surface
## 1         995  AB&P Vaassen Roermond        7     123
## 2         737  AB&P Vaassen     Echt        5      80
## 3        1500  AB&P Vaassen Roermond        7     100
## 4         865  AB&P Vaassen     Echt        7     100
## 5         690  AB&P Vaassen     Echt        4      40
## 6          NA  AB&P Vaassen Roermond       10     340
## 7          NA  AB&P Vaassen Roermond       10     262
## 8          NA  AB&P Vaassen Roermond        4      90
## 9          NA  AB&P Vaassen Roermond        6     118
```

```
Rent %>% filter(bedrooms==10) ##
```

```
##    house_price                          rental_agency          city bedrooms
## 1         3000                       123Wonen Alkmaar Heerhugowaard       10
## 2         1650             Havos Vastgoedbelegging bv     Uithuizen       10
## 3         2475                     Rotsvast Eindhoven   Westerhoven       10
## 4         2950                       Tameling Verhuur       Katwijk       10
## 5         5000 Best Intermediair Vastgoed Makelaardij      Oirschot       10
## 6         3500                      123Wonen Den Haag    Moordrecht       10
## 7         4000                 ’t Gooi Estate Rentals     Hilversum       10
## 8         4950                        Het Hoofse Huis    Maastricht       10
## 9         5000         Von Poll Real Estate - Centrum     Amsterdam       10
## 10          NA                           AB&P Vaassen      Roermond       10
## 11          NA                           AB&P Vaassen      Roermond       10
## 12        1950                Visschedijk Makelaardij    Maastricht       10
## 13        3500        NumberXII Exclusive Real Estate      Den Haag       10
## 14        1650            Zuyd Makelaardij & Vastgoed    Maastricht       10
## 15        5000                            Vivir Wonen         Spijk       10
##    surface
## 1       435
## 2       130
## 3       400
## 4       235
## 5       362
## 6      346
## 7      255
## 8      288
## 9      200
## 10     340
## 11     262
## 12     161
## 13     189
## 14     110
## 15     350
```

```
Rent$house_price[8082] <- 5000
Rent$house_price[8083] <- 4000
v <- Rent %>% filter(bedrooms==4, surface==90)
Rent$house_price[8837] <- mean(v$house_price, na.rm=TRUE)
v <- Rent %>% filter(bedrooms==6, surface==118)
Rent$house_price[8919] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="123Wonen Flevoland")
```

```
##   house_price      rental_agency          city bedrooms surface
## 1        1750 123Wonen Flevoland        Almere        5     130
## 2        1000 123Wonen Flevoland        Almere        2      50
## 3        1650 123Wonen Flevoland       Dronten        6     118
## 4          NA 123Wonen Flevoland Biddinghuizen        2      56
```

```
v <- Rent %>% filter(bedrooms==2, surface==56)
Rent$house_price[9167] <- mean(v$house_price, na.rm=TRUE)

#Appears once
Rent %>% filter(rental_agency=="Makelaarskantoor Paul Schreinemachers")
```

```
##   house_price                         rental_agency  city bedrooms surface
## 1          NA Makelaarskantoor Paul Schreinemachers Venlo        2      28
```

```
Rent %>% filter(city=="Venlo", bedrooms==2, surface==28)
```

```
##   house_price                         rental_agency  city bedrooms surface
## 1          NA Makelaarskantoor Paul Schreinemachers Venlo        2      28
```

```
v = Rent %>% filter(bedrooms==2, surface==28)
Rent$house_price[2728] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="vastgoedPROmakelaar.nl")
```

```
##   house_price          rental_agency  city bedrooms surface
## 1          NA vastgoedPROmakelaar.nl Weert        4      85
```

```
Rent %>% filter(city=="Weert", bedrooms==4, surface==85)
```

```
##   house_price          rental_agency  city bedrooms surface
## 1          NA vastgoedPROmakelaar.nl Weert        4      85
```

```
v = Rent %>% filter(bedrooms==4, surface==85)
Rent$house_price[4360] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Huijers Vastgoed Makelaardij")
```

```
##   house_price                rental_agency  city bedrooms surface
## 1          NA Huijers Vastgoed Makelaardij Weert        4     135
```

```
Rent %>% filter(city=="Weert", bedrooms==4, surface==135)
```

```
##   house_price                rental_agency  city bedrooms surface
## 1          NA Huijers Vastgoed Makelaardij Weert        4     135
```

```
v = Rent %>% filter(bedrooms==4, surface==135)
Rent$house_price[7943] <- mean(v$house_price, na.rm=TRUE)
Rent %>% filter(rental_agency=="Gerro de Boer Makelaardij & Taxaties o.g.")
```

```
##   house_price                             rental_agency      city bedrooms
## 1        1100 Gerro de Boer Makelaardij & Taxaties o.g.      Edam        3
## 2          NA Gerro de Boer Makelaardij & Taxaties o.g. Purmerend        6
##   surface
## 1      59
## 2     290
```

```
head(Rent %>% filter(bedrooms==6))
```

```
##   house_price          rental_agency       city bedrooms surface
## 1        1500         Vesta Vastgoed Maastricht        6     120
## 2        1335            vdpvastgoed    Katwijk        6     157
## 3        2900           Rent a Stone   Den Haag        6     149
## 4         950                ViaDaan   Kerkrade        6     155
## 5        1495 HouseHunting Eindhoven  Eindhoven        6     135
## 6        1355     Van Gerwen Housing Maastricht        6     120
```

```
Rent$house_price[8520] <- 4500
Rent %>% filter(rental_agency=="NL en Wonen")
```

```
##   house_price rental_agency       city bedrooms surface
## 1        1200   NL en Wonen  Hilversum        4      70
## 2        1250   NL en Wonen     Almere        3      54
## 3        1000   NL en Wonen     Bussum        2      55
## 4          NA   NL en Wonen IJzendijke       15     360
## 5          NA   NL en Wonen     Bussum        5     144
```

```
v <- Rent %>% filter(bedrooms==5, surface==144)
Rent$house_price[7617] <- mean(v$house_price, na.rm=TRUE)
```

Two N/As were omitted from the data set, this is due to the data not having any similarities with other data and this is due to both having a large number of bedrooms and not being able to be imputed.

```r
Rent %>% filter(rental_agency=="Grand Prix Rentals")
```

```
##   house_price      rental_agency         city bedrooms surface
## 1        3560 Grand Prix Rentals Valkenswaard        8     340
## 2          NA Grand Prix Rentals Valkenswaard        9     878
```

```r
Rent %>% filter(bedrooms==9)
```

```
##    house_price                  rental_agency             city bedrooms surface
## 1         4750            First Class Housing       Amstelveen        9     210
## 2         1600 Residence Housing & Relocation          Eijsden        9     123
## 3         3250     Floberg Makelaardij Bussum           Bussum        9     143
## 4         4000         ’t Gooi Estate Rentals           Bussum        9     280
## 5         1450             123Wonen Den Bosch Heeswijk-Dinther        9     210
## 6         3950            First Class Housing         Aalsmeer        9     240
## 7         3000       Dorenbos Rasch Makelaars       Loosdrecht        9     271
## 8           NA             Grand Prix Rentals     Valkenswaard        9     878
## 9         3500    VERRA Real Estate Rotterdam        Rotterdam        9     310
## 10        4500 The Hague Real Estate Services         Den Haag        9     264
## 11        4350          Estata Makelaars O.G.         Den Haag        9     325
```

```r
Rent <- Rent[-c(8037), ]
row.names(Rent) <- NULL

Rent %>% filter(rental_agency=="NL en Wonen")
```

```
##   house_price rental_agency       city bedrooms surface
## 1    1200.000   NL en Wonen  Hilversum        4      70
## 2    1250.000   NL en Wonen     Almere        3      54
## 3    1000.000   NL en Wonen     Bussum        2      55
## 4          NA   NL en Wonen IJzendijke       15     360
## 5    2133.333   NL en Wonen     Bussum        5     144
```

```r
Rent %>% filter(surface==360)
```

```
##   house_price rental_agency       city bedrooms surface
## 1          NA   NL en Wonen IJzendijke       15     360
```

```r
Rent <- Rent[-c(7616), ]
row.names(Rent) <- NULL

x <- which(is.na(Rent))
Rent[x, ]
```

```
## [1] house_price   rental_agency city          bedrooms      surface
## <0 rows> (or 0-length row.names)
```

Task 3: Data Analysis

Sample Means

After cleaning the data, a sample using systematic sampling was created. The sample means for the variables house_price, bedrooms and surface were calculated to produce the average values for a typical property. Below is the code to create a data frame with the cleaned data.

```r
##create a data frame
housePrice<-c(Rent$house_price)
rentalAgency<-c(Rent$rental_agency)
cityLocation<-c(Rent$city)
bedrooms<-c(Rent$bedrooms)
surfaceArea<-c(Rent$surface)
rentDataSet.data<-data.frame(housePrice, rentalAgency, cityLocation, bedrooms, surfaceArea)
##str(rentDataSet.data)
```

After this the sample_n() function was used to create a sample from the population.

```r
##Create sample from population
##9717/3=3239
sampleData = sample_n(rentDataSet.data, (nrow(rentDataSet.data)/3), FALSE)
##print(sampleData)
```

The mean() function was applied to the previously mentioned variables to produce the average values, some of which are rounded to two decimal places. The code below demonstrates this. The results for the three sample mean values can be found below the related code.

```r
##Task 3 Part 1
sampleMeanPrice = mean(sampleData$housePrice, na.rm = TRUE)
sampleMeanPrice<- round(sampleMeanPrice, digits=2)
print(sampleMeanPrice)
```

```
## [1] 1440.22
```

```r
sampleMeanBedroom = mean(sampleData$bedrooms, na.rm = TRUE)
sampleMeanBedroom<- round(sampleMeanBedroom, digits=0)
print(sampleMeanBedroom)
```

```
## [1] 3
```

```r
sampleMeanSurfaceArea = mean(sampleData$surfaceArea, na.rm = TRUE)
sampleMeanSurfaceArea<- round(sampleMeanSurfaceArea, digits=2)
print(sampleMeanSurfaceArea)
```

```
## [1] 79.05
```

Most Expensive / Cheapest

To find the most expensive and cheapest cities, a data frame was created to store each unique city with the price per m^2.

```
City <- NULL
PriceperSqm <- NULL
df <- data.frame(City, PriceperSqm)
```

To populate this data frame, both the names of cities and price per m^2 need to imputed and this is done with a for loop that goes over every unique city, filters the city inside test, calculates price per m^2 by doing $\frac{houseprice}{surface}$ for all rows of that city and finally calculating the mean of each price per m^2 per city, rounded to two decimal places, and populating the result alongside the city name in the new data set.

```
for(i in unique(Rent$city)){
  Test <- Rent %>% filter(city==i)
  Test$PriceperSqmeter <- Test$house_price / Test$surface
  PriceperSqm <- mean(Test$PriceperSqmeter, na.rm=TRUE)
  PriceperSqm <- round(PriceperSqm, digits=3)
  City <- i
  new_row <- c(City, PriceperSqm)
  df <- rbind(df, new_row)
}
names(df)[1]<-paste("City")
names(df)[2]<-paste("PriceperSqm")
head(df)
```

```
##          City PriceperSqm
## 1      Diemen      21.496
## 2     Utrecht      25.777
## 3   Rotterdam      20.322
## 4 Spijkenisse       15.19
## 5     Tilburg      21.214
## 6   Amsterdam      26.127
```

After the loop the column *PriceperSqm* was in a string format so **as.numeric** was used to transform the data to numerical ones, **max()** and **min()** were both used on the data frame to find the most expensive, the largest value, and the cheapest, the smallest value.

```
is.numeric(df$PriceperSqm)
```

```
## [1] FALSE
```

```
df$PriceperSqm <- as.numeric(df$PriceperSqm)
is.numeric(df$PriceperSqm)
```

```
## [1] TRUE
```

```
max <- df %>% filter(PriceperSqm==max(df$PriceperSqm))
min <- df %>% filter(PriceperSqm==min(df$PriceperSqm))
```

With this implementation **Beinsdorp** was found to be the most expensive city, while **Wegenborgen** was the cheapest.

```
## [1] "Most expensive:  Beinsdorp , 44.048 per m^2"
```

```
## [1] "Cheapest:  Wagenborgen , 0.295 per m^2"
```

Heatmap

Below is the code to create a data frame for the population, from which a sample of the population was created. A sample was used to create the heatmap because…..

```
##create a data frame
housePriceHM<-c(Rent$house_price)
rentalAgencyHM<-c(Rent$rental_agency)
cityLocationHM<-c(Rent$city)
bedroomsHM<-c(Rent$bedrooms)
surfaceAreaHM<-c(Rent$surface)

rentDataSetHM.data<-data.frame(housePriceHM, rentalAgencyHM, cityLocationHM, bedroomsHM, surfaceAreaHM)
str(rentDataSetHM.data)
```

```
## 'data.frame':    9717 obs. of  5 variables:
##  $ housePriceHM   : num  575 835 1095 1295 425 ...
##  $ rentalAgencyHM: chr  "OurCampus Amsterdam Diemen" "Nido Student" "Rotterdam Apartments" "Rotterdam
##  $ cityLocationHM: chr  "Diemen" "Utrecht" "Rotterdam" "Rotterdam" ...
##  $ bedroomsHM    : int  1 1 1 2 1 5 1 3 3 2 ...
##  $ surfaceAreaHM : int  27 20 40 55 17 111 32 84 50 100 ...
```

```
##Create sample from population
##9717/3=3239
sampleDataHM = sample_n(rentDataSetHM.data, (nrow(rentDataSetHM.data)/3), FALSE)
##head(sampleDataHM)
##str(sampleDataHM)
##table(sampleDataHM$housePriceHM)
##table(sampleDataHM$cityLocationHM)
```

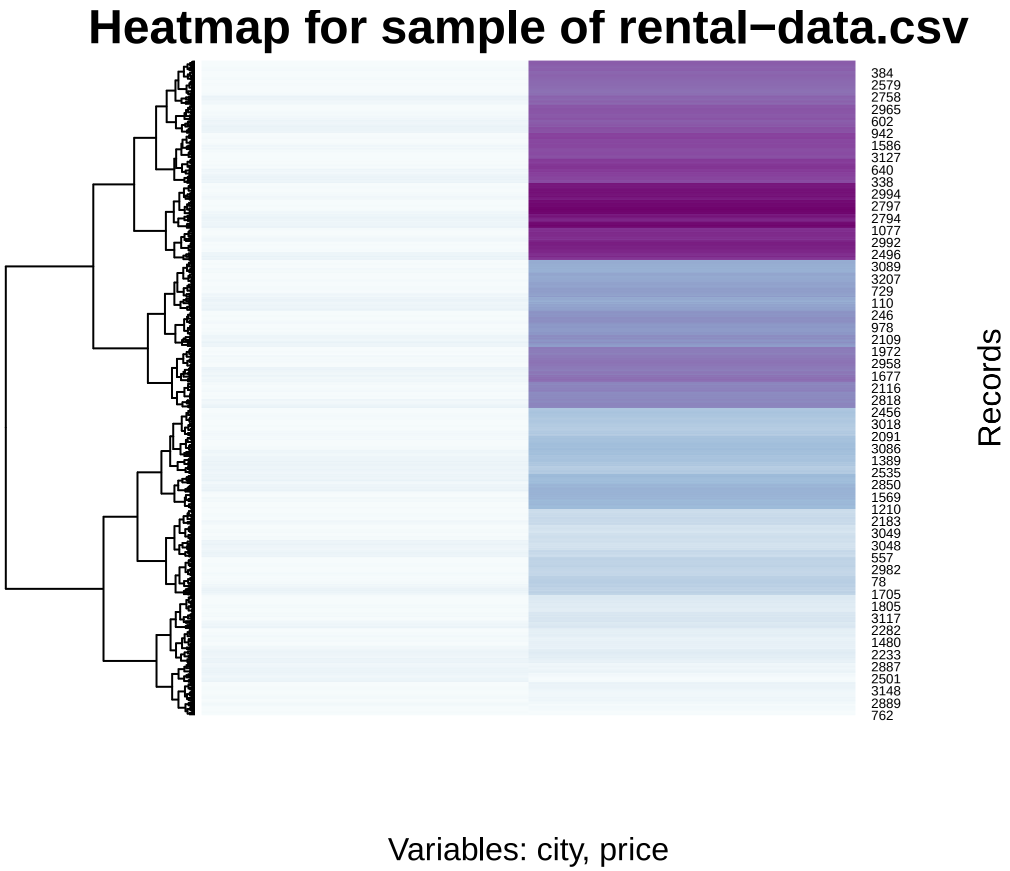
The variables location and house_price were separated, then bound and stored in a matrix to be passed as the data used by the heatmap. However, first the data for location was converted to a numeric equivalent using the factor() function, as matrices only accept numeric input which was necessary to create the heatmap. Secondly, the data was stored in the matrix in ascending order according to house_price.

```
city<-factor(sampleDataHM$cityLocationHM)##locationFactor
matrixPrice<-order(sampleDataHM$housePriceHM)#ascending
price<-as.numeric(matrixPrice)
heatMapMatrix<-cbind(city, price)
##print(heatMapMatrix)

##table(city)
```

The RColourBrewer library was used to generate the colours for the heatmap. It utilises the data consisting of location and house_price and shows the relationship between them. The variables are on the x-axis with the row numbers on the y-axis. Furthermore the the gradient of the colours spans from light to dark, with lighter colours representing the lower range of prices for a property according to location and darker colours representing more expensive properties.

```
colouring<-colorRampPalette(brewer.pal(8,"BuPu"))(3239)

##High values are dark, low values are light
```

```
rentalHeatmap<-heatmap(heatMapMatrix,
                        Colv = NA,
                        #Rowv = NA,
                        scale="none",
                        col = colouring,
                        xlab = "Variables: city, price",
                        ylab = "Records",
                        labCol = FALSE,
                        main = "Heatmap for sample of rental-data.csv")
```



It is noted that the heatmap does contain some errors. The variable city would have ideally been on the y-axis instead of beside price on the x-axis. This occurred because of the lack of row names for each individual record in the data set and two columns of data were required to create the matrix used by heatmap().

However the heatmap can be interpreted, it was concluded that there is a direct relationship between the price of a property and the city it is situated in.

Some of most expensive cities, according to the sample, include:

- Wassenaar
- Amsterdam
- Den Haag
- Oirschot
- Spijk
- Maastricht

Some of the cheaper cities, according to the sample, include:

- Tilberg
- Waardenburg
- Onstwedde
- Eindhoven
- Gaanderen
- Arnhem

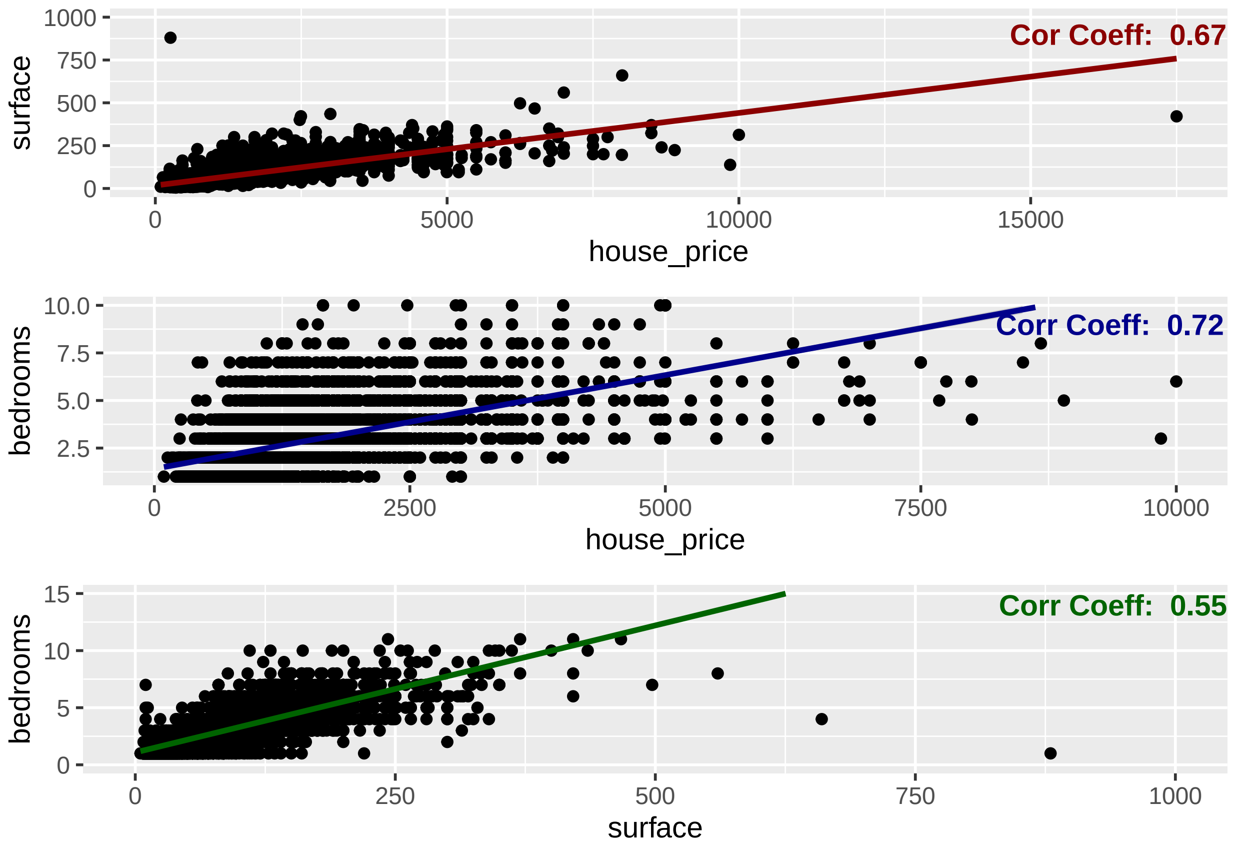
Correlations

To identify the correlation between the values, three scatter plots were created of all the numerical data compared against one another. **ggplot** was used to visualize the three scatter plots whilst **grid.arrage()** was used to put them all together.

Apart from this the correlation co-efficiency was calculated using the **cor()**, rounded to 2 decimal places, and this was inserted into each scatter plot to give both a visual indication of each correlation while also a numerical one.

```
corone <-cor(Rent$house_price, Rent$surface)
corone <- round(corone, digits=2)
messageone <- paste("Cor Coeff: ", corone)
cortwo <- cor(Rent$bedrooms, Rent$surface)
cortwo <- round(cortwo, digits=2)
messagetwo <- paste("Corr Coeff: ", cortwo)

corthree <- cor(Rent$house_price, Rent$bedrooms)
corthree <- round(corthree, digits=2)
messagethree <- paste("Corr Coeff: ", corthree)
plot1 <- ggplot(Rent, aes(x=house_price, y=surface, messageone)) + geom_point()+
  geom_smooth(method=lm, color="darkred") + ylim(0, 1000)+ annotate("text", x = 16500, y = 900, label =
                                                                    colour = "darkred", fontface =2)
plot2 <- ggplot(Rent, aes(x=house_price, y=bedrooms)) + geom_point()+
  geom_smooth(method=lm, color="darkblue") + ylim(1, 10) + xlim(0, 10000)+ annotate("text", x = 9350, y
                                                                                    colour = "darkblue"
plot3 <- ggplot(Rent, aes(x=surface, y=bedrooms)) + geom_point()+
  geom_smooth(method=lm, color="darkgreen") + ylim(0, 15) + xlim(0, 1000)+ annotate("text", x = 940, y
                                                                                    colour = "darkgreen
grid.arrange(plot1, plot2, plot3, ncol=1)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
## ‘geom_smooth()‘ using formula ’y ~ x’
```



With this it was concluded that there is a positive relationship between the continuous quantitative variables. *bedrooms vs surface* has the strongest correlation of 0.72 while *house_price vs surface* has the weakest out of the three with a moderate correlation of 0.55.

Distribution and Standard Deviation

The distribution and standard deviation between Amsterdam and Rotterdam were identified with two histograms layered on one anther showing the difference between the two while the standard deviation was calculated for both and presented in the plot.

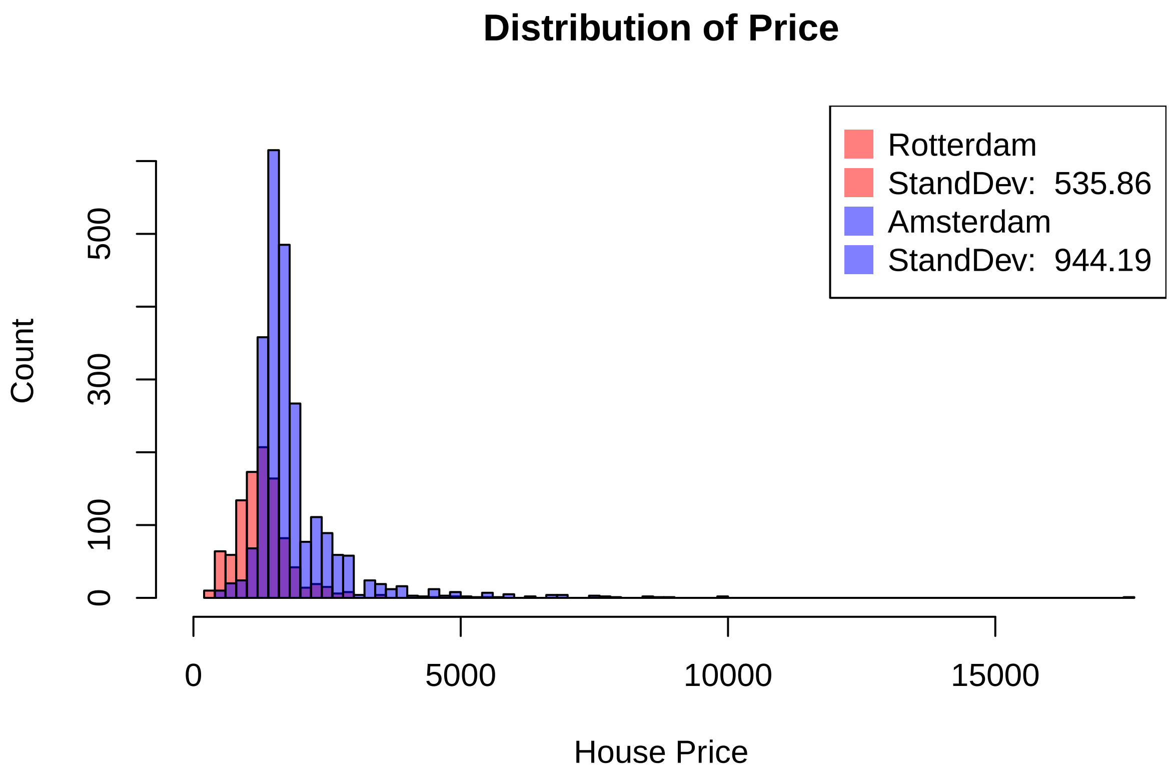
The data set was filtered by the required cities and the standard deviation was calculated using **var()** which calculates the variance of the data and after the

$$\sqrt{variance}$$

was found to get the standard deviation.

```r
Rotter <- Rent %>% filter(city=="Rotterdam")
Amster <- Rent %>% filter(city=="Amsterdam")
Varone <- var(Rotter$house_price)
Stdevone <- sqrt(Varone)
Stdevone <- round(Stdevone, digits=2)
Vartwo<- var(Amster$house_price)
Stdevtwo <- sqrt(Vartwo)
Stdevtwo <- round(Stdevtwo, digits=2)
message1 <- paste("StandDev: ", Stdevone)
message2 <- paste("StandDev: ", Stdevtwo)
```

```
hist(Rotter$house_price, breaks=20, xlim=c(0,17500),ylim=c(0,650) , col=rgb(1,0,0,0.5), xlab="House Pri
     ylab="Count", main="Distribution of Price" )
hist(Amster$house_price, breaks=80, xlim=c(0,17500),ylim=c(0,650), col=rgb(0,0,1,0.5), add=T)
legend("topright", legend=c("Rotterdam",message1,"Amsterdam",message2), col=c(rgb(1,0,0,0.5), rgb(1,0,0
                                                                              rgb(0,0,1,0.5), rgb(0,0,1
```



Between Rotterdam and Amsterdam, Rotterdam has the smallest standard deviation meaning that values are more tightly clustered around the mean. Amsterdam has a greater range of values which are spread out, this is also shown on the plot.

To check for skewness the mean, mode and median were all calculated with *mean()* and *median()* while for mode the function **getmode** was created since R doesn't have it built in by standard.

the library *e1071* was utilized to calculate the skewness of both data frames.

```
library(e1071)

getmode <- function(v) {
  mode <- unique(v)
  mode[which.max(tabulate(match(v, mode)))]
}

skewone <- skewness(Rotter$house_price)
skewone <- round(skewone, digits=2)

skewtwo <- skewness(Amster$house_price)
skewtwo <- round(skewtwo, digits=2)
```

```
meanone <- mean(Rotter$house_price)
meanone <- round(meanone, digits=2)

meantwo <- mean(Amster$house_price)
meantwo <- round(meantwo, digits=2)

modeone <- getmode(Rotter$house_price)
modetwo <- getmode(Amster$house_price)

medianone <- median(Rotter$house_price)
mediantwo <- median(Amster$house_price)
```

```
## [1] "Rotterdam mean/mode/median/skewness  1250 / 1250 / 1320.1 / 1.96"
```

```
## [1] "Amsterdam mean/mode/median/skewness  1500 / 1650 / 1899.24 / 5.01"
```

Therefore it was concluded that both Rotterdam and Amsterdam have a positive skewness which is also visualized in the plot, Amsterdam has a higher skew with that of 5.01 then Rotterdam with 1.96.

Regression

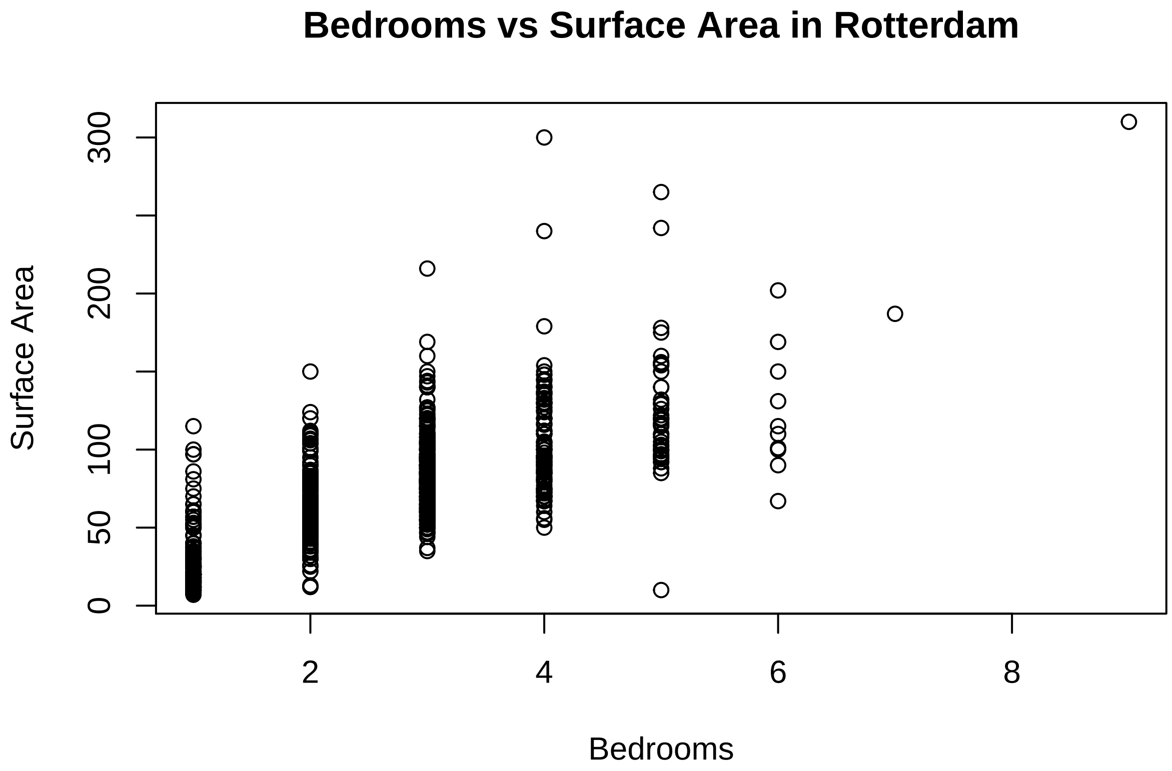
Regression was used to infer predictions for various continuous quantitative variables with respect to a discrete qualitative variable, city, specifically for Amsterdam and Rotterdam.

Initially the population was filtered according to city for Amsterdam and Rotterdam. A scatter plot was created to display the relationship and the correlation calculated between two continuous qualitative variables for the respective cities. This was carried out for:

- bedrooms VS surface.
- house_price VS surface.

For Rotterdam:

```
#Uses population
#(1) the typical m2 apartment with 3 bedrooms in Amsterdam and Rotterdam;
rotterdamSurface<-Rent %>% filter(city=="Rotterdam")

plot(rotterdamSurface$bedrooms, rotterdamSurface$surface, xlab="Bedrooms", ylab="Surface Area", main="Be
```



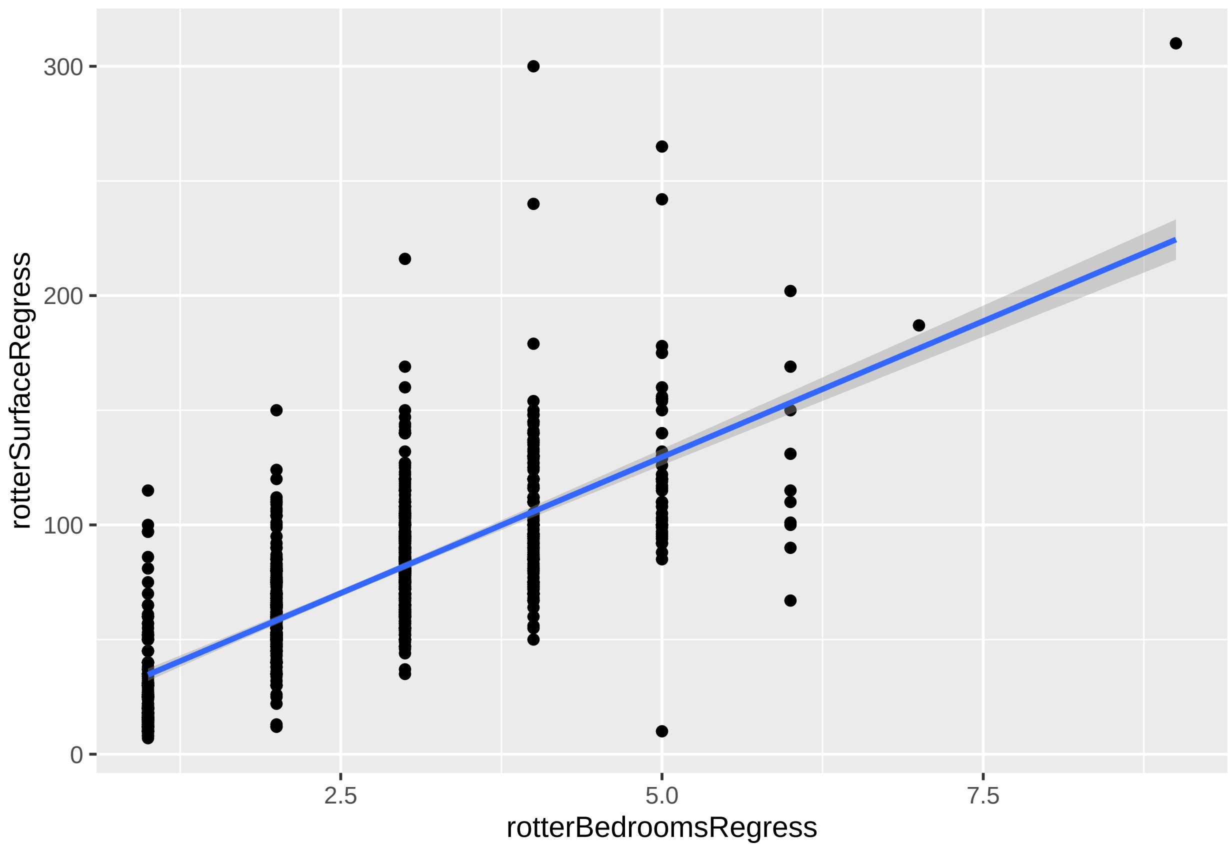
```
cor(rotterdamSurface$surface,rotterdamSurface$bedrooms)#0.7328429
```

```
## [1] 0.7328429
```

A resulting correlation of 0.7328429 indicated that the variables are dependent meaning that a linear model can be created and regression utilised to predict the surface area for a property with three bedrooms in Rotterdam. A scatter plot including the linear model displays the relationship and the result of the prediction is in the code below.

```
rotterSurfaceRegress<-rotterdamSurface$surface
rotterBedroomsRegress<-rotterdamSurface$bedrooms
rotterDataFrame<-data.frame(rotterBedroomsRegress,rotterSurfaceRegress)
rotterRegres<-lm(formula = rotterSurfaceRegress~rotterBedroomsRegress, data = rotterDataFrame)
ggplot(rotterdamSurface, aes(x=rotterBedroomsRegress, y=rotterSurfaceRegress)) + geom_point() + geom_sm
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```r
summary(rotterRegres)
```

```
##
## Call:
## lm(formula = rotterSurfaceRegress ~ rotterBedroomsRegress, data = rotterDataFrame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.547  -14.653   -3.376   11.624  194.176
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             10.9290     2.0050   5.451 6.31e-08 ***
## rotterBedroomsRegress  23.7237     0.6955  34.111  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.53 on 1003 degrees of freedom
## Multiple R-squared:  0.5371, Adjusted R-squared:  0.5366
## F-statistic:  1164 on 1 and 1003 DF,  p-value: < 2.2e-16
```

```r
print(rotterRegres)
```

```
##
```

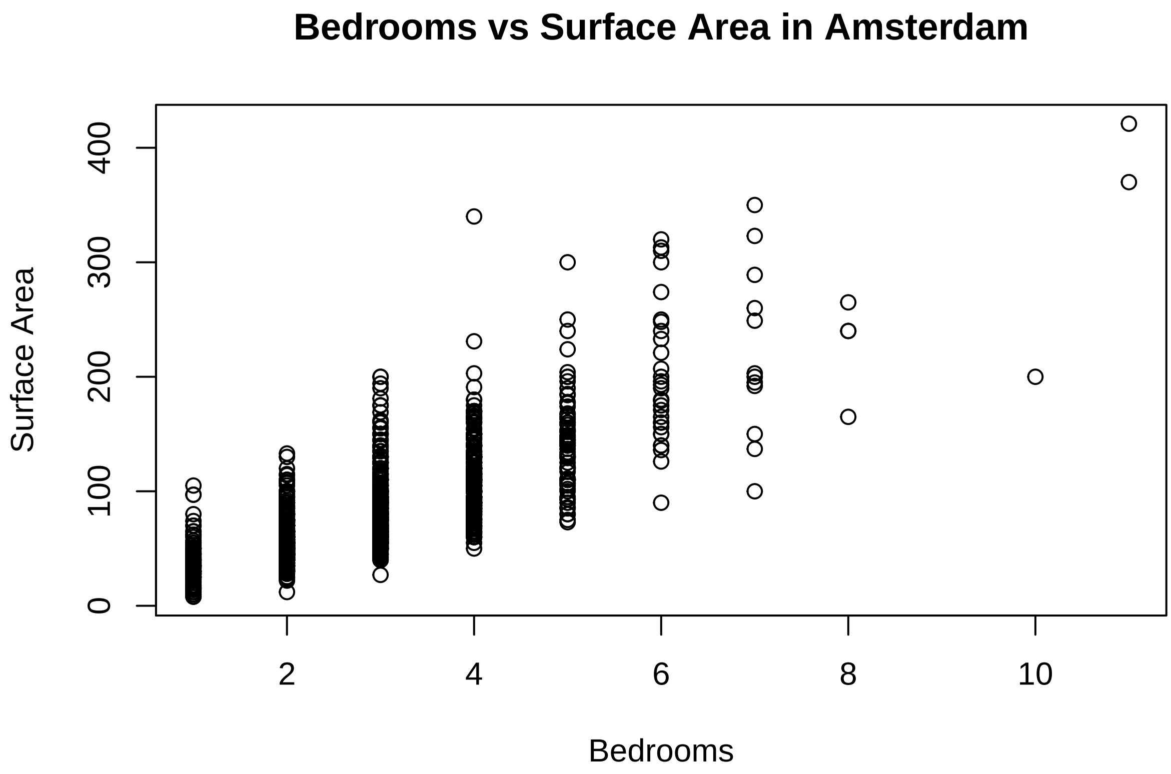
```
## Call:
## lm(formula = rotterSurfaceRegress ~ rotterBedroomsRegress, data = rotterDataFrame)
##
## Coefficients:
##           (Intercept)  rotterBedroomsRegress
##                 10.93                  23.72
```

```
#Manual way to predict: surfaceArea = 10.93 + (23.72*bedrooms)
rotterPredict<-predict(rotterRegres, list(rotterBedroomsRegress = 3))
rotterPredict#82.1 m^2
```

```
##    1
## 82.1
```

For Amsterdam:

```
amsterdamSurface<-Rent %>% filter(city=="Amsterdam")

plot(amsterdamSurface$bedrooms, amsterdamSurface$surface, xlab="Bedrooms", ylab="Surface Area", main="Be
```



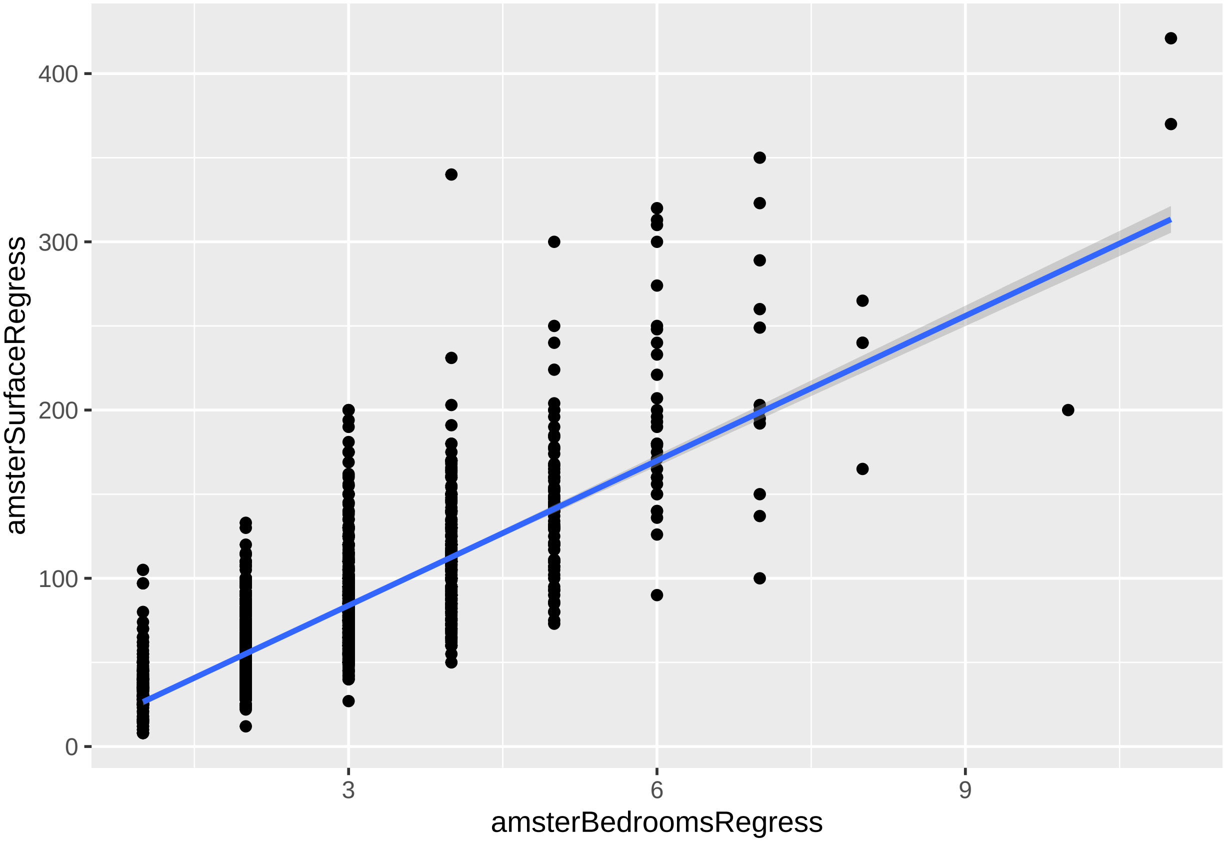
```
cor(amsterdamSurface$surface,amsterdamSurface$bedrooms)#0.7682397
```

```
## [1] 0.7682397
```

A resulting correlation of 0.7682397 indicated that the variables are dependent meaning that a linear model can be created and regression utilised to predict the surface area for a property with three bedrooms in Amsterdam. A scatter plot including the linear model displays the relationship and the result of the prediction is in the code below.

```
amsterSurfaceRegress<-amsterdamSurface$surface
amsterBedroomsRegress<-amsterdamSurface$bedrooms
amsterDataFrame<-data.frame(amsterBedroomsRegress,amsterSurfaceRegress)
amsterRegres<-lm(formula = amsterSurfaceRegress~amsterBedroomsRegress, data = amsterDataFrame)
ggplot(amsterdamSurface, aes(x=amsterBedroomsRegress, y=amsterSurfaceRegress)) + geom_point() + geom_smo
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
summary(amsterRegres)
```

```
##
## Call:
## lm(formula = amsterSurfaceRegress ~ amsterBedroomsRegress, data = amsterDataFrame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -98.606 -14.158  -3.158  10.152 227.463
##
## Coefficients:
```

```
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.2210     1.4625  -1.519    0.129
## amsterBedroomsRegress  28.6896     0.4899  58.558   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.16 on 2381 degrees of freedom
## Multiple R-squared:  0.5902, Adjusted R-squared:   0.59
## F-statistic:  3429 on 1 and 2381 DF,  p-value: < 2.2e-16
```

```
print(amsterRegres)
```

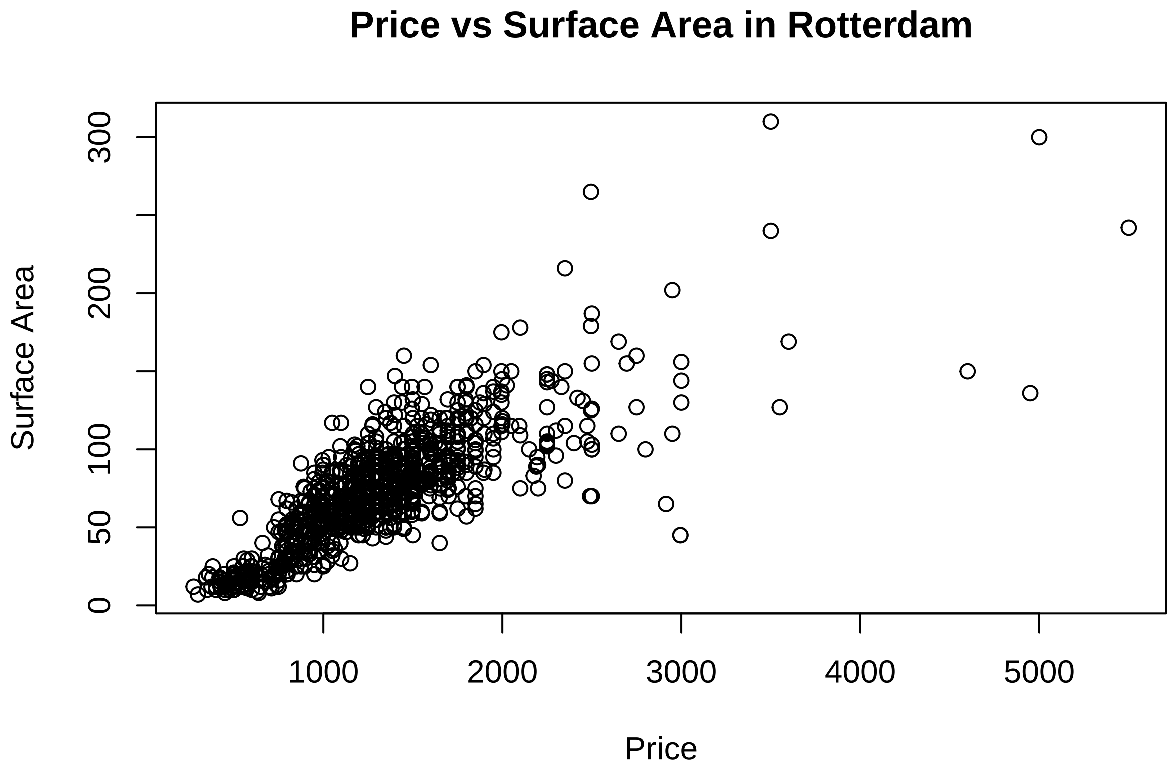
```
##
## Call:
## lm(formula = amsterSurfaceRegress ~ amsterBedroomsRegress, data = amsterDataFrame)
##
## Coefficients:
##           (Intercept)  amsterBedroomsRegress
##                -2.221                 28.690
```

```
#Manual way to predict: surfaceArea = -2.2210 + (28.6869*bedrooms)
amsterPredict<-predict(amsterRegres, list(amsterBedroomsRegress = 3))
amsterPredict #83.84779 m^2
```

```
##        1
## 83.84779
```

For Rotterdam:

```
#(2) the monthly rent for a 125 m2 apartment in Amsterdam and Rotterdam

rotterdamPrice<-Rent %>% filter(city=="Rotterdam")

plot(rotterdamPrice$house_price, rotterdamPrice$surface, xlab="Price", ylab="Surface Area", main="Price
```



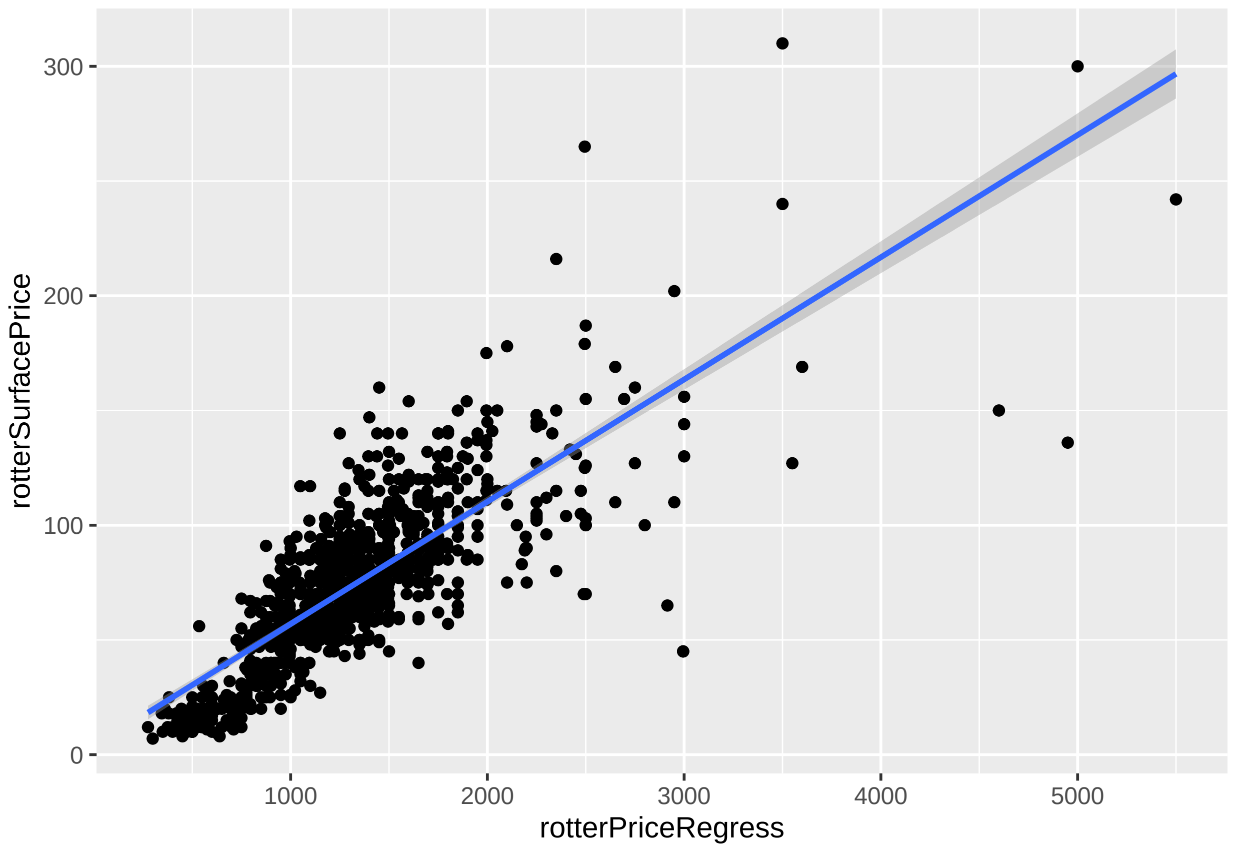
```
cor(rotterdamPrice$surface,rotterdamPrice$house_price)#0.7922818
```

```
## [1] 0.7922818
```

A resulting correlation of 0.7922818 indicated that the variables are dependent meaning that a linear model can be created and regression utilised to predict the monthly rent for a property with a surface area of 125mˆ2 in Rotterdam. A scatter plot including the linear model displays the relationship and the result of the prediction is in the code below.

```
rotterPriceRegress<-rotterdamPrice$house_price
rotterSurfacePrice<-rotterdamPrice$surface
rotterRegresPrice<-lm(rotterPriceRegress~rotterSurfacePrice)
ggplot(rotterdamPrice, aes(x=rotterPriceRegress, y=rotterSurfacePrice)) + geom_point() + geom_smooth(me
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



```
summary(rotterRegresPrice)
```

```
##
## Call:
## lm(formula = rotterPriceRegress ~ rotterSurfacePrice)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1075.46  -175.98   -25.43   118.11  2899.63
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         447.7931    23.5885   18.98   <2e-16 ***
## rotterSurfacePrice  11.7837     0.2865   41.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 327.1 on 1003 degrees of freedom
## Multiple R-squared:  0.6277, Adjusted R-squared:  0.6273
## F-statistic:  1691 on 1 and 1003 DF,  p-value: < 2.2e-16
```

```
print(rotterRegresPrice) #Price = 447.79 + (11.78*surface)
```

```
##
```

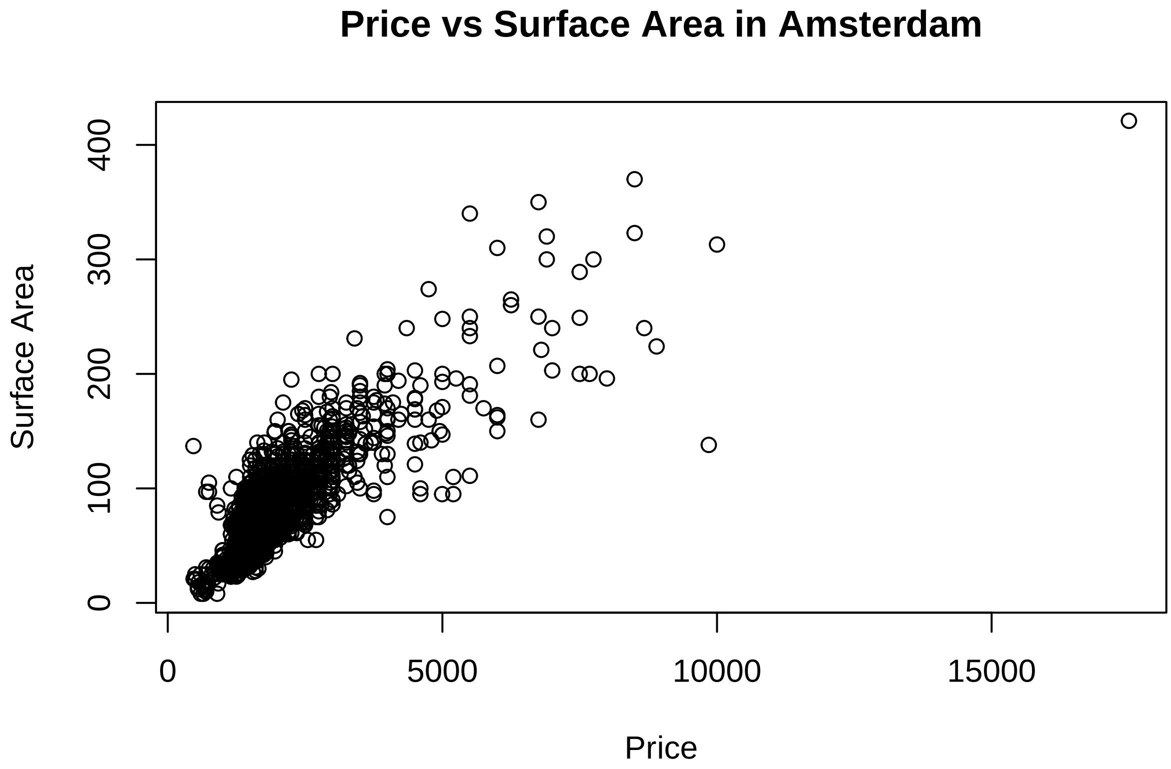
```
## Call:
## lm(formula = rotterPriceRegress ~ rotterSurfacePrice)
##
## Coefficients:
##        (Intercept)  rotterSurfacePrice
##             447.79               11.78
```

```
rotterPredictPrice<-predict(rotterRegresPrice, list(rotterSurfacePrice = 125))
rotterPredictPrice #1920.751
```

```
##        1
## 1920.751
```

For Amsterdam:

```
amsterdamPrice<-Rent %>% filter(city=="Amsterdam")

plot(amsterdamPrice$house_price, amsterdamPrice$surface, xlab="Price", ylab="Surface Area", main="Price
```



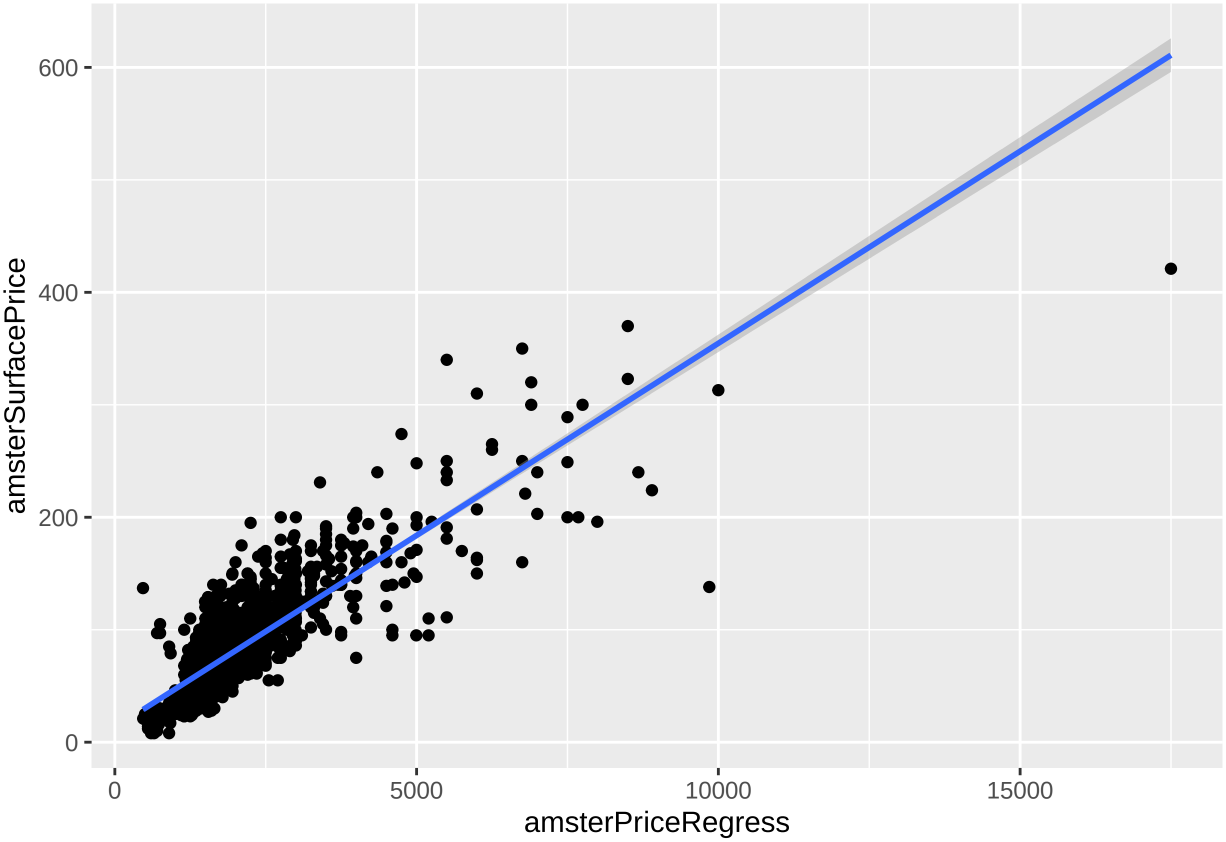
```
cor(amsterdamPrice$surface,amsterdamPrice$house_price)#0.821016
```

```
## [1] 0.821016
```

A resulting correlation of 0.821016 indicated that the variables are dependent meaning that a linear model can be created and regression utilised to predict the monthly rent for a property with a surface area of 125mˆ2 in Amsterdam. A scatter plot including the linear model displays the relationship and the result of the prediction is in the code below.

```
amsterPriceRegress<-amsterdamPrice$house_price
amsterSurfacePrice<-amsterdamPrice$surface
amsterRegresPrice<-lm(amsterPriceRegress~amsterSurfacePrice)
ggplot(amsterdamPrice, aes(x=amsterPriceRegress, y=amsterSurfacePrice)) + geom_point() + geom_smooth(me
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
summary(amsterRegresPrice)
```

```
##
## Call:
## lm(formula = amsterPriceRegress ~ amsterSurfacePrice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2597.9  -241.5    20.2   203.2  8831.6
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        361.7226    24.5368   14.74   <2e-16 ***
## amsterSurfacePrice  19.7309     0.2812   70.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 539.2 on 2381 degrees of freedom
## Multiple R-squared:  0.6741, Adjusted R-squared:  0.6739
## F-statistic:  4924 on 1 and 2381 DF,  p-value: < 2.2e-16
```

```
print(amsterRegresPrice) #Price = 361.7226 + (19.7309*surface)
```

```
##
## Call:
## lm(formula = amsterPriceRegress ~ amsterSurfacePrice)
##
## Coefficients:
##        (Intercept)  amsterSurfacePrice
##             361.72               19.73
```

```
amsterPredictPrice<-predict(amsterRegresPrice, list(amsterSurfacePrice = 125))
amsterPredictPrice #2828.084
```

```
##        1
## 2828.084
```