[redacted]

Our Ref: MGLA010620-3263

24 June 2020

Dear [redacted]

Thank you for your request for information which the Greater London Authority (GLA) received on 31 May 2020. Your request has been dealt with under the Freedom of Information Act 2000.

You asked for:

> The detailed calculations for each ward that sit behind the London Living Rent Ward Benchmark Data published on the Mayor's website in the following location - https://www.london.gov.uk/what-we-do/housing-and-land/improving-private-rented-sector/london-living-rent
>
> This should include the +/- 20% adjustment to reflect neighbourhood level market conditions and the data and full calculations that underpin these adjustments discussed here - https://www.london.gov.uk/what-we-do/housing-and-land/improving-private-rented-sector/london-living-rent#acc-i-47691.

Please find the relevant information attached, in a series of files.

The content of these files is as follows:

- 'London Living rent benchmarks 2020_21.pdf': An explanation of the process followed to produce the 2020/21 London Living Rent benchmarks, with accompanying code in the R programming language.
- '2019.10.29 LLR benchmarks update note.docx': A short summary of the changes in the 2020/21 benchmarks when compared to the previous year.
- 'earnings_data.xlsx': The borough-level earnings data used during the process to estimate average borough-level earnings.
- '2018.09.05 Ward-level benchmark analysis.xlsx': Calculation of ward-level median house prices, as used during the benchmark derivation process. The same ward-level house prices were used in the calculation of the 2020/21 LLR benchmarks, to reduce volatility from year to year.

Please note that we are not able to provide you with one of the datasets used in this process – the Households Below Average Income dataset. The GLA obtains this data as registered users of the UK Data Service, subject to conditions including using it only for specified purposes and not sharing it outside the organisation. Therefore, this information is exempt from disclosure under section 21 of the FOIA (Information accessible via other means).

However, I have included below average household income figures derived from the Households Below Average Income data, as these are the only elements that are used for the calculation of London Living Rent benchmarks.

Year	Median weekly income	Annualised median income
2015/16	£723	£37,705
2016/17	£691	£36,022
2017/18	£737	£38,435
Average	£717	£37,387

If you have any further questions relating to this matter, please contact me, quoting the reference at the top of this letter.

Yours sincerely

[redacted]
Housing Research and Analysis Manager

If you are unhappy with the way the GLA has handled your request, you may complain using the GLA's FOI complaints and internal review procedure, available at:
https://www.london.gov.uk/about-us/governance-and-spending/sharing-our-information/freedom-information

2020/21 London Living Rent benchmarks

[redacted], 29 Oct 2019

This note summarises the 2020/21 London Living Rent benchmarks published by the Greater London Authority.

This update follows the same methodology as in previous years. The procedure is broadly as follows: calculate a third of median gross annual household income in London from the Households Below Average Income dataset, rescale this at borough level using ONS Annual Survey of Hours and Earnings data, and rescale that at ward level using Land Registry data on house prices.

Median household income is calculated on a three-year rolling average to reduce volatility, and this year the estimated figure is £37,387, a 2.5% increase from the previous year.

Borough-level variation around this level is driven by earnings changes. The increases in borough-level average benchmarks cluster around 2.5%, but range from a fall of 3.6% in Kensington and Chelsea to an increase of 5.7% in Lambeth.

The sharp fall in Kensington and Chelsea is partly the result of relatively sparse data in this borough, which resulted in no estimates being reported between 2013 and 2018. The earnings figure used for Kensington and Chelsea is therefore an average of the (higher) 2013 estimate and the (lower) 2018 estimate.

To reduce volatility in the estimates the ward-level ratios have been carried over from the previous year (also the year before that), so the benchmarks change by the same amount in every ward in a given borough.

Excluding the City of London, which is unlikely to see any London Living Rent homes developed, the ward-level monthly benchmarks for two-bedroom homes range from £663 in Boleyn (Newham) to £1,626 in Knightsbridge (Westminster).

London Living Rent benchmarks 2020/21

[redacted]

Introduction

This notebook sets out the analysis required to generate the maximum ward-level rents for London Living Rent properties funded by the Greater London Authority. These maximum rents are published annually on the London Living Rent page (https://www.london.gov.uk/what-we-do/housing-and-land/renting/london-living-rent) on london.gov.uk. The analysis was previously carried out in a combination of SPSS and Excel, but in a point-and-click manner that lacked reproducibility. As long as the input data remains the same, runningg the code in the boxes below should produce the same results. Future updates can change the inputs and the code to obtain different results.

The procedure is broadly as follows: calculate median household income from the Households Below Average Income dataset, rescale this at borough level using Annual Survey of Hours and Earnings data, and rescale *that* at ward level using Land Registry data on house prices.

```
library(haven) # for importing SPSS files
library(tidyverse) # general data wrangling
library(readxl) # for importing Excel data
library(zoo) # for rolling averages
library(spatstat) # for calculating weighted medians
```

Calculate median household income from HBAI data

Let's start by importing the data we want. First, make sure you take note of a very important quirk of the HBAI data. There are two versions of each year's data available: * The single year datasets, in which all the financial values are given in *current* (i.e. not inflation-adjusted) prices * And the harmonised multi-year dataset, in which the financial values are inflation adjusted to *constant* prices (2017/18 in the latest iteration).

The harmonised dataset is very useful for many purposes, but not for what we want to do right now. Because we'll be repeating this analysis every year using current prices, we need to use the single year datasets. The code below imports data for the four most recent years. In a rather clunky fashion, it filters each dataset on import before combining the rows together - I'm sure there's a better way of doing this …

```
# first let's write a function to import the data, since we'll be doing it four times
hbai.import <- function(x){
  read_spss(x) %>% # import the SPSS file
    modify_if(is.labelled, as_factor) %>% # turn the labelled variables into factors for ease of
analysis
    filter(GVTREGN == "London") %>% # filter to London
    select(SERNUM, EGRINCHH, GS_NEWHH, GVTREGN) # select the variables of interest
}

# now we apply that function to our data sources
hh14 <- hbai.import("Q:\\Evidence base\\i. Non-specific sources and reports\\Households below av
erage income\\1994-95 onwards\\UKDA-5828-spss\\spss\\spss24\\hbai1415_g4.sav")
hh15 <- hbai.import("Q:\\Evidence base\\i. Non-specific sources and reports\\Households below av
erage income\\1994-95 onwards\\UKDA-5828-spss\\spss\\spss24\\hbai1516_g4.sav")
hh16 <- hbai.import("Q:\\Evidence base\\i. Non-specific sources and reports\\Households below av
erage income\\1994-95 onwards\\UKDA-5828-spss\\spss\\spss24\\hbai1617_g4.sav")
hh17 <- hbai.import("Q:\\Evidence base\\i. Non-specific sources and reports\\Households below av
erage income\\1994-95 onwards\\UKDA-5828-spss\\spss\\spss24\\h1718.sav")

# now let's bind them together
datasets <- list("2014/15" = hh14, "2015/16" = hh15, "2016/17" = hh16, "2017/18" = hh17)
hh <- bind_rows(datasets, .id = "source")
```

Now we can calculate the median weekly household income by year. The income variable used is 'FRS extended - gross income for the household'. The `averageinc` column is a rolling three-year mean average of the annual medians.

```
income_table <- hh %>%
  group_by(source) %>%
  distinct(SERNUM, .keep_all = TRUE) %>% # keep the first row for each household (denoted by SER
NUM)
  summarise(median_income = spatstat::weighted.median(EGRINCHH, GS_NEWHH)) %>% # median weekly i
ncome
  mutate(averageinc = rollmean(median_income, 3, fill = NA, align = "right"), # rolling 3-year m
ean
         annualisedinc = averageinc * 52.15) # annualised
```

The key figure of interest here is the last rolling average of annual median incomes, so let's extract that.

```
median_income <- income_table$annualisedinc[4]
median_income
```

```
## [1] 37387.24
```

Re-scale income using ASHE data

The next stage is to analyse data on individual earnings at borough level. These are from the Annual Survey of Hours and Earnings, and are available to download from Nomis (http://www.nomisweb.co.uk/). The data we want is the median gross annual earnings for all workers on the basis of both resident and workplace. The procedure for extracting this from Nomis is fairly complicated, so rather than trying to replicate it using R code I have just placed the resulting dataset in the relevant folder, which is accessed as below.

```
earnings_res <- read_excel("Q:\\Intermediate housing\\London Living Rent\\Modelling\\2019 benchm
arks\\earnings_data.xlsx", sheet = 1)
earnings_wp <- read_excel("Q:\\Intermediate housing\\London Living Rent\\Modelling\\2019 benchma
rks\\earnings_data.xlsx", sheet = 2)
```

Next we need to calculate average earnings per borough over the last three years, and fill in the blank against the City of London using data on a workplace basis.

```
earnings_res <- earnings_res %>% rowwise() %>%
  mutate(average = mean(c(`2016`, `2017`, `2018`), na.rm = T))
earnings_wp <- earnings_wp %>% rowwise() %>%
  mutate(average = mean(c(`2016`, `2017`, `2018`), na.rm = T))
earnings_res$average[earnings_res$Borough == "City of London"] <- earnings_wp$average[earnings_w
p$Borough == "City of London"] # fill blank for City of London using workplace-based earnings es
timate

# We also need to dampen the change in the earnings data in Kensington and Chelsea,
# as it is based on just one year of data. The previous estimate was £39,570 in 2013,
# so let's take an average of that and the new 2018 figure.
earnings_res$average[earnings_res$Borough == "Kensington and Chelsea"] <- mean(c(earnings_res$av
erage[earnings_res$Borough == "Kensington and Chelsea"], 39570))
```

Nowe we can re-scale the London-wide gross household income variable using the borough-level earnings data.

```
London_average_earnings <- earnings_res %>% filter(Borough == "London") %>% select(average) %>%
 as.numeric() # extract average London-wide earnings

income_scaled <- earnings_res %>%
  select(Borough, Code, average) %>%
  mutate(income_scaled = average / London_average_earnings * median_income) %>%
  filter(Borough != "London")
```

Re-scale to ward level using house price data

Having derived our borough-level estimates of average household income, the next stage is to disaggregate these to ward level, using data on the ratio of average prices in each ward to average prices for the relevant borough. To reduce year-to-year volatility in the benchmarks, we'll use the same ward-level ratios as last year.

```
ward_ratios <- read_excel("Q:\\Intermediate housing\\London Living Rent\\Modelling\\2019 benchma
rks\\2018.09.06 Ward benchmarks 2018.xlsx", skip = 4) %>% # import file
  select(1:4, 'Ward ratio') # select only the columns we need

ward_benchmarks <- left_join(ward_ratios, income_scaled, by = c("Borough code" = "Code")) %>% #
 join on the borough income data
  select(-Borough, -average, borough_income = income_scaled, ward_ratio = 'Ward ratio') # remove
unneeded columns and rename others

ward_benchmarks <- ward_benchmarks %>%
  mutate(borough_benchmark = borough_income/3/12) %>% # monthly borough benchmark is annual inco
me divided by 3 and then divided by 12
  mutate(ward_benchmark = borough_benchmark * ward_ratio) # ward benchmark is the product of the
borough benchmark and the ward ratio
```

The ward benchmarks we have derived are assumed to set the maximum monthly rent for two-bedroom homes. The final stage is to vary these for different unit sizes.

```
final <- ward_benchmarks %>%
  mutate(`One bedroom` = ward_benchmark * 0.9,
         `Two bedrooms` = ward_benchmark,
         `Three bedrooms` = ward_benchmark * 1.1,
         `Four bedrooms` = ward_benchmark * 1.2,
         `Five bedrooms` = ward_benchmark * 1.3,
         `Six bedrooms` = ward_benchmark * 1.4) %>%
  select(-c(`Borough code`, ward_ratio, borough_income, borough_benchmark, ward_benchmark))
head(final) %>%
  knitr::kable(format.args = list(big.mark = ","), digits = 0)
```

Ward Code	Ward name	Borough name	One bedroom	Two bedrooms	Three bedrooms	Four bedrooms	Five bedrooms	Six bedrooms
E09000001	City of London	City of London	1,714	1,904	2,095	2,285	2,476	2,666
E05000026	Abbey	Barking and Dagenham	800	889	977	1,066	1,155	1,244
E05000027	Alibon	Barking and Dagenham	712	791	871	950	1,029	1,108
E05000028	Becontree	Barking and Dagenham	880	978	1,076	1,173	1,271	1,369
E05000029	Chadwell Heath	Barking and Dagenham	776	863	949	1,035	1,122	1,208
E05000030	Eastbrook	Barking and Dagenham	782	868	955	1,042	1,129	1,216

Earnings Data - Resident

Borough	Code	2014	2015	2016	2017	2018
Barking and Dagenham	E09000002	23,228	23,901	23,601	24,640	25,500
Barnet	E09000003	25,785	26,099	26,000	28,040	28,553
Bexley	E09000004	26,675	25,789	26,754	27,323	27,567
Brent	E09000005	24,763	24,710	24,604	24,645	25,423
Bromley	E09000006	29,934	31,413	31,513	32,827	33,733
Camden	E09000007	32,268	33,429	33,082	32,685	34,531
City of London	E09000001					
Croydon	E09000008	26,335	26,435	27,897	28,163	29,304
Ealing	E09000009	26,560	26,402	27,197	26,847	27,749
Enfield	E09000010	24,739	24,577	25,818	25,270	26,143
Greenwich	E09000011	27,231	28,000	28,860	28,078	30,807
Hackney	E09000012	28,427	28,739	28,847	30,016	29,321
Hammersmith and Fulham	E09000013	32,511	34,438	34,685	35,995	36,016
Haringey	E09000014	26,775	27,082	26,477	28,698	28,334
Harrow	E09000015	25,750	26,753	27,487	27,969	27,758
Havering	E09000016	26,570	27,028	28,106	28,826	27,462
Hillingdon	E09000017	26,824	26,603	27,456	27,085	26,446
Hounslow	E09000018	25,485	25,528	26,797	26,474	27,496
Islington	E09000019	33,598	33,079	31,680	33,219	37,271
Kensington and Chelsea	E09000020					36,340
Kingston upon Thames	E09000021	30,011	30,916	31,434	31,839	32,112
Lambeth	E09000022	28,777	28,772	29,317	31,186	33,465
Lewisham	E09000023	27,537	27,513	29,202	28,092	29,965
Merton	E09000024	28,701	28,865	30,725	30,834	32,331
Newham	E09000025	20,491	22,041	22,501	23,429	24,923
Redbridge	E09000026	27,799	29,331	28,690	28,999	29,623
Richmond upon Thames	E09000027	34,313	35,000	37,704	36,361	36,120
Southwark	E09000028	28,047	29,980	30,000	30,483	32,285
Sutton	E09000029	27,400	26,337	26,679	27,383	28,556
Tower Hamlets	E09000030	30,424	29,552	30,824	32,554	32,435
Waltham Forest	E09000031	23,733	25,620	26,043	26,343	28,628
Wandsworth	E09000032	35,644	35,761	36,538	36,590	38,004
Westminster	E09000033	36,856	37,930	35,534	38,089	42,181
London	E12000007	27,988	28,282	28,832	29,625	30,311

Earnings data - Workplace

Borough	Code	2014	2015	2016	2017	2018
Barking and Dagenham	E09000002	26,003	25,053	24,456	22,175	26,763
Barnet	E09000003	23,979	24,742	23,542	23,897	22,253
Bexley	E09000004	22,347	20,946		23,275	24,761
Brent	E09000005	22,965	25,235	25,259	23,923	22,773
Bromley	E09000006	20,862	23,396	23,708	21,769	23,450
Camden	E09000007	32,816	33,726	34,116	34,496	35,676
City of London	E09000001	52,083	50,505	52,406	53,835	56,532
Croydon	E09000008	24,823	25,072	26,452	26,716	28,226
Ealing	E09000009	24,544	25,049	23,528	25,392	25,433
Enfield	E09000010	21,575	22,692	23,356	23,369	21,902
Greenwich	E09000011	24,950	26,240	25,181	26,537	28,061
Hackney	E09000012	28,003	27,230	28,591	28,738	27,711
Hammersmith and Fulham	E09000013	30,165	31,173	32,224	33,658	37,685
Haringey	E09000014	25,018	24,332	24,964	25,420	24,512
Harrow	E09000015	18,004	17,370	22,699		20,691
Havering	E09000016	21,979	22,983	24,165	23,935	23,905
Hillingdon	E09000017	29,098	27,858	28,325	28,409	26,993
Hounslow	E09000018	29,216	30,096	32,369	30,339	29,976
Islington	E09000019	33,866	33,779	36,281	36,377	36,068
Kensington and Chelsea	E09000020	26,597	27,656	26,728	27,925	29,563
Kingston upon Thames	E09000021	25,726	24,416	25,906	25,826	26,165
Lambeth	E09000022	31,777	32,140	31,579	31,818	32,720
Lewisham	E09000023	23,457	23,621	25,563	26,000	23,462
Merton	E09000024	23,494	23,230	24,213	23,898	25,165
Newham	E09000025	23,526	23,629	24,655	23,448	24,753
Redbridge	E09000026	23,018	23,231	24,045		
Richmond upon Thames	E09000027	24,977	27,445	27,479	27,700	27,813
Southwark	E09000028	31,406	34,065	34,213	34,718	36,778
Sutton	E09000029	21,169	22,037	21,983	22,876	
Tower Hamlets	E09000030	44,026	41,696	40,183	43,668	44,687
Waltham Forest	E09000031	17,980	22,166	22,007	22,745	22,870
Wandsworth	E09000032	24,783	27,021	27,002	28,853	29,356
Westminster	E09000033	34,011	35,571	36,387	38,489	39,717
London	E12000007	30,329	30,769	31,357	32,163	32,976

annual survey of hours and earnings - resident analysis

sex	Total
item name	Median
pay	Annual pay - gross
confidence	Standard error as a percentage of the figure

annual survey of hours and earnings - workplace analysis

sex	Total
item name	Median
pay	Annual pay - gross
confidence	Standard error as a percentage of the figure

These figures are suppressed as statistically unreliable.

- These figures are missing.

Results for 2003 and earlier exclude supplementary surveys. In 2006 there were a number of methodological changes made. For further details goto : http://www.nomisweb.co.uk/articles/341.aspx. Estimates for 2011 and subsequent years use a weighting scheme based on occupations which have been coded according to Standard Occupational Classification (SOC) 2010 that replaced SOC 2000. Therefore care should be taken when making comparisons with earlier years.