MINISTRY OF HOUSING
AND URBAN DEVELOPMENT

Housing Affordability Measure Method

HAM version 1.4

New Zealand Government

Authors
James Kerr, Flynn Valentine-Robertson

Acknowledgements

This research was conducted by the authors while employed by the Ministry of Business, Innovation and Employment (MBIE). This work moved to the Ministry of Housing and Urban Development (HUD) on 1 July 2019.

Feedback on this report has also been received from research colleagues and staff members from MBIE, HUD and Stats NZ.

Published in July 2019

Ministry of Housing and Urban Development
PO Box 82
Wellington 6140

hud.govt.nz

ISBN: 978-0-473-48868-0

Contents

1. Introduction

This paper presents detail of the methodology used to calculate the Housing Affordability Measure (HAM). Specifically, this paper covers version 1.4 of the HAM method, first published in July 2019. HAM comprises two indicators:

1. **HAM Buy**: housing affordability for first home buyers. This indicator addresses whether a household that is currently renting can feasibly afford to own a home.
2. **HAM Rent**: housing affordability for renters. This indicator addresses whether a household that rents can feasibly afford to live in its current accommodation.

The HAM indicators calculate housing affordability based on the proportion of households spending more than 30% of their income on housing costs:

1. **HAM Buy** measures the proportion of renting households who would spend more than 30% of their income on housing costs if they bought a lower quartile-valued dwelling in their territorial authority,[1] based on their income in the year ending that quarter and an estimate of what their housing costs would be on purchasing this dwelling.
2. **HAM Rent** measures the proportion of renting households who spend more than 30% of their income on housing costs, based on their income and rent in their current home in the year ending that quarter.

For users interested in alternative measures of housing affordability, we have also released two alternate HAM series. These series report the share of households that have less income minus housing costs than the median household and income minus housing costs than the 10th percentile household, adjusting for inflation and household size. Further details of these measures can be found in sections 3.6 and 3.7.

[1] Or ward area, within Auckland.

2. Data used to construct HAM

2.1. Data sources

The majority of the data used to create the HAM is sourced from Stats NZ's Integrated Data Infrastructure (IDI). The IDI contains anonymised microdata about people and households. This data is supplied by a range of government agencies and non-governmental organisations, and also includes data from Stats NZ surveys, including Census 2013.[2]

The IDI's data on New Zealanders is extremely comprehensive, making it possible to examine the state of housing affordability in New Zealand in fine detail without the sampling errors that come from using survey data or the jumps in a time series caused by the low frequency of censuses.

In addition to the IDI the following data was used to create the HAM:

- consumers price index (CPI)
- summary rateable value and E-valuer data supplied by CoreLogic New Zealand
- interest rate data from the Reserve Bank of New Zealand.

2.2. Frequency of data updates

The HAM is presented as a quarterly series starting in March 2003. At the time of publication (July 2019) the latest quarter of data available is December 2018. The reason for the lag is that it takes time for all the data required to produce the HAM to be reported to Stats NZ and incorporated into the IDI.

At the time of publication, the IDI is being refreshed every six months. For this reason, HAM releases are scheduled six-monthly with two new quarters of data being produced per release.

2.3. The reference date

Every quarter of data is produced relative to a reference date. This date is the last day of that quarter. For example, the reference date for the HAM values for March 2013 is 31 March 2013. The ways in which the reference dates are used are explained in subsequent sections.

2.4. Data used to calculate incomes

The HAM calculates individual personal income using the following sources in the IDI:

- Inland Revenue Department (IRD): Employee Monthly Schedule (EMS) incomes. These are monthly income reports from employers about their employees. EMS incomes include taxable benefits supplied by the Ministry of Social Development (MSD) to the IRD.
- IRD: self-employment income (annual returns based on the IR4S, IR3 and IR20 income tax returns).
- MSD: non-taxable benefit income.

[2] http://archive.stats.govt.nz/browse_for_stats/snapshots-of-nz/integrated-data-infrastructure/idi-data.aspx

All income from these sources is collected for the year ending on the relevant reference date (see section 2.3 above). Note that because data for tax payable on self-employment income is not available in the IDI, all incomes are calculated before tax.

On 31 March 2013 there were 1,528,683 households with some reported income during the year to that reference date (the closest to the date of the most recent census) before applying any filters to the data.

2.5. Data used to define the HAM population

The IDI's address data, in conjunction with its 'Personal details' dataset, is used to determine the number of people who are alive in New Zealand and have a known address on the reference date for each quarter. The 'Personal details' dataset is also used to calculate individuals' ages on the quarter's reference date: this data is used for equivalising household income (see section 3.4).

On 31 March 2013, this IDI data produced a total population of 5,465,886 people, relative to a census population of 4,125,894 on 5 March 2013. The variance between IDI and census populations suggests the IDI is over-counting people in New Zealand, presumably by treating one person as several people due to discrepancies between administrative data sources. To address this issue, we employ several filters to better reconcile IDI populations and Census populations.

2.5.1. Appearance in administrative data

This filter was developed by Gibb, Bycroft and Matheson-Dunning (2016)[3] as part of a project to reconcile the information in the IDI with that gathered by the census.

The filter used by Gibb et al. requires an individual to meet one of the following two criteria:

1) Have had an interaction with one of the following agencies in the past year:
    - Inland Revenue Department
    - Ministry of Social Development
    - Ministry of Health
    - Ministry of Education
    - Accident Compensation Corporation

2) Be a child under five years old.

This is still higher than the census, but reduces the variance considerably. Note that this filter should have no effect on household income, as any person reporting an income in the past year meets criterion 1 by definition.

2.5.2. Accounting for emigration

[3] Gibb, S., Bycroft, C., & Matheson-Dunning, N. (2016). *Identifying the New Zealand resident population in the Integrated Data Infrastructure (IDI)*. Retrieved from http://archive.stats.govt.nz/methods/research-papers/topss/identifying-nz-resident-pop-in-idi.aspx

The IDI is only able to assign New Zealand addresses to people, and an address in the IDI is only invalidated when a person reports a new address in New Zealand. This leads to people who have left New Zealand still being counted as resident at their last New Zealand address.

HAM accounts for the people who have left New Zealand by looking at individuals who were not in the country on the reference date, and then checking to see if they had returned within 30 days of departing. If not, they are excluded from the population.

2.5.3. Addressing the effect of lags in address notifications

The address information in the IDI combines data from a range of administrative sources, but people can end up informing government agencies of an address change at different times. One difficulty this can present is that one household may move out of an address but inform government agencies of the change after the household that is moving in does. This would create a time window where the IDI would count both households as resident – potentially doubling the household's apparent income.

To prevent address notification lags from overstating the number of large and high-income households, HAM excludes all individuals from the population who changed address within 30 days of the reference date (before or after). This filter is not applied to address changes caused by Censuses, as those changes are all coming from a single consistent source.

2.6. Data used to define households

Using the address data in the IDI, all individuals in the filtered HAM population are assigned into households based on their address on the quarter's reference date. To improve data quality, the following 'households' are excluded:

- Households composed of more than 15 residents, to account for addresses that are not truly residences. For example, some people register their accountant's office with the IRD and this can make the accountant's office appear to be a residence with a large number of people living at it.
- Households with no residents aged 15 years or older. This is because a household without a resident 15 years or older is unlikely to be a real household; more plausibly, the household has moved elsewhere, but some of the address information for individuals in the household has not been fully updated.
- Households with zero or negative income. While it is possible for a household to have no or negative income for a short time, such incomes cannot be a reflection of the household's standard of living. It is likely that households with zero or negative income either experienced a temporary loss of income or have income sources that are difficult for the IDI to detect.

Note that this approach to household construction, while necessary due to the nature of the data relies on the assumption that the household will retain its composition for the near future. This presents an issue for some households in the potential first home buying population, as there are certain cases where this will not hold true, such as if:

- The household would split up or lose members if it became owner-occupying (eg a group of flatmates), or

- The household would add members if it became owner-occupying (eg buying a dwelling to make room for relatives).

In these cases, HAM Buy will not correctly identify the level of housing affordability that the household would experience in purchasing a dwelling.
The household data produced for HAM is used unweighted, as earlier investigations into HAM version 1.0 suggested the HAM data is representative at the territorial authority and ward level.

2.7. Data used to define the renting population

The HAM population is composed of renting households. To gather data on these, we use the tenancy bond database, which the Ministry of Business, Innovation and Employment (MBIE) permits to be incorporated into the IDI. Under the *Residential Tenancies Act 1986*, all landlords who require a bond of their tenants must register this bond with MBIE. The bond form provides information on the address and weekly rent of the tenancy, as well as indicating when a property became tenanted.

The tenancy bond data identifies 417,290 tenanted properties on 31 March 2013, which compares with a 2013 census count of 453,132 renting households. The census number is larger in part because not all tenancies end up having a bond associated with them. If the tenancy is for a short period or the landlord does not require a bond then the landlord would not be required to lodge a bond with MBIE.

Properties with no registered tenancy are treated as not being rentals and are excluded from the population. While we have no direct data on the excluded households, our validation of HAM data against Household Economic Survey (HES) data in Annex 4 suggests the potential bias in HAM Rent and HAM Buy is quite small.

More discussion on the limitations and validation of the rental data can be found in Section 4.

2.8. Data used to calculate housing costs for HAM Buy

Data from CoreLogic New Zealand, as well as Stats NZ's HES, is used to calculate the housing costs of the 'modest dwelling' (the lower quartile value) used in constructing HAM Buy. CoreLogic New Zealand provides information on the rateable value of residences in addition to their own E-valuer estimate of market values.[4]

CoreLogic collects data on rateable values and sales prices through a relationship with territorial authorities, and uses secondary sources to fill in the gaps in their primary data sources. Each address is assigned to a geographical area (either a territorial authority or, in Auckland, a ward area) to act as the basis for estimating the cost of a modest dwelling.

Households are assigned one of three dwelling size categories based on the number of bedrooms they have in the property they are currently renting:

- small (1-2 bedrooms)

[4] CoreLogic's E-valuer uses the sales prices of similar dwellings sold in a given time period to estimate a market value for all dwellings. The result is similar to a sales price series, but without the variation caused by the specific characteristics of the dwellings sold in each period. For example, if a number of expensive houses get sold in a given quarter, this would bias prices up for that quarter. Using E-valuer data removes this effect.

- medium (3 bedrooms), and
- large (4 or more bedrooms).

These categories were selected because they were the finest gradation of dwelling size that was stable. Separating 1 bedroom or 5+ bedroom dwellings into their own categories resulted in extreme swings in values for some of the smaller territorial authorities.

The “modest home” value each household is assigned depends on their geographical area and bedrooms category (so, for example, a household renting a 3-bedroom dwelling in Wellington is assigned the lower quartile value for 3-bedroom Wellington dwellings in the quarter ending on the reference date). Households with unknown number of bedrooms or geographical area are excluded.

For HAM Buy, housing cost data is estimated by comparing CoreLogic’s E-valuer and rateable value data to HES data. This data is used to calculate the average expenditure on rates per dollar of capital value, and insurance expenditure per dollar of E-valuer market value.

Data on mortgage interest rates is sourced from the Reserve Bank of New Zealand.[5]

2.9. Households with negative residual income

Households with negative residual income are excluded from the HAM population. In practice, it is not possible for a household to sustainably spend more than their income on housing, implying that households in this state have income sources not visible to the IDI or suffered some temporary setback, such as suffering a business loss that year. In either case, these households represent cases where a household’s actual ability to afford housing is not properly being represented by the data.

[5] 5-year rate from series hB21 - New special residential mortgage interest rates.

3. Calculating affordability

3.1. Differences between the measures

3.1.1. Differences between HAM Buy and HAM Rent

Both indicators examine the same ultimate population: all households who are renting on the quarter's reference date, as measured by having an active bond lodged against the property in which the household lives (less those who are excluded, as described in section 2).

The differences between the indicators reflect the fact that each indicator examines a different aspect of housing affordability. HAM Rent is concerned with a household's current housing circumstances, while HAM Buy is concerned with what a household's circumstances would be if they became home-owners.

This means that while both measures use the same income, they use different housing costs – HAM Rent uses the existing household's rental housing costs, while HAM Buy considers the housing costs the household might have if they bought a first home.

3.1.2. Differences between HAM Percent, HAM Median, HAM 10th Percentile and HAM Index

Each of these measures compares income and housing costs but does so in different ways, to permit different comparisons.

HAM Percent considers the ratio of housing costs to incomes – this means that it will tend to stay constant if housing costs grow at the same rate as incomes. This measure is better suited to a concept of affordability related to the share of a household's budget being devoted to housing.

HAM Median and HAM 10th Percentile are based on the income a household has left over after paying for housing costs. They will tend to show improving affordability if housing costs grow at the same rate as incomes, as the amount of money people have left over will also be growing. These measures are better suited to a concept of affordability related to the non-housing standard of living people can afford to sustain.

HAM Index is an index based on HAM Median, scaled so that that national HAM Rent Index value in March 2003 is 1000. All other index values are proportional to this figure, meaning that HAM index can be compared between regions and between Rent and Buy, as well as over time.

3.2. Calculating household incomes

All individual incomes at an address are combined to form household income.

Household income (*HI*) is calculated as follows:

$$HI_{p,t} = \sum_{i} I_{i,p,t}$$

Where:
I = amount of individual income (before tax)

i = source of individual income earned
p = address
t = time (income sources from the past 12 months are included)

3.3. Calculating housing costs

3.3.1. Calculating housing costs for first home buyers

The hypothetical housing costs (HCB) for each household are calculated based on the following formula:

$$HCB_{b,a,t} = Ins_{b,a,t} + \; LR_{b,a,t} + MP_{b,a,t}$$

Where:
Ins = insurance
LR = local body rates
MP = mortgage payments
b = number of bedrooms the household's existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t = time, the quarter ending the reference date

Ins and *LR* are calculated as ratios of, respectively, the dwelling price and capital value of a modest dwelling:

$$Ins_{b,a,t} = IR \times LQ(P_{b,a,t})$$
$$LR_{b,a,t} = RR \times LQ(CV_{b,a,T})$$

Where:
IR = insurance ratio (average insurance cost as a fraction of dwelling price, based on CoreLogic E-valuer estimates)
LQ(P) = lower quartile market value
RR = rates ratio (average rates cost as a fraction of capital value)
LQ(CV) = lower quartile capital value (rateable value of land and building/s)
b = number of bedrooms the household's existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t=time, the quarter ending the reference date
T = time, the year within the most recent 3-year period containing the largest number of rates revisions

Most territorial authorities only revise rates once every three years. However, this cycle is different for different territorial authorities. Further complicating the rateable value data is the fact that some properties have their rates revised outside the regular cycle – this can lead to some rateable value data being available for a given area, but with a much lower count than usual. For this reason the rateable value data is divided into three-year blocks, with the year containing the largest count of distinct rateable values in that area taken as the representative year for that 3-year block.

IR and RR are calculated by comparing the expenditures of households in the HES with the HES dwellings' sales prices and capital values of dwellings.

The reason MP is calculated based on the entire price of the property (i.e. that no allowance is made for a deposit) is that HAM Buy needs to account for the entire financial loss associated with purchasing a home. This includes both the ongoing financial cost of servicing a mortgage, and the loss of the deposit, which needed to be saved over time. The formula used to convert a one-off loss of money into an annual cost over time is exactly the same as the formula used to calculate mortgage payments, which means that the following formula holds regardless of how large or small the deposit is:

$$MP_{b,a,t} = \frac{LQ(P_{b,a,t}) * r_t}{1 - (1 + r_t)^{-30}}$$

Where:
LQ(P) = lower quartile dwelling price
r = Reserve Bank's 5-year fixed rate new mortgage interest rate
b = number of bedrooms the household's existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t = time, the quarter ending in the reference date
30 = term of the mortgage in years

Based on data from CoreLogic on the behaviour of first home buyers, the definition of a modest dwelling is set as the lower quartile of dwellings in the same bedroom class in the area (territorial authority, or ward for Auckland) in which the household resides.

3.3.2. Calculating housing costs for renters

The housing costs (HCR) for each household in rental tenure are calculated as follows:

$$HCR_{p,t} = R_{p,t} \times 52$$

Where:
R = weekly rent from the tenancy bond database (taken from the last bond lodged against that property (at time *t*) that is still active).
p = address
t = time, the quarter ending in the reference date
52 = number of weeks per year

3.4. Calculating equivalisation factor

The equivalisation factor (HEF) is used to calculate Equivalised Residual Income for HAM Median and HAM 10th Percentile.

$$HEF_{p,t} = 1 + 0.5\ (A_{p,t} - 1) + 0.3\ C_{p,t}$$

Where:
A = number of people in the household aged 14 years and older
C = number of people in the household aged under 14 years
p = address
t = time, the quarter ending in the reference date

This formula is the Eurostat version of the OECD equivalisation formula.[6]

3.5. Calculating HAM Percent

HAM Percent Buy (HPB) and HAM Percent Rent (HPR) are calculated for each household by comparing their housing costs to their household income. Each household is determined to have either spent more or less than 30% of their income on housing. The reported HAM Percent statistics are the proportions of households who have spent more than 30% of their income on housing costs. 30% was selected due to it being a commonly used threshold for affordability, both within New Zealand and overseas.

3.5.1. Calculating HAM Percent Buy

HPB is "More" if:

$$\frac{HCB_{b,a,t} - ASI_{p,t}}{HI_{p,t} - ASI_{p,t}} > 0.3$$

Else HPB is "Less".

Where:

HCB = cost of buying a house (see section 3.3.1)
ASI = income received from the Accommodation Supplement
HI = household Income (see section 3.2)
b = number of bedrooms the household's existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t = time, the quarter ending in the reference date
p = address

Note that for HAM percent, income from the Accommodation Supplement is subtracted from housing costs instead of being added to income. This is because the Accommodation Supplement is specifically intended to defray housing costs – this creates parity in the HAM calculations between the Accommodation Supplement and other forms of housing support, such as the Income-Related Rent scheme.

3.5.2. Calculating HAM Percent Rent

HPR is "More" if:

$$\frac{HCR_{p,t} - ASI_{p,t}}{HI_{p,t} - ASI_{p,t}} > 0.3$$

Else HPR is "Less".

Where:

[6] https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Equivalised_disposable_income

HCR is Cost of renting a house (see section 3.3.2)
ASI is the income received from the Accommodation Supplement
HI is household Income (see section 3.2)
p = address
t = time, the quarter ending in the reference date

Note that for HAM percent, Income from the Accommodation Supplement is subtracted from housing costs instead of being added to income (see section 3.5.1).

3.6. Calculating HAM Median

HAM Median Buy (HMB) and HAM Median Rent (HMR) are calculated for each household by subtracting their housing costs from their household income. Each household is determined to either have more or less than the median average income after housing costs. The reported HAM Median statistics are the proportions of households who have less than average income after housing costs.

3.6.1. Calculating average income minus housing costs (ARI)

Average income minus housing costs is calculated using equivalised residual incomes from the 2013 HES[7]. The equivalisation formula used is the same modified OECD formula described in section 3.4. The housing cost share is the average share of household income spent on housing-related expenses (rent, mortgage payments, body corporate fees, rates and insurance), calculated from 2013 HES data. The normal HES weightings are used to determine the proper median.

Average income minus housing costs is **$662 per week** (in 2013 dollars). Note that the benchmark is only directly applicable to one-person households. A larger household needs a correspondingly higher residual income to be considered to have above-average affordability.

This value is inflation-adjusted using CPI data for each quarter in the HAM time series. This is necessary because the equivalised residual incomes are in nominal (face value) terms.

The definition of average income minus housing costs should be rebased from time to time. This approach is used in a number of other statistical collections, such as the production measure of constant price gross domestic product and the CPI.

3.6.2. Calculating HAM Median Buy

HMB is “Below” if:

$$\frac{HI_{p,t} - HCB_{b,a,t}}{HEF_{p,t}} < ARI_t$$

Else HMB is “Above”.

[7] Including both home-owning and renting households.

Where:

HI = household Income (see section 3.2)
HCB = cost of buying a house (see section 3.3.1)
ARI = average income minus housing costs (see section 3.6.1)
HEF = Household Equivalisation Factor (see section 3.4)
p = address
b = number of bedrooms the household's existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t = time, the quarter ending in the reference date

3.6.3. Calculating HAM Median Rent

HMR is "Below" if:

$$\frac{HI_{p,t} - HCR_{p,t}}{HEF_{p,t}} < ARI_t$$

Else HMR is "Above".

Where:

HI = household Income (see section 3.2)
HCR = rental costs of the household (see section 3.3.2)
ARI = average income minus housing costs (see section 3.6.1)
HEF = Household Equivalisation Factor (see section 3.4)
p = address
t = time, the quarter ending in the reference date

3.7. Calculating HAM 10th Percentile

HAM 10th Percentile Buy (HTB) and HAM 10th Percentile Rent (HTR) are calculated in a similar way to HAM Median, but with a different residual income threshold.

3.7.1. Calculating 10th percentile income minus housing costs

10th percentile income minus housing costs is calculated using the same HES data as average income minus housing costs (see section 3.6.1). The only difference is that the 10th percentile is taken instead of the median.

10th percentile income minus housing costs is $215 per week (in 2013 dollars). The same caveats apply to this value as for average income minus housing costs.

3.7.2. Calculating HAM 10th Percentile Buy

HTB is "Below" if:

$$\frac{HI_{p,t} - HCB_{b,a,t}}{HEF_{p,t}} < TRI_t$$

Else HTB is “Above”.

Where:

HI = household Income (see section 3.2)
HCB = cost of buying a house (see section 3.3.1)
TRI = 10th percentile income minus housing costs (see section 3.7.1)
HEF = Household Equivalisation Factor (see section 3.4)
p = address
b = number of bedrooms the household’s existing property has (either 1-2, 3 or 4+)
a = area (ward for addresses in Auckland; territorial authority for other addresses)
t = time, the quarter ending in the reference date

3.7.3. Calculating HAM 10th Percentile Rent

HTR is “Below” if:

$$\frac{HI_{p,t} - HCR_{p,t}}{HEF_{p,t}} < TRI_t$$

Else HTR is “Above”.

Where:

HI = household Income (see section 3.2)
HCR = rental costs of the household (see section 3.3.2)
TRI = 10th percentile income minus housing costs (see section 3.7.1)
HEF = Household Equivalisation Factor (see section 3.4)
p = address
t = time, the quarter ending in the reference date

4. Limitations of HAM version 1.4

4.1. The effect of credit constraints on HAM Buy

There are two financial barriers to buying an asset (such as a home) that is too expensive to be bought with cash on hand:

- The financial cost of purchasing the asset, whether this is money the purchaser has saved up, or the cost of servicing debt repayments (Mortgage Cost).
- The restrictions on how much debt someone can borrow to pay for whatever part of the asset's cost they have not saved up for (Capital Constraints).

While HAM accounts for the first of these barriers, it does not account for the second. Someone with enough income to afford a 100% mortgage on a modest home might not have enough saving to provide the deposit banks require. This would leave them with "affordable housing" by the definitions of HAM, but still be unable to purchase a modest home in practice.

While capital constraints are related to housing affordability they are logically distinct, as they depend as much on prudential lending standards as house prices and incomes. A research project would be required to relate income and savings together, and then compare a household's estimated savings with the deposit they would require to obtain a mortgage for a modest home in their area.

4.2. The effect of household composition changes on HAM Buy

HAM Buy assumes that each renting household would retain its present composition if it becomes owner-occupying. In cases where this is not true, either because:

- the household would split up or lose members if it became owner-occupying (eg a group of flatmates), or
- the household would add members if it became owner-occupying (eg buying a dwelling to make room for relatives)

then HAM Buy will not correctly identify the level of housing stress that the household would experience in purchasing a dwelling.

4.3. The effect of willingness to move on HAM Buy

HAM Buy assumes that a household will buy a dwelling in the same area as they currently live (territorial authority, or ward area for Auckland) and be of the same size (as measured by number of bedrooms). If a household is willing and able to move to a smaller dwelling or to a lower-cost area to buy a dwelling, HAM Buy will overestimate potential housing stress from buying a dwelling.

4.4. Households with unusually high non-housing expenses

Some households will inevitably have higher-than-usual non-housing expenses for a household of their size, including:

- student loan repayments
- child support for children not residing in the household
- higher-than-usual transport costs due to long commutes
- costs due to health or disability issues.

Households with higher-than-usual non-housing costs may be experiencing more constrained cash flow than their housing stress category implies.

4.5. Missing tenancy data

Rental data for the HAM measures depends on Residential Tenancy Bond data. The Residential Tenancies Act requires that all tenancy bonds are lodged with MBIE, but there is no legal requirement for a landlord to require a bond of their tenants. This means that while the Tenancy Bond Database's coverage of tenancies is extensive, it is not exhaustive.

Households that do not have a bond lodged against them are presumed to be owner-occupied. This may result in a misclassification of some renting households as owner-occupied, resulting in the count of renting households using HAM data to be an undercount of actual renting households. For this reason we do not recommend using the HAM data to estimate the number or proportion of households that are renting.

The tenancy bond data is incomplete in other ways:

- Some bonds could not be address-coded due to address quality issues. These bonds needed to be excluded from HAM as they could not be joined to any other data in the IDI.
- Some bonds did not have a reported rent value. HAM Rent could not be computed for these households.
- Some bonds did not have a number of bedrooms recorded for the household. HAM Buy could not be computed for these households.

Overall there were 264,201 households included in HAM Rent in March 2013. These were households with coded addresses, a non-zero income and a reported rent value.

There were 247,587 households included in HAM Buy in March 2013. These were households with coded addresses, a non-zero income and a reported number of bedrooms.

By contrast, Census 2013 reports 453,135 households paying rent. This means that HAM Rent has 58% coverage and HAM Buy has 57% coverage.

Appendix A and Appendix B discuss the representativeness of the HAM data by comparing it to Census 2013 and HES.

5. Response to StatsNZ Review of HAM version 1.0

In June 2017, Stats NZ completed a review of the originally-published version of HAM (HAM version 1.0). The review made several recommendations for further improving HAM. Each sub-heading in this section is one of the Stats NZ recommendations, with MBIE's response as the text below.

5.1. MBIE to revise the Methodology paper based on Stats NZ review comments provided to MBIE in June 2017

This paper is a revision of the original HAM method paper from June 2017.

5.2. MBIE to undertake further work on improving communication of HAM to improve the public's understanding

MBIE has improved the communication of HAM in several ways:

1. The publication of infographics explaining how HAM Buy and HAM Rent are calculated.
2. The development of HAM Percent, an alternative affordability measure that is easier to explain to a non-technical audience.

5.3. Provide comparative information alongside HAM

MBIE has created a household income series that uses the HAM data to construct mean, median and quintile household incomes for renters and for all households.

Housing Costs can be derived from the summary housing data used as an input to HAM, and from Tenancy Bond Data.

While comparisons between HAM and most affordability measures are difficult due to the differences in their construction, a comparison between HAM and HES is included in Appendix B.

5.4. Building on other work on poverty measurement, develop a benchmark measure of absolute affordability for use in the housing affordability measure

Defining poverty in an absolute sense is beyond the scope of MBIE's responsibility. It would be inappropriate for MBIE to attempt to redefining poverty measurement independently.

5.5. Improve collection of Tenancy Bond addresses

Work to improve addresses is ongoing, including using New Zealand Post GEO-PAF coding for Tenancy Bond Data.

5.6. Base extraction of the IDI resident population on the IDI-ERP code available in the IDI Wiki.

The ERP approach to border movements proved to be problematic for HAM as its rules regarding border movement would engender a large lag in the HAM data (up to 15 months).

5.7. Update personal income to include investment income (available in the IDI from August 2017)

MBIE has investigated the investment income data, and all available income data from the IDI is included.

5.8. Improved methodology for calculating non-taxable government transfers (Stats NZ, MSD)

Non-taxable government transfers are now included in the HAM income data.

5.9. Review interest rate used in HAM Buy housing costs

HAM now uses a different interest rate to calculate HAM Buy. This rate was selected in consultation with the Reserve Bank, who first identified the issue with the interest rate we were using.

5.10. Improve quality of underlying data used to construct households

This is a good idea, but beyond the scope of the HAM project.

5.11. Undertake methodological work to develop quality measures for the housing affordability measure

HAM now has a range of diagnostic measures to assist the analysts producing the measure to determine quality:

- Stability checking to determine whether the HAM back-series has changed since the last IDI refresh.
- Cohort testing to determine whether changes in HAM are due to changes in household composition or changes in incomes and housing costs.
- Row count and data loss calculations at various stages of the HAM calculation process.
- Final count data for each breakdown.

5.12. Investigate alternative approaches to deriving housing costs for HAM Buy

As part of HAM version 1.4, the method of calculating housing costs for HAM Buy now account for the number of bedrooms the household has in their current home. This means that HAM Buy no longer assumes housing costs scale with household size in the same way non-housing costs do.

5.13. Consider moving to an annual series for the HAM measures

This would reduce the information value of HAM as enough changes can occur on a quarter-to-quarter basis that the public would be less well-informed by an annual series.

5.14. Develop methods for calculating disposable household income for the housing affordability measures

The reason HAM is before-tax is due to limitations in the self-employment income. Also, HES uses pre-tax income and as HES is used to construct the income thresholds for HAM Median and HAM 10th Percentile, using pre-tax income is more consistent.

5.15. Develop ways to include ability to save for a deposit in the HAM Buy measure

The effect of deposit requirements on the ability of first-home buyers to buy a home is not precisely a question of affordability, though it is related. Understanding the effect of deposit requirements on the ability to buy a first home is a worthwhile extension to Housing affordability research, but would require investigating the relationship between income and capacity to save.

5.16. Construct housing affordability measures for low income households by geographic area

The existing HAM measures are able to target low-income households reasonably well, but further investigation of HAM for vulnerable populations, including low-income households, would be a valuable extension to HAM.

5.17. Further work on housing affordability for home owners should consider use of small domain models to produce territorial authority measures from the Household Economic Survey

The development of HAM Own (a HAM measure for Owner-Occupiers in their current home) could not be completed within the scope of MBIE’s responsibility for the HAM project. While HAM Buy and HAM Rent stand well enough on their own as affordability measures, it would be beneficial to develop a robust owner-occupier affordability measure. There are, however, some substantial obstacles to producing such a measure including:

- Identifying which households are owner-occupiers through the IDI is difficult (see 4.5).
- The amount of equity a household may have in their home can vary a lot – initial efforts to model the relationship between income, house price and mortgage expenses failed.

Appendix A: Constructing Households with the IDI

The IDI represents a huge opportunity for New Zealand income data. Its size means that it can supply information on incomes for small groups of the population, something the existing survey-based series cannot do. Also, unlike the Census, the IDI can provide a high-frequency data series, and one that is not limited by the income bands used in the Census.

However, there are difficulties in using the IDI to compose household incomes. The IDI address information is built up from multiple administrative sources and any errors or omissions in that data can lead to missing people, or people applied to the wrong addresses. For the IDI to be useful as a data source it has to be able to provide a reasonable depiction of the distribution of household incomes.

To ensure the IDI income and household composition data is suitable for producing household incomes, we tested IDI summary data for household incomes and household composition against Census 2013 data. This helps test the validity of the IDI data, and the filters applied as part of the HAM method, for producing income and household composition statistics.

Income sources

The following income sources are captured as part of HAM:

- Wage, salary and taxable benefit income, as it appears in the IRD EMS table.
- Self-employment income.
- Non-taxable benefit income.

Method

The individual income for each person is collected based on data source for the year up to and including the reference quarter. The income data is then aggregated to the person level. The following filters are then applied to people to adjust for the tendency of the IDI to duplicate people:

- Any person over the age of five who has not interacted with one of the following administrative sources is excluded:
  - Ministry of Education
  - Ministry of Health
  - Accident Compensation Corporation
  - Ministry of Social Development
  - Inland Revenue
- Any person who has left the country on the last day of the quarter and does not return within 30 days of leaving is excluded, as they are assumed to have left long-term. This is to account for people who are no longer resident in New Zealand.
- Any person who changed address within 30 days either side of the quarter’s end is excluded. This is to account for cases where lags in address changes cause some households to end up doubled-up in one house. Note that this filter is not applied to address changes caused by Censuses, as those changes are all coming from a single consistent source.

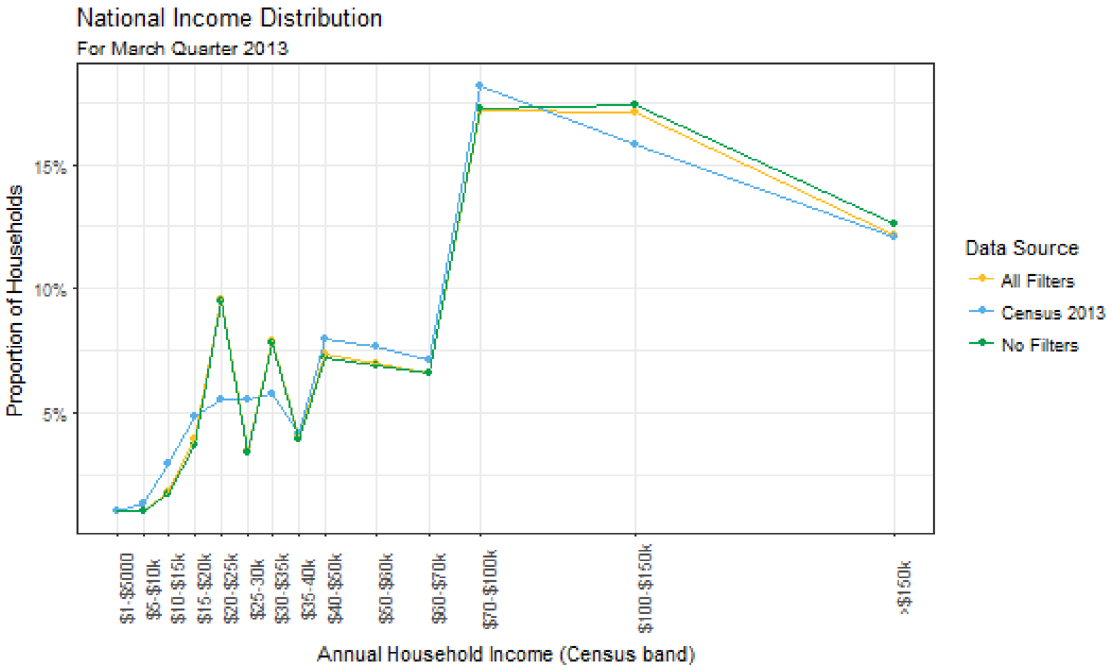
Validating IDI income data

It is difficult to properly validate the IDI data as there is a shortage of high-quality income data sets:

- The Household Economic Survey (HES) has too small a sample to be used to evaluate income distributions, especially when evaluating sub-national distributions.
- The Household Labour Force Survey (HLFS) only captures income from paid employment (ie wages and salaries) and government transfers, making it unsuitable as a validation tool for the more comprehensive IDI Income. Also, while its sample size is larger than HES, it is still of limited use for smaller geographical areas.
- Census 2013 only exists for a single point in time, and its income data is reported in bands.

Of these sources, Census 2013 is the only one that is large enough to allow for proper evaluation of IDI income distributions, so it is the data source we have used here.

Comparison against Census 2013



The added filters incrementally reduce the proportion of very high income households, but other than that the filters do not affect household income much.

There are two main variances between HAM 1.4 and Census 2013:

- Fewer households with incomes under $20,000 in the HAM Data
- More households with incomes between $20,000-$25,000 and $30,000-$35,000 in the HAM data

The likely explanation for the difference in low income households between HAM 1.4 and Census 2013 can be found in the MSD benefit rates[8] that applied between April 2012 and April 2013.

[8] https://www.workandincome.govt.nz/products/benefit-rates/benefit-rates-april-2012.html#null

The lowest income a benefit-receiving household could be expected to have is a one-person household with a single adult under 20 who is unemployed and in an HNZC house (thereby being ineligible for the Accommodation Supplement). Such a household would have received $190.84 per week, or $9,923.68 per year, with the majority of one-person households receiving more than $10,000 per year due to other benefits such as the Accommodation Supplement.

Similarly, the least a couple could be receiving would be $381.68 per week or $19,847.36 per year, with other benefits leading to most couples receiving more than $20,000 per year. This means that one would expect that very few households would report incomes below the $10,000 - $15,000 category and almost no multi-person households should report incomes below the $20,000 - $25,000 category. And yet 9.6% of renting households reported an income this low in Census 2013 (compared to 2.1% calculated with the HAM version 1.4 data).

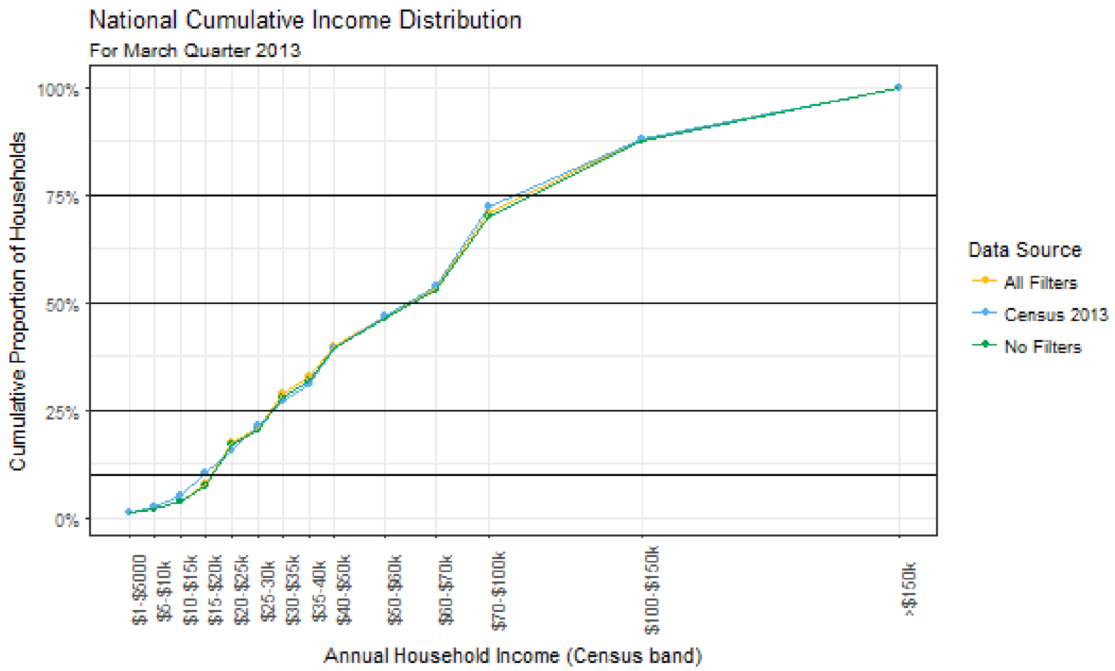
The Guide Notes for Census 2013[9] for converting post-tax income to pre-tax income does not account for student loan payments or child support; this may have led to systematic under-reporting of pre-tax incomes in Census 2013. Given these issues, it is likely inadvisable to try to make the IDI data more closely resemble the Census data.

A similar explanation likely applies for the spikes in households in the $20,000-$25,000 and $30,000-$35,000 ranges. A single person receiving Superannuation and most couples receiving an unemployment benefit would fall into the $20,000 - $25,000 income range, while couples receiving superannuation would fall into the $30,000 - $35,000 income range.

Implications for summary statistics

To see what effect these differences would have on income summary statistics, here is a comparison of the cumulative income distributions. The 10th, 25th, 50th and 75th percentiles are marked for clarity.

[9] http://archive.stats.govt.nz/~/media/Statistics/Census/2013%20Census/forms/2013-guide-notes.pdf



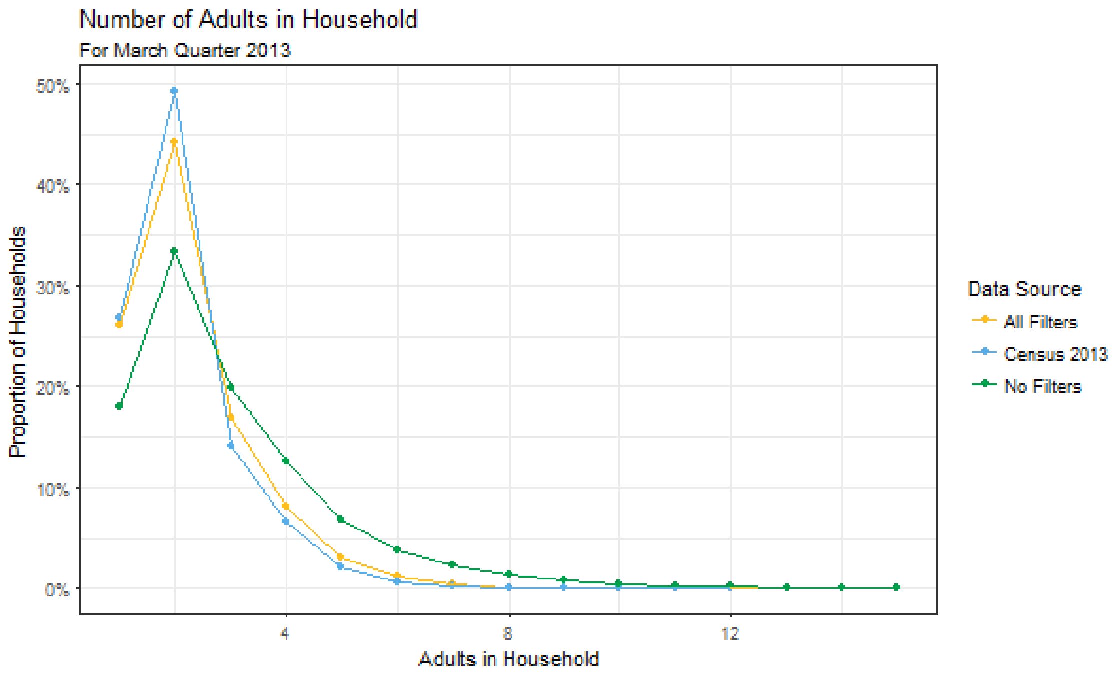
It would appear that these income distributions are sufficiently similar that summary statistics calculated from IDI income data will be almost identical to Census results. The differences should not be any larger than $2000 on annual income, and will in most cases be much less.

Estimating Household size using IDI Data

One of the difficulties with using the IDI to estimate household size is that some people are not easily identified by the administrative sources that comprise the IDI. The effects of this can be seen in Stats NZ’s comparisons between the IDI population and the Estimated Resident Population[10].

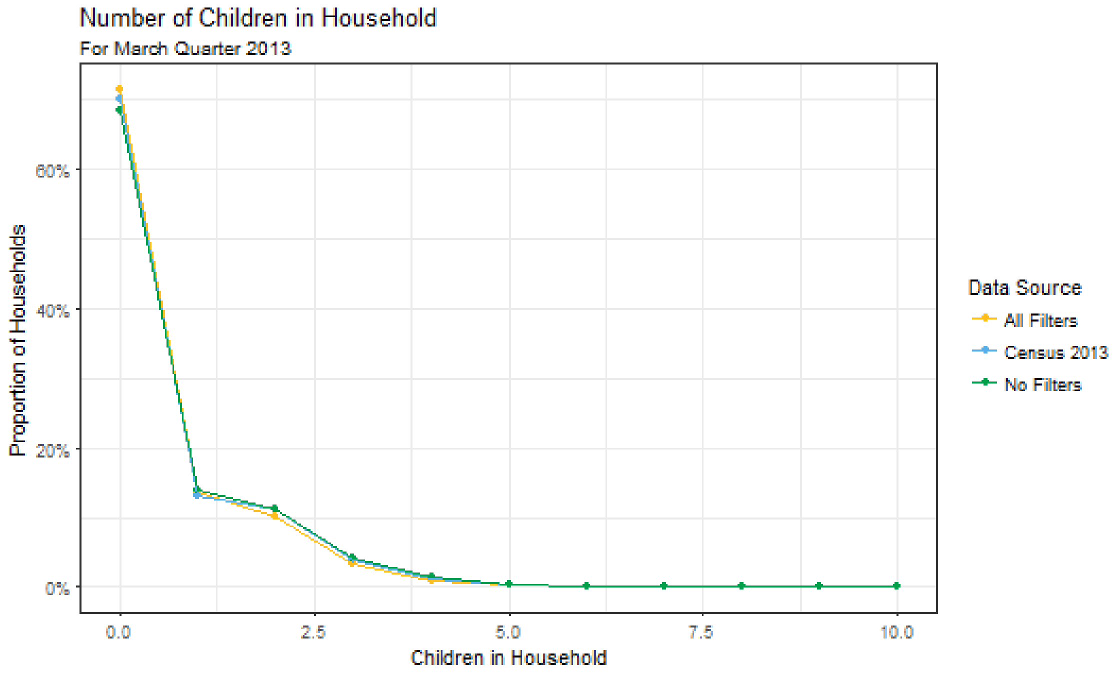
As shown below, the filters applied as part of the income estimation process have a profound effect on the number of adults (people aged 14 and up) per household.

[10] http://archive.stats.govt.nz/methods/research-papers/topss/identifying-nz-resident-pop-in-idi/results.aspx



The filters have reduced the proportion of households with 3 or more adults, while increasing the proportion of 1 and 2-adult households. This brings the distribution of the IDI data more in line with the Census, though there is still a gap between the IDI data and the Census data. It is possible that with additional filtering this distribution could be improved further.

By contrast, the data is already matching children under 14 per household very well.



Appendix B: Comparing HES Data with HAM Percent

Both HAM and HES have measures that aim to help understand housing affordability for the renting population. It is helpful to compare these methods with one another in order to understand the similarities and differences between the information each provides. In particular, while the HAM data allows for more frequent data releases and finer geographical breakdowns, administrative data can have missing data that limits its representativeness, and the method used to construct households out of IDI data is new. For this reason, comparing HES affordability data with HAM Percent is a useful validation test of the IDI data and the method used to construct HAM.

Methods

Both HAM Percent and HES employ an 'outgoing-to-income' ratio (OTI) in their measures of housing affordability. Both are constructed by dividing the housing related costs of a rental household by the total household income, to calculate the percentage of a household's income that is spent on accommodation.

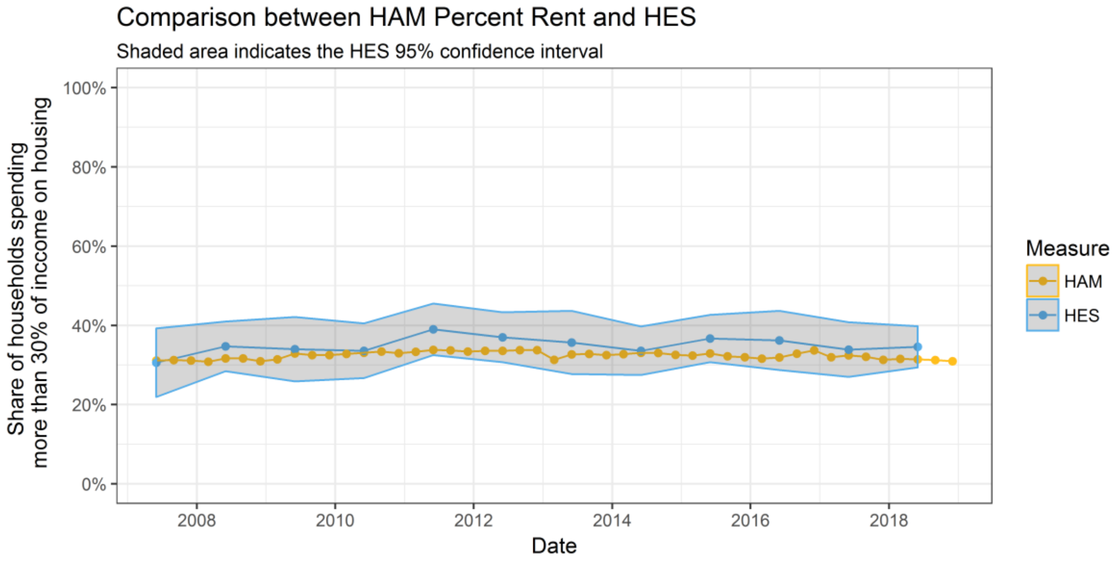
Both housing affordability measures are then constructed by calculating the percentage of households that spend over 30% of their income on housing related costs.

While both measures are constructed in the same way, there are differences between the two outputs based on the underlying data used to calculate them.

The Stats NZ measure uses the Household Economic Survey (HES) as its data source, which is a survey that takes a sample of about 5000 houses around New Zealand.[11]

See Section 2 for more detail on HAM Percent Rent data sources.

Comparison



[11] http://archive.stats.govt.nz/survey-participants/survey-resources/hes-resource.aspx

Over the period analysed, HAM Percent Rent consistently sat within the HES OTI margin of error. This implies the differences between the estimates generated using each data source are not statistically significant.

Another similarity is that, over the period observed, neither method measure trended noticeably upwards or downwards. The HAM Percent Rent value remained between 30% and 34% and the HES OTIs were between 30% and 40%.

During the 11 year period observed, the HES OTI was significantly more volatile than HAM Percent Rent. This is likely due to representation problems within HES's smaller sample rather than real fluctuations in the proportion of households in New Zealand spending more than 30% of their income on housing.

For the majority of the period in which data was available, HAM Percent Rent was slightly lower than the HES OTI. This implies there may be a small systematic difference between the two measures. Some possible reasons for this difference include a systematic under-reporting of incomes in the HES survey or small errors remaining in the way HAM creates household compositions within the IDI.

Appendix C: HAM Production Code

This code is written in R using the packages noted. It requires the IDI to operate and therefore cannot be run outside the Data Lab environment. The HAM Master script activates the other scripts as required, though HAM Household Expense and HAM Income Distribution are not normally re-run.

HAM Master

```
#---------- HAM Master Script
#---------- For HAM 1.4
#---------- Last Updated by James Kerr on 2019-05-27

#---------- STRUCTURE
# This script controls all the other scripts in the HAM Project
# It loads the required packages, sets up the ODBC connections and runs the
other scripts.
# The following scripts are run by HAM Master.R:
  # HAM Data Load.R - Performs all the non-looped data queries
  # HAM Housing Expense.R - Estimates housing costs for HAM Buy (only needs
to be re-run occaisonally)
  # HAM Quarterly Loop.R - Generates unit record HAM data for each Quarter
  # HAM Primary Table.R - creates the main HAM table - HAM by TA/Ward by
Quarter

#---------- Load Packages
require(dplyr)
require(tidyr)
require(RODBC)
require(survey)

#---------- Define IDI Version (Must be Adjusted Manually when updating
HAM)
IDI_version <- "[IDI_Clean_20190420]"

#---------- Permit Writing (no files will be saved if set to FALSE)
write_permitted <- TRUE

#---------- Define File Location
user_dir <- getwd()
## NOTE - This line must be altered when cloning or integrating a fork
setwd(paste0("~/Network-Shares/DataLabNas/MAA/MAA2014-10 Measuring Housing
Affordability/HAM Production"))

#---------- Establish ODBC Connections
idi_connect <- paste0("DRIVER=ODBC Driver 11 for SQL Server;",
                      "SERVER=#########.stats.govt.nz\\######,#####,",
                      "DATABASE=IDI_Clean;",
                      "Trusted_Connection=Yes")

IDI <- odbcDriverConnect(connection=idi_connect)

#---------- Define Thresholds
# source("HAM Income Distribution.R") # Re-run only when necessary
thresholds_2013 <- read.csv("HAM Income Thresholds.csv", h = TRUE)

#---------- Data Load
source("HAM Data Load.R")

#---------- Ownership Expenses
```

```
  # Run Analysis Script (when update Required - Re-Check when HES data
Updates)
  # source("HAM Housing Expense.R")

  # Read in Expense Multipliers and Mortgage Model
  expense_pc <- read.csv("HAM Buy Expense Shares.csv", h=TRUE)
  Rates_rate <- as.numeric(expense_pc$Rates)
  Insurance_rate <- as.numeric(expense_pc$Insurance)

#---------- Household Random Seeds
# Used for Random Rounding - ensures consistent outputs on re-run
source("HAM Random Seeding.R")

#---------- Quarterly Series Loop
# i <- 41 # use instead of loop command (line 19) to produce single
quarter, 41 is code for 2013-03
diagnostics <- FALSE # Change to TRUE to enable diagnostic recording for a
period

for(i in 1:length(periods)) {

  #---------- Define required Dates and figures
  YearEnd <-
as.Date(paste(ifelse(months[i]==12,years[i]+1,years[i]),ifelse(months[i]==1
2,1,months[i]+1),1,sep="-")) # First day past end of year
  YearStart <- as.Date(paste(ifelse(months[i]==12,years[i],years[i]-
1),ifelse(months[i]==12,1,months[i]+1),1,sep="-")) # First day of year
  Ago5 <- as.Date(paste(ifelse(months[i]==12,years[i]-4,years[i]-
5),ifelse(months[i]==12,1,months[i]+1),1,sep="-")) # 5 years before end of
year

  mortgage_int <- fin$Interest[i]

  # Generate Personal Income
  source("HAM Loop Income.R")

  # Construct Households
  source("HAM Loop Household.R")

  # Calculate Housing Costs
  source("HAM Loop Costs.R")

  # Calculate HAM
  source("HAM Loop Affordability.R")

}

#---------- Create Primary HAM Table
source("HAM Primary Table.R")

#---------- Calculate Household Incomes
source("HAM Household Income.R")
```

HAM Income Distribution

```
#---------- HAM HES Income Distribtuion Data
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Created by James Kerr on 2019-05-27

#---------- Load Additonal Package
library(car)

#---------- Age Query
age_blank = "SELECT [hes_per_hes_year_code]
,[snz_hes_uid]
,[snz_hes_hhld_uid]
,[hes_per_age_nbr]
FROM [IDI_Clean].[hes_clean].[hes_person]
WHERE [hes_per_hes_year_code] = 1213"

age_query <- gsub("[IDI_Clean]", IDI_version, age_blank, fixed = TRUE)

age <- sqlQuery(IDI, age_query, stringsAsFactors=FALSE)

#---------- Generate Household Income
# Load HES Household Data
hhd_blank <- "SELECT DISTINCT h.[hes_hhd_hes_year_code]
      ,h.[snz_hes_hhld_uid]
  ,h.[hes_hhd_weight_nbr]
  ,h.[hes_hhd_reg_council_desc_text]
  ,h.[hes_hhd_tenure_code]
  ,h.[hes_hhd_dwell_type_desc_text]
  ,h.[hes_hhd_total_hhold_reginc_amt]
  ,h.[hes_hhd_total_hhold_income_amt]
  ,h.[hes_hhd_hhold_comp_code]
  FROM [IDI_Clean].[hes_clean].[hes_household] h"

hhd_query <- gsub("[IDI_Clean]", IDI_version, hhd_blank, fixed = TRUE)
hes_hhd <- sqlQuery(IDI, hhd_query, stringsAsFactors=FALSE)

# Caluclate Equivalisation Factors
age_adjust <- age %>% group_by(snz_hes_hhld_uid) %>%
  summarise(Old = sum(hes_per_age_nbr >= 14),
            Young = sum(hes_per_age_nbr < 14)) %>%
  mutate(Equiv = 1 + 0.5*(Old-1) + 0.3*Young) # Modified OECD Formula

# Calculate Weighted Medians
working <- hes_hhd %>%
  filter(hes_hhd_hes_year_code == 1213) %>%
  left_join(age_adjust, by = "snz_hes_hhld_uid") %>%
  mutate(Equiv_Income = hes_hhd_total_hhold_reginc_amt/Equiv) %>%
  arrange(Equiv_Income) %>%
  mutate(Fractional_Weight = hes_hhd_weight_nbr/sum(hes_hhd_weight_nbr),
         Cumulative = cumsum(Fractional_Weight))

percent10 <- working %>% filter(Cumulative >= 0.1) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)
percent30 <- working %>% filter(Cumulative >= 0.3) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)
percent50 <- working %>% filter(Cumulative >= 0.5) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)

#---------- Generate Housing Costs for 2012/13
```

```
# Load HES Expendtiure
exp_blank <- "SELECT [hes_exp_hes_year_code]
,[snz_hes_hhld_uid]
,[hes_exp_coding_topic_group_text]
,[hes_exp_coding_topic_text]
,[hes_exp_coding_desc_text]
,[hes_exp_nzhec_code]
,[hes_exp_amount_amt]
,[hes_exp_property_value_amt]
,[hes_exp_land_value_amt]
FROM [IDI_Clean].[hes_clean].[hes_expend]
WHERE [hes_exp_hes_year_code] = 1213"

exp_query <- gsub("[IDI_Clean]", IDI_version, exp_blank, fixed = TRUE)
hes_exp <- sqlQuery(IDI, exp_query, stringsAsFactors=FALSE)

# Code Expenditure Types and Remove Expenditures with No Type
expense_cat <- hes_exp %>% mutate(Expense_Type = "Non_Housing",
                                  Expense_Type = ifelse(hes_exp_nzhec_code
== "04.6.00.0.0.01","BodyCorp",Expense_Type),
                                  Expense_Type =
ifelse(substr(hes_exp_nzhec_code,start=1,stop=9) ==
"04.4.03.1","Rates",Expense_Type),
                                  Expense_Type =
ifelse(substr(hes_exp_nzhec_code,start=1,stop=7) ==
"11.4.02","Insurance",Expense_Type),
                                  Expense_Type = ifelse(hes_exp_nzhec_code
%in%
c("04.2.01.2.0.01","04.2.01.2.0.02","13.1.01.0.1.01","13.1.01.0.1.02"),"Mor
tgage",Expense_Type),
                                  Expense_Type = ifelse(hes_exp_nzhec_code
== "04.1.01.1.0.02", "Rent",Expense_Type))

# Aggregate to Household Level
expense_cat <- expense_cat %>% group_by(snz_hes_hhld_uid, Expense_Type) %>%
  summarise(Expenditure = sum(hes_exp_amount_amt, na.rm=TRUE)) %>%
  spread(Expense_Type, Expenditure, fill = 0)

expenditure <- hes_hhd %>% filter(hes_hhd_hes_year_code == 1213) %>%
  inner_join(expense_cat,by="snz_hes_hhld_uid")

# Determine House Expenditure as % of Income
house_exp_share <- expenditure %>%
  filter(BodyCorp+Insurance+Mortgage+Rates+Rent >0) %>%
  summarise(BodyCorp = weighted.mean(BodyCorp, w = hes_hhd_weight_nbr),
            Insurance = weighted.mean(Insurance, w = hes_hhd_weight_nbr),
            Mortgage = weighted.mean(Mortgage, w = hes_hhd_weight_nbr),
            Rates = weighted.mean(Rates, w = hes_hhd_weight_nbr),
            Rent = weighted.mean(Rent, w = hes_hhd_weight_nbr),
            Income = weighted.mean(hes_hhd_total_hhold_reginc_amt, w =
hes_hhd_weight_nbr)) %>%
  mutate(House_Share = (BodyCorp+Insurance+Mortgage+Rates+Rent)/Income)

exp_percent <- house_exp_share$House_Share

# Determine House Expenditure as % of Income - renters only
house_exp_share_rent <- expenditure %>%
  filter(BodyCorp+Insurance+Mortgage+Rates+Rent >0 & hes_hhd_tenure_code ==
21) %>%
  summarise(BodyCorp = weighted.mean(BodyCorp, w = hes_hhd_weight_nbr),
            Insurance = weighted.mean(Insurance, w = hes_hhd_weight_nbr),
            Mortgage = weighted.mean(Mortgage, w = hes_hhd_weight_nbr),
            Rates = weighted.mean(Rates, w = hes_hhd_weight_nbr),
            Rent = weighted.mean(Rent, w = hes_hhd_weight_nbr),
            Income = weighted.mean(hes_hhd_total_hhold_reginc_amt, w =
hes_hhd_weight_nbr)) %>%
  mutate(House_Share = (BodyCorp+Insurance+Mortgage+Rates+Rent)/Income)

exp_percent_rent <- house_exp_share_rent$House_Share

# Calculate Weighted Median Residual Income
working <- expenditure %>%
  filter(hes_hhd_hes_year_code == 1213) %>%
  left_join(age_adjust, by = "snz_hes_hhld_uid") %>%
  mutate(Resid_Income = hes_hhd_total_hhold_reginc_amt-BodyCorp-Insurance-
Mortgage-Rates-Rent,
         Equiv_Income = Resid_Income/Equiv) %>%
  arrange(Equiv_Income) %>%
  mutate(Fractional_Weight = hes_hhd_weight_nbr/sum(hes_hhd_weight_nbr),
         Cumulative = cumsum(Fractional_Weight))

percent10 <- working %>% filter(Cumulative >= 0.1) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)
percent30 <- working %>% filter(Cumulative >= 0.3) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)
percent50 <- working %>% filter(Cumulative >= 0.5) %>% filter(Cumulative ==
min(Cumulative)) %>% select(Income = Equiv_Income)

# Collate and output results
thresholds <- data.frame(Base_Year = "2012/13", Percentile = c(10,30,50),
bind_rows(percent10,percent30,percent50))
names(thresholds) <- c("Base_Year", "Percentile", "Threshold")

#thresholds$Housing_Share <- round(exp_percent,3)
#thresholds$Threshold <- thresholds$Income * (1-exp_percent)

if(write_permitted){write.csv(thresholds,"HAM Income Thresholds.csv",
row.names = FALSE)}

#---------- Generate Household Demographics
dem <- age %>% group_by(snz_hes_hhld_uid) %>%
  summarise(Old = sum(hes_per_age_nbr >= 14),
            Young = sum(hes_per_age_nbr < 14)) %>%
  inner_join(select(hes_hhd,hes_hhd_weight_nbr, snz_hes_hhld_uid,
hes_hhd_total_hhold_reginc_amt), by = "snz_hes_hhld_uid") %>%
  inner_join(expense_cat, by = "snz_hes_hhld_uid") %>%
  filter(Non_Housing > 0)

dem$Pop_Unit <- dem$Old + 0.6*(dem$Young)
dem$Pop_1 <- dem$Pop_Unit == 1
dem$All_Expense <-
rowSums(select(dem,Rates,Insurance,BodyCorp,Rent,Mortgage,Non_Housing))

income_model <- lm(All_Expense ~ Pop_Unit+Pop_1,
                   data = dem, weight = hes_hhd_weight_nbr)

sum(income_model$coefficients)/income_model$coefficients["Pop_Unit"]

expense_model <- lm(Non_Housing ~ Pop_Unit+Pop_1,
                   data = dem, weight = hes_hhd_weight_nbr)

sum(expense_model$coefficients)/expense_model$coefficients["Pop_Unit"]
```

HAM Data Load

```
#---------- HAM Master Data Load Script
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Last Updated by James Kerr on 2019-05-27

# This script loads all the data that is not loaded from within a loop
# It also defines data such as the HAM thresholds

#---------- Finance and CPI Data
fin <- read.csv("Input Data/Finance and CPI.csv",h=TRUE,na.strings="")

CPI2013 <- fin$CPI[fin$Date == "1/03/2013"]

#---------- Geogrpahical Concordance Table
geo_query <- "SELECT [MB2018_V1_00] Meshblock
,[TA2018_V1_00_NAME] TA_Name
,[WARD2018_V1_00_NAME] Ward_Name
,[REGC2018_V1_00_NAME] Region

FROM
[IDI_Metadata].[clean_read_CLASSIFICATIONS].[meshblock_current_higher_geogr
aphy]"

geo <- sqlQuery(IDI,geo_query,stringsAsFactors=FALSE)

geo <- mutate(geo, Area = ifelse(TA_Name=="Auckland",paste0("Auckland:
",Ward_Name),as.character(TA_Name))) %>%
  select(Meshblock,Area,Region)

if(write_permitted) {save(geo, file = "Unit Record/Area Definitions.rda")}

#---------- Self-Employed Income Query for annual series
  self_query <- "select i.snz_uid,
  s.snz_uid,
  p.snz_uid,
  i.ir_ir3_return_period_date,
  s.ir_ir4_return_period_date,
  p.ir_ir20_return_period_date,
  i.ir_ir3_tot_pship_income_amt IR3_Pship,
  i.ir_ir3_tot_sholder_salary_amt IR3_Share,
  i.ir_ir3_net_rents_826_amt IR3_Rent,
  i.ir_ir3_net_profit_amt IR3_SoleTrader,
  i.ir_ir3_tot_expenses_claimed_amt IR3_Expense,
  s.ir_ir4_tot_sholder_sal_809_amt IR4s_Share,
  p.ir_ir20_tot_share_of_inc_865_amt IR20_Pship

  from [IDI_Clean].[ir_clean].[ird_rtns_keypoints_ir3] i

  full outer join [IDI_Clean].[ir_clean].[ird_attachments_ir4s] s
on(i.snz_uid = s.snz_uid and i.ir_ir3_return_period_date =
s.ir_ir4_return_period_date)

  full outer join [IDI_Clean].[ir_clean].[ird_attachments_ir20] p
on(i.snz_uid = p.snz_uid and i.ir_ir3_return_period_date =
p.ir_ir20_return_period_date)"

  self_query <- gsub("[IDI_Clean]", IDI_version, self_query, fixed = TRUE)
  self_inc <- sqlQuery(IDI,self_query,stringsAsFactors=FALSE)
  # Consolidate data from each form
  self_inc <- self_inc %>% mutate(IR3_Date =
as.Date(ir_ir3_return_period_date),
                                IR4_Date =
as.Date(ir_ir4_return_period_date),
                                IR20_Date =
as.Date(ir_ir20_return_period_date)) %>%
  transmute(snz_uid =
ifelse(!is.na(snz_uid),snz_uid,ifelse(!is.na(snz_uid.1),snz_uid.1,snz_uid.2
)),
            Period = as.Date(ifelse(!is.na(IR3_Date),IR3_Date,

ifelse(!is.na(IR4_Date),IR4_Date,IR20_Date)),origin="1970-01-01"),
            Pship = ifelse(is.na(IR3_Pship),IR20_Pship,IR3_Pship),
            Shareholder = ifelse(is.na(IR3_Share),IR4s_Share,IR3_Share),
            Rent = IR3_Rent,
            SoleTrader = IR3_SoleTrader,
            Expenses = IR3_Expense)

  self_inc$Income <- rowSums(self_inc[,-1:-2],na.rm=TRUE)

#---------- House Price Summary Data
evaluer <- read.csv("Input Data/Model Evaluer Summary Data.csv",h=TRUE)

evaluer <- evaluer %>%
  mutate(Date = as.Date(Date))

# Revaluations are normally done in regular cycles
# In some cases there may be a limited revalaution in a year, these partial
years may be unrepresentative.


rateable_values <- read.csv("Input Data/Rateable History Summary
Data.csv",h=TRUE)

# Define three-year blocks
year_blocks <- data.frame(Date = unique(rateable_values$Date)) %>%
  filter(Date >= 2001) %>% # 2001-2003 is the first block
  arrange(Date)

year_blocks <- year_blocks %>%
  mutate(Block = ceiling(1:nrow(year_blocks)/3))

# Identify the highest count in each 3-year block and delete the others
rateable_values <- rateable_values %>%
  left_join(year_blocks, by = "Date") %>%
  group_by(Area, Beds, Block) %>%
  mutate(Max_Count = max(Count)) %>%
  ungroup() %>%
  filter(Count == Max_Count, !is.na(Block))

#---------- Master Bond Query
  bond_query <- "SELECT [snz_dbh_bond_uid] bond
  ,[dbh_bond_rental_amt] rent
  ,[dbh_bond_bond_lodged_date] lodged
  ,[dbh_bond_bond_closed_date] closed
  ,[snz_idi_address_register_uid] address
  ,[dbh_bond_address_type_text] address_type
  ,[dbh_bond_property_type_text] property_type
  ,[dbh_bond_bedroom_count_code] bedrooms
  ,[dbh_bond_region_code]
  ,[dbh_bond_ta_code]
  ,[snz_dbh_property_uid]
  ,[dbh_bond_tenancy_end_date]
  FROM [IDI_Clean].[dbh_clean].[bond_lodgement]"

  bond_query <- gsub("[IDI_Clean]", IDI_version, bond_query, fixed = TRUE)

  bonds <- sqlQuery(IDI,bond_query,stringsAsFactors=FALSE)
  bonds <- bonds %>% filter(!duplicated(bonds[,c("address", "lodged")],
fromLast = TRUE), !is.na(address))

  bonds$rent[bonds$rent == 0] <- NA # 0 is a missing value code

#---------- Set up Date sequence
  years <- as.numeric(substr(fin$Date,6,9))
  months <- as.numeric(substr(fin$Date,3,4))

  periods <- format(as.Date(paste(years,months,1,sep="-")),format="%Y-%m")
  annuals <- periods[months == 3] # subsets periods to get March periods
only

  tax_years <- as.Date(paste0(years,"-03-31")) # tax years for
self_employment Income
```

HAM Housing Expense

```
#---------- HAM Housing Expense Percentages
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Created by James Kerr on 2019-05-27

# This script takes HES and housing data and estimates housing costs for
owner-ocucpiers.
# It need not be run every quarter, as HES data is only updated annually.

#---------- Collate HES Expenditure Data
  # Load HES Expendtiure
exp_blank <- "SELECT [hes_exp_hes_year_code]
      ,[snz_hes_hhld_uid]
      ,[hes_exp_coding_topic_group_text]
      ,[hes_exp_coding_topic_text]
      ,[hes_exp_coding_desc_text]
      ,[hes_exp_nzhec_code]
      ,[hes_exp_amount_amt]
        ,[hes_exp_property_value_amt]
        ,[hes_exp_land_value_amt]
  FROM [IDI_Clean].[hes_clean].[hes_expend]"

  exp_query <- gsub("[IDI_Clean]", IDI_version, exp_blank, fixed = TRUE)
  hes_exp <- sqlQuery(IDI, exp_query, stringsAsFactors=FALSE)

  # Code Expenditure Types and Remove Expenditures with No Type
  expenditure <- hes_exp %>% mutate(Expense_Type = NA,
                                Expense_Type = ifelse(hes_exp_nzhec_code ==
"04.6.00.0.0.01","BodyCorp",Expense_Type),
                                Expense_Type =
ifelse(substr(hes_exp_nzhec_code,start=1,stop=9) ==
"04.4.03.1","Rates",Expense_Type),
                                Expense_Type =
ifelse(substr(hes_exp_nzhec_code,start=1,stop=7) ==
"11.4.02","Insurance",Expense_Type),
                                Expense_Type = ifelse(hes_exp_nzhec_code
%in%
c("04.2.01.2.0.01","04.2.01.2.0.02","13.1.01.0.1.01","13.1.01.0.1.02"),"Mor
tgage",Expense_Type)) %>%
    filter(!is.na(Expense_Type))

  # Aggregate to Household Level
  expenditure <- expenditure %>% group_by(hes_exp_hes_year_code,
snz_hes_hhld_uid, Expense_Type) %>%
    summarise(Expenditure = sum(hes_exp_amount_amt, na.rm=TRUE)) %>%
    spread(Expense_Type, Expenditure)

  # Collate Rateable Value Data
  rv <- hes_exp %>% filter(!is.na(hes_exp_property_value_amt) |
!is.na(hes_exp_land_value_amt)) %>%
    select(hes_exp_hes_year_code, snz_hes_hhld_uid,
hes_exp_property_value_amt, hes_exp_land_value_amt) %>%
    unique()

#---------- Collate Ratings Data
  # Load HES Household Data
  hhd_blank <- "SELECT DISTINCT h.[hes_hhd_hes_year_code]
      ,h.[snz_hes_hhld_uid]
  ,h.[hes_hhd_weight_nbr]
  ,h.[hes_hhd_reg_council_desc_text]
  ,h.[hes_hhd_tenure_code]
  ,h.[hes_hhd_dwell_type_desc_text]
  ,a.[snz_idi_address_register_uid]
  ,h.[hes_hhd_total_hhold_reginc_amt]
  FROM [IDI_Clean].[hes_clean].[hes_household] h

  JOIN [IDI_Clean].[hes_clean].[hes_address] a
  ON(h.[snz_hes_hhld_uid] = a.[snz_hes_hhld_uid])"

  hhd_query <- gsub("[IDI_Clean]", IDI_version, hhd_blank, fixed = TRUE)
  hes_hhd <- sqlQuery(IDI, hhd_query, stringsAsFactors=FALSE)

  # Identify Owner-Occupiers
  hes_hhd <- hes_hhd %>% mutate(Own_Occ = hes_hhd_tenure_code %in%
c(10:12,30:32)) %>%
    filter(!is.na(snz_idi_address_register_uid) & Own_Occ)

#---------- Collate HES Age Data
  age_blank <- "SELECT [snz_hes_hhld_uid]
      ,AVG_Adult = sum([hes_per_age_nbr])/count([hes_per_age_nbr])

  FROM [IDI_Clean].[hes_clean].[hes_person]

  where [hes_per_in_child_role_code] = 0
  and [hes_per_age_nbr] is not NULL

  group by [snz_hes_hhld_uid]"

  age_query <- gsub("[IDI_Clean]", IDI_version, age_blank, fixed = TRUE)
  hes_age <- sqlQuery(IDI, age_query, stringsAsFactors=FALSE)

  hes_hhd <- left_join(hes_hhd, hes_age, by ="snz_hes_hhld_uid")

#---------- Collate Sales Price Data
  # Load Sales Sata
  sales_query <- "SELECT d.[snz_idi_address_register_uid]
      ,s.[sale_id]
  ,s.[sale_date]
  ,s.[sale_price]
  FROM [IDI_Adhoc].[clean_read_DBH_QV].[QV_Sales_201808] s

  JOIN [IDI_Adhoc].[clean_read_DBH_QV].[QV_Dwelling_201808] d
  ON s.[QPID_UID] = d.[QPID_UID]

  where s.[sale_type] = 'S11'
  OR s.[sale_type] = 'S12'"

  sales <- sqlQuery(IDI, sales_query, stringsAsFactors=FALSE)

  # Filter to HES dwellings and Remove duplicate sales on same date
  sales <- sales %>% filter(snz_idi_address_register_uid %in%
hes_hhd$snz_idi_address_register_uid)

  sales <- sales %>% filter(!duplicated(sales %>%
select(snz_idi_address_register_uid,sale_date), fromLast = TRUE))

  # Generate Latest Sale by Date
  sales$sale_date <- as.Date(sales$sale_date)

  hes_periods <- unique(hes_hhd$hes_hhd_hes_year_code)
  hes_years <- ifelse(hes_periods < 1000, hes_periods + 9000 , hes_periods)
%>%
    substr(3,4) %>%
    as.numeric() + 2000

  hes_dates <- paste0(hes_years,"-06-30") %>% as.Date()

  prices <- data.frame(snz_idi_address_register_uid = numeric(0))

  for(i in 1:length(hes_periods)) {
    sampled_props <- filter(hes_hhd, hes_hhd_hes_year_code ==
hes_periods[i]) %>%
      mutate(Year_Ending = hes_years[i])

    latest_sales <- sales %>% filter(sale_date <= hes_dates[i]) %>%
      group_by(snz_idi_address_register_uid) %>%
      summarise(sale_date = max(sale_date))

    latest_prices <- inner_join(sales,latest_sales) %>%
      inner_join(sampled_props, by = "snz_idi_address_register_uid")

    prices <- bind_rows(prices, latest_prices)
  }

#---------- Collate E-Valuer Data
  # Load E-valuer Data
  evalue_query <- "SELECT d.[snz_idi_address_register_uid]
      ,e.[val_date_year]
  ,e.[val_date_Qtr_ended]
  ,e.[Est_value]
  FROM [IDI_Adhoc].[clean_read_DBH_QV].[QV_Evaluer_History_201808] e

  JOIN [IDI_Adhoc].[clean_read_DBH_QV].[QV_Dwelling_201808] d

  ON d.QPID_UID = e.QPID_UID

  WHERE e.[val_date_Qtr_ended] = 'Q2'
  AND e.[val_date_year] > 2006"

  evaluer <- sqlQuery(IDI, evalue_query, stringsAsFactors=FALSE)
  evaluer <- evaluer %>% mutate(hes_exp_hes_year_code =
as.numeric(paste0(as.numeric(substr(val_date_year,3,4))-1,

substr(val_date_year,3,4))))

#---------- Join Data Sets
  hes <- inner_join(prices, expenditure) %>%
    inner_join(rv) %>%

inner_join(evaluer,by=c("hes_exp_hes_year_code","snz_idi_address_register_u
id"))

#---------- Generate Average Expenditure Calculations
  expend <- hes %>% summarise(BodyCorp = weighted.mean(BodyCorp/Est_value,
w=hes_hhd_weight_nbr,na.rm=TRUE),
                                   Insurance =
weighted.mean(Insurance/Est_value, w=hes_hhd_weight_nbr,na.rm=TRUE),
                                   Rates =
weighted.mean(Rates/hes_exp_property_value_amt,
w=hes_hhd_weight_nbr,na.rm=TRUE))
```

```
  if(write_permitted){write.csv(expend,"HAM Buy Expense
Shares.csv",row.names=FALSE)}

#---------- Perform Mortgage Analysis - Meant for HAM OWN, not used in
current HAM version
  # Prepare data for modelling
  model_data <- hes %>% mutate(hes_code = ifelse(hes_hhd_hes_year_code <
1000, hes_hhd_hes_year_code + 9000 , hes_hhd_hes_year_code),
                              hes_year = as.numeric(substr(hes_code,3,4))
+2000,
                              hes_date = as.Date(paste0(hes_year,"-06-
01")),
                              Payment_Mult = Mortgage/sale_price) %>%
    select(hes_date, sale_price, sale_date,AVG_Adult, Payment_Mult,
Mortgage,
           Income = hes_hhd_total_hhold_reginc_amt,
           Weight = hes_hhd_weight_nbr,
           Region = hes_hhd_reg_council_desc_text)

  fin$Date <- as.Date(fin$Date, format= "%d/%m/%Y")

  model_data <- left_join(model_data, fin, by=c("hes_date" = "Date")) %>%
select(-CPI,-Floating)

  model_data <- model_data %>% filter(complete.cases(model_data) & Income >
0)

  model_data$sale_age <- as.numeric(model_data$hes_date -
model_data$sale_date) + 30 # 30 is number of days in June

  # Calculate Regression Model
  mortgage_model <- lm(Payment_Mult ~ log(Income)*log(AVG_Adult) +
log(Effective) + log(sale_age) + log(sale_price),
                       data = model_data, weight = Weight)


  summary(mortgage_model)
  AIC(mortgage_model)

  # Calculate Estimated Mortgage Model
  model_data <- model_data %>%
    mutate(Est_Mortgage = (0.8*sale_price*Effective)/(1-(1+Effective)^-30))

  # Measure magnitude of errors
  fitted_data <- predict(mortgage_model,newdata = model_data,weights =
model_data$Weight)

  output <- model_data %>% mutate(model_var = abs(fitted_data*sale_price -
Mortgage),
                                  naive_var =
abs(mean(Payment_Mult)*sale_price - Mortgage),
                                  est_var = abs(Est_Mortgage - Mortgage))

  summary(output$model_var)
  summary(output$naive_var)
  summary(output$est_var)

  # Write model
  if(write_permitted){
    write.csv(mortgage_model$coefficients,"HAM-SO mortgage
model.csv",row.names=TRUE)
    save(mortgage_model,file="HAM-SO mortgage model.Rda")
  }
```

HAM Random Seeding

```
#---------- HAM Random Seed Generator
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Last Updated by James Kerr on 2019-05-27

# Assigns each address a unique number that is persistent for each address
in the refresh.
# This ensures the code will produce the same random-rounded counts when
re-run

# Load all households from address_notification

seed_blank <- "SELECT DISTINCT [snz_idi_address_register_uid] address

  FROM [IDI_Clean].[data].[address_notification]

  ORDER BY address"

seed_query <- gsub("[IDI_Clean]", IDI_version, seed_blank, fixed = TRUE)

seed_table <- sqlQuery(IDI,seed_query,stringsAsFactors=FALSE)

# Apply Random seeds
set.seed(201906)
seed_table <- seed_table %>%
  mutate(Random_Seed = runif(nrow(seed_table)))

# Save seed Table
if(write_permitted) {save(seed_table, file = "Unit Record/Random
Seeds.rda")}
```

HAM Loop Income

```
#---------- HAM Income Estimation
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function, Runs Inside Loop
#---------- Created by James Kerr on 2019-05-01

# NOTE: i denotes the quarter the data is being run on.

#---------- Personal Income
# EMS Income
ems_blank <- "SELECT [snz_uid]
,sum([ir_ems_gross_earnings_amt]) Income
,count([ir_ems_gross_earnings_amt]) Returns
FROM [IDI_Clean].[ir_clean].[ird_ems]

where ir_ems_return_period_date >= 'YearStart'
and ir_ems_return_period_date < 'YearEnd'
group by [snz_uid]"

ems <- ems_blank
ems <- gsub("[IDI_Clean]",IDI_version,ems, fixed = TRUE)
ems <- gsub("YearEnd",YearEnd,ems, fixed = TRUE)
ems <- gsub("YearStart",YearStart,ems, fixed = TRUE)

ems_inc <- sqlQuery(IDI,ems,stringsAsFactors=FALSE) %>% mutate(Source =
"EMS")

# Tier 2 benefit income
tier2_blank <- "SELECT [snz_uid]
,[msd_ste_supp_serv_code]
,[msd_ste_start_date]
,[msd_ste_end_date]
,[msd_ste_daily_gross_amt]
FROM [IDI_Clean].[msd_clean].[msd_second_tier_expenditure]

where [msd_ste_end_date] >= 'YearStart'
and [msd_ste_start_date] < 'YearEnd'"

tier2 <- tier2_blank
tier2 <- gsub("[IDI_Clean]",IDI_version,tier2, fixed = TRUE)
tier2 <- gsub("YearEnd",YearEnd,tier2, fixed = TRUE)
tier2 <- gsub("YearStart",YearStart,tier2, fixed = TRUE)

tier2_inc <- sqlQuery(IDI,tier2,stringsAsFactors=FALSE) %>% mutate(Source =
"Tier 2 Benefit")

tier2_inc$msd_ste_start_date[tier2_inc$msd_ste_start_date < YearStart] <-
YearStart
tier2_inc$msd_ste_end_date[tier2_inc$msd_ste_end_date >= YearEnd] <-
YearEnd-1

tier2_inc <- tier2_inc %>% mutate(Duration = as.numeric(msd_ste_end_date -
msd_ste_start_date +1),
                                  Income = msd_ste_daily_gross_amt *
Duration,
                                  AS_Income = msd_ste_daily_gross_amt *
Duration * msd_ste_supp_serv_code == 471)

# 471 is code for Accommodation Supplement (AS)
# Tier 3 benefit income
tier3_blank <- "SELECT [snz_uid]
,[msd_tte_decision_date]
,[msd_tte_pmt_amt] Tier3_Inc
,[msd_tte_recoverable_ind]
FROM [IDI_Clean].[msd_clean].[msd_third_tier_expenditure]

where [msd_tte_decision_date] >= 'YearStart'
and [msd_tte_decision_date] < 'YearEnd'"

tier3 <- tier3_blank
tier3 <- gsub("[IDI_Clean]",IDI_version,tier3, fixed = TRUE)
tier3 <- gsub("YearEnd",YearEnd,tier3, fixed = TRUE)
tier3 <- gsub("YearStart",YearStart,tier3, fixed = TRUE)

tier3_inc <- sqlQuery(IDI,tier3,stringsAsFactors=FALSE)
tier3_inc <- tier3_inc %>% group_by(snz_uid) %>%
  summarise(Payments = n(),
            Recoverable = sum(msd_tte_recoverable_ind == "Y"),
            Income = sum(Tier3_Inc)) %>% mutate(Source = "Tier 3 Benefit")

# Self-Employment Income NOT AVAILABLE FOR QUARTERLY SERIES
se_inc <- self_inc %>% filter(Period == tax_years[i]) %>%
  select(snz_uid,Income) %>%
  mutate(Source = "Self-Employment") %>%
  filter(Income > 0)

# Combine income for all sources at individual level
incomes <- bind_rows(select(ems_inc,snz_uid,Income,Source,Returns),

se_inc,select(tier2_inc,snz_uid,Income,AS_Income,Source),
                     select(tier3_inc,snz_uid,Income,Source)) %>%
  group_by(snz_uid) %>%
  summarise(Income = sum(Income),
            Income_Excl_SE = sum(ifelse(Source == "Self-
Employment",0,Income)),
            AS_Income = sum(AS_Income),
            EMS = sum(Returns, na.rm=TRUE),
            Tier2 = sum(Source == "Tier 2 Benefit"),
            Tier3 = sum(Source == "Tier 3 Benefit"),
            SE = sum(Source == "Self-Employment"))
```

HAM Loop Household

```
#---------- HAM Household Construction
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function, Runs Inside Loop
#---------- Created by James Kerr on 2019-05-27

# NOTE: i denotes the quarter the data is being run on.

#---------- Household Characteristics
hhold_blank <- "SELECT  p.[snz_uid]
,a.[snz_idi_address_register_uid]
,a.[ant_meshblock_code] Meshblock
,a.[ant_notification_date]
,a.[ant_replacement_date]
,p.[snz_birth_year_nbr]
,p.[snz_birth_month_nbr]
,p.[snz_deceased_year_nbr]
,p.[snz_deceased_month_nbr]
,a.[ant_address_source_code]

from [IDI_Clean].[data].[address_notification] a

join [IDI_Clean].[data].[personal_detail] p on (a.[snz_uid] = p.[snz_uid])

where a.[snz_idi_address_register_uid] is not NULL
and a.[ant_notification_date] < 'YearEnd'
and a.[ant_replacement_date] >= 'YearEnd'
and p.[snz_spine_ind] != 0"

hhold <- hhold_blank
hhold <- gsub("[IDI_Clean]",IDI_version,hhold, fixed = TRUE)
hhold <- gsub("YearEnd",YearEnd,hhold, fixed = TRUE)

hhold_char <- sqlQuery(IDI,hhold,stringsAsFactors=FALSE)

#---------- Diagnostic testing
if(diagnostics) {
  hhold_geo <- left_join(hhold_char, geo, by = "Meshblock")

  hholds_by_area <- hhold_geo %>% group_by(Area) %>%
    summarise(People = n(), Addresses =
sum(!duplicated(snz_idi_address_register_uid)))

  write.csv(hholds_by_area, "Diagnostics/HHolds and people no
validation.csv", row.names = FALSE)
}

# Find Valid Population IDs
blank_moh <- "select    distinct snz_uid
from [IDI_Clean].[moh_clean].[gms_claims]
where moh_gms_visit_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moh_clean].[lab_claims]
where moh_lab_visit_date between 'YearStart' and 'YearEnd'

union
```

```
select      distinct snz_uid
from [IDI_Clean].[moh_clean].[nnpac]
where moh_nnp_service_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moh_clean].[pharmaceutical]
where moh_pha_dispensed_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moh_clean].[pho_enrolment]
where moh_pho_last_consul_date between 'YearStart' and 'YearEnd'
or moh_pho_enrolment_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moh_clean].[pub_fund_hosp_discharges_event]
where moh_evt_evst_date between 'YearStart' and 'YearEnd'"

valid_moh <- blank_moh
valid_moh <- gsub("[IDI_Clean]",IDI_version,valid_moh, fixed = TRUE)
valid_moh <- gsub("YearEnd",YearEnd,valid_moh)
valid_moh <- gsub("YearStart",YearStart,valid_moh)

moh_valids <- sqlQuery(IDI,valid_moh,stringsAsFactors=FALSE) %>% unique()

blank_moe <- "select    distinct snz_uid
from [IDI_Clean].[moe_clean].[enrolment]
where moe_enr_prog_start_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moe_clean].[student_enrol]
where moe_esi_start_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[moe_clean].[tec_it_learner]
where moe_itl_start_date between 'YearStart' and 'YearEnd'"

valid_moe <- blank_moe
valid_moe <- gsub("[IDI_Clean]",IDI_version,valid_moe, fixed = TRUE)
valid_moe <- gsub("YearEnd",YearEnd,valid_moe)
valid_moe <- gsub("YearStart",YearStart,valid_moe)

moe_valids <- sqlQuery(IDI,valid_moe,stringsAsFactors=FALSE) %>% unique()

blank_acc <- "select    distinct snz_uid
from [IDI_Clean].[acc_clean].[claims]
where acc_cla_lodgement_date between 'YearStart' and 'YearEnd'"

valid_acc <- blank_acc
valid_acc <- gsub("[IDI_Clean]",IDI_version,valid_acc, fixed = TRUE)
valid_acc <- gsub("YearEnd",YearEnd,valid_acc)
valid_acc <- gsub("YearStart",YearStart,valid_acc)
```

```
acc_valids <- sqlQuery(IDI,valid_acc,stringsAsFactors=FALSE) %>% unique()

blank_ird <- "select    distinct snz_uid
from [IDI_Clean].[ir_clean].[ird_ems]
where ir_ems_return_period_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[ir_clean].[ird_rtns_keypoints_ir3]
where ir_ir3_return_period_date between 'YearStart' and 'YearEnd'
union

select      distinct snz_uid
from [IDI_Clean].[ir_clean].[ird_attachments_ir4s]
where ir_ir4_return_period_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[ir_clean].[ird_attachments_ir20]
where ir_ir20_return_period_date between 'YearStart' and 'YearEnd'"

valid_ird <- blank_ird
valid_ird <- gsub("[IDI_Clean]",IDI_version,valid_ird, fixed = TRUE)
valid_ird <- gsub("YearEnd",YearEnd,valid_ird)
valid_ird <- gsub("YearStart",YearStart,valid_ird)

ird_valids <- sqlQuery(IDI,valid_ird,stringsAsFactors=FALSE) %>% unique()

blank_msd <- "select    distinct snz_uid
from [IDI_Clean].[msd_clean].[msd_third_tier_expenditure]
where msd_tte_decision_date between 'YearStart' and 'YearEnd'

union

select      distinct snz_uid
from [IDI_Clean].[msd_clean].[msd_second_tier_expenditure]
where msd_ste_start_date <= 'YearEnd'
and msd_ste_end_date > 'YearStart'"

valid_msd <- blank_msd
valid_msd <- gsub("[IDI_Clean]",IDI_version,valid_msd, fixed = TRUE)
valid_msd <- gsub("YearEnd",YearEnd,valid_msd)
valid_msd <- gsub("YearStart",YearStart,valid_msd)

msd_valids <- sqlQuery(IDI,valid_msd,stringsAsFactors=FALSE) %>% unique()

all_valids <- bind_rows(acc_valids, ird_valids, moe_valids, moh_valids,
msd_valids) %>% unique()

# Subset to people alive during time period
hhold_char <- hhold_char %>% mutate(Birthday =
as.Date(paste(snz_birth_year_nbr,snz_birth_month_nbr,1,sep="-"),format="%Y-
%m-%d"),
                                    Deathday =
as.Date(paste(snz_deceased_year_nbr,snz_deceased_month_nbr,1,sep="-
"),format="%Y-%m-%d"))
```

```
hhold_char <- hhold_char %>% filter(Birthday < YearEnd & (Deathday >=
YearEnd | is.na(Deathday)))

# Identify people Under 5
hhold_char$Under5 <-  (YearEnd - hhold_char$Birthday) <= 365.25*5 # Days in
Year times 5

# Remove People 5 or over without valid ID
hhold_char$Valid_Pop <- hhold_char$snz_uid %in% all_valids$snz_uid |
hhold_char$snz_uid %in% incomes$snz_uid| hhold_char$Under5

#---------- Diagnostic testing
if(diagnostics){
  hhold_geo2 <- left_join(hhold_char, geo, by = "Meshblock")

  hholds_by_area2 <- hhold_geo2 %>% group_by(Area) %>%
    summarise(People = n(), Valid_People = sum(Valid_Pop), Addresses =
sum(!duplicated(snz_idi_address_register_uid)))

  write.csv(hholds_by_area2, "Diagnostics/HHolds and people validated.csv",
row.names = FALSE)
}

#---------- Check for people out-of-country
blank_border <- "SELECT b.[snz_uid]
,max([pos_applied_date]) last_left
,max([pos_ceased_date]) last_arrived

FROM [IDI_Clean].[data].[person_overseas_spell] b

INNER JOIN [IDI_Clean].[data].[address_notification] a

ON a.[snz_uid] = b.[snz_uid]

WHERE [pos_applied_date] <= 'YearEnd'
AND [pos_ceased_date] >= 'YearEnd'
AND [snz_idi_address_register_uid] is not NULL
AND [ant_notification_date] <= 'YearEnd'
AND [ant_replacement_date] >= 'YearEnd'

GROUP BY b.[snz_uid]"

valid_border <- blank_border
valid_border <- gsub("[IDI_Clean]",IDI_version,valid_border, fixed = TRUE)
valid_border <- gsub("YearEnd",YearEnd,valid_border)
valid_border <- gsub("YearStart",YearStart,valid_border)

# Query data from IDI
border_movement <- sqlQuery(IDI,valid_border,stringsAsFactors=FALSE)

# Convert dates to YYYY-MM-DD format
border_movement$last_left <- substr(border_movement$last_left,1,10) %>%
as.Date()
border_movement$last_arrived <- substr(border_movement$last_arrived,1,10)
%>% as.Date()

# Check Time out of Country
border_movement <- border_movement %>%
  mutate(Left_Country = last_left < YearEnd & last_arrived >= YearEnd-1,
         Gone_for_30 = Left_Country & last_arrived > YearEnd+30)
```

```
hhold_char <- left_join(hhold_char, border_movement, by = "snz_uid")

# Out of country variables should be FALSE if they were not in the
border_movement table
hhold_char$Left_Country[is.na(hhold_char$Left_Country)] <- FALSE
hhold_char$Gone_for_30[is.na(hhold_char$Gone_for_30)] <- FALSE

# Stable address identification
# Checks for people who have been in their current address for a period of
time either side of the reference date
# Centre of window is last date of quarter = YearEnd -1

window_centre <- YearEnd - 1 # Last day of quarter

hhold_char <- hhold_char %>%
  mutate(Stable_61 = (ant_notification_date <= window_centre-30 &
ant_replacement_date >= window_centre+30) |
           ant_address_source_code == "CEN ") # "CEN " is code for Census -
prevents address loss due to census

# Personal Information Table
person_char <- hhold_char %>%
  mutate(Quarter = periods[i]) %>%
  left_join(incomes,by="snz_uid") %>%
  select(Quarter, snz_uid, snz_idi_address_register_uid, Meshblock, Income,
AS_Income, Birthday, Deathday, Under5, Valid_Pop, Left_Country,
Gone_for_30, Stable_61) %>%
  mutate(Age = (YearEnd - Birthday)/365.25,
         Pass_All_Filters = Valid_Pop & !Gone_for_30 & Stable_61) %>%
  ungroup()

if(write_permitted) {save(person_char, file = paste0("Unit Record/HAM
Microdata ","Household Map ",periods[i],".rda"))}


# Create Household Income
hhold_income <- person_char %>%
  filter(Pass_All_Filters) %>%
  group_by(snz_idi_address_register_uid, Meshblock) %>%
  summarise(Hhold_Size = n(),
            Income_Count = sum(Income > 0 & !is.na(Income)),
            Old = sum((Age >= 14) | (Income > 0 & !is.na(Income))),
            Young = sum(Age < 14),
            Income = sum(Income,na.rm=TRUE),
            AS_Income = sum(AS_Income, na.rm = TRUE)) %>%
  filter(Hhold_Size <= 15 & Old > 0) %>%
  rename(address = snz_idi_address_register_uid)

#---------- Diagnostic testing
if(diagnostics){
  hhold_income_geo <- left_join(hhold_income, geo, by = "Meshblock")


  hhold_count <- hhold_income_geo %>% group_by(Area) %>%
    summarise(People = sum(Hhold_Size), Addresses = n(), Mean_Income =
mean(Income))

  write.csv(hhold_count, "Diagnostics/HHolds and income.csv", row.names =
FALSE)
}
```

HAM Loop Costs

```
#---------- HAM Housing Cost Estimation
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function, Runs Inside Loop
#---------- Created by James Kerr on 2019-05-01

# NOTE: i denotes the quarter the data is being run on.

#---------- Rent and Bond at Period
latest_rent <- bonds %>% filter(lodged < YearEnd &
                                 (closed >= YearEnd | is.na(closed)) &
                                 (dbh_bond_tenancy_end_date >= YearEnd |
is.na(dbh_bond_tenancy_end_date))) %>%
  group_by(address) %>%
  summarise(lodged = max(lodged))

rents <- inner_join(latest_rent,bonds,by=c("lodged","address")) %>%
  mutate(bond_age = as.numeric(YearStart - lodged)) %>%
  select(-lodged,-closed)

output <- left_join(hhold_income,rents,by="address") %>%
  mutate(Tenure = ifelse(is.na(bond),"Own","Rent")) %>%
  select(-bond)

#---------- Subset to target population
output <- output %>% filter(Tenure == "Rent")

# Assign time period
output <- output %>% mutate(Quarter = periods[i])


# Apply territorial authority / ward
output <- left_join(output, geo, by="Meshblock")

#---------- Diagnostic testing
if(diagnostics){
  rental_test <- output %>% group_by(Area) %>%
    summarise(People = sum(Hhold_Size), Addresses = n(), Mean_Income =
mean(Income))

  write.csv(rental_test,"Diagnostics/Rentals.csv", row.names = FALSE)
}

#---------- HAM Buy Affordability
# Evaluer for Quarter
price <- filter(evaluer, format.Date(Date,format = "%Y-%m") == periods[i])
# First 7 characters of Date gives YYYY-MM format

# Most Recent Rateable Value
  rate_value <- filter(rateable_values, Date <= years[i]) # Discard data
from future years

  # Find higest year for each Area
  rate_value <- rate_value %>%
    group_by(Area, Beds) %>%
    mutate(Max_Year = max(Date)) %>%
    filter(Date == Max_Year) %>%
    ungroup()

# Calculate Buy Cost
```

```
buy_cost <- full_join(price, rate_value, by = c("Area", "Beds"), suffix =
c("_EV","_RV")) %>%
  select(Area, Beds, LQ_Price = LQ_Price_EV, LQ_RV = LQ_Price_RV) %>%
  mutate(Mort_Cost =  (LQ_Price*mortgage_int)/(1-(1+mortgage_int)^-30),
         Insurance_Cost = Insurance_rate * LQ_Price,
         Rates_Cost = Rates_rate * LQ_RV,
         Buy_Cost = Mort_Cost + Insurance_Cost + Rates_Cost)

# Group Households into bedroom groups and apply buy costs
output <- output %>%
  mutate(Beds = bedrooms,
         Beds = ifelse(bedrooms %in% 1:2, "1+2", Beds),
         Beds = ifelse(bedrooms %in% c(4, "5+"), "4+", Beds)) %>%
  left_join(buy_cost, by = c("Area","Beds"))

##### TESTING - apply old method for comparison
output <- output %>%
  left_join(filter(buy_cost, Beds == "1+2"), by = "Area", suffix =
c("","_Old"))


#---------- Diagnostic testing
if(diagnostics){
  buy_test <- output %>% group_by(Area) %>%
    summarise(People = sum(Hhold_Size), Addresses = n(), Mean_Income =
mean(Income), No_Buy = sum(is.na(Buy_Cost)))

  write.csv(buy_test,"Diagnostics/Buy Costs.csv", row.names = FALSE)
}

#---------- Convert Rent to Annual
output <- output %>% mutate(Housing_Cost = rent * 52) # 52 weeks per year
```

HAM Loop Affordability

```
#---------- HAM Affordability Calculation
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function, Runs Inside Loop
#---------- Created by James Kerr on 2019-05-02

# NOTE: i denotes the quarter the data is being run on.

#---------- Flag and Remove Household with Incomes of 0 or less
output$No_Income <- output$Income <= 0
output$Income[output$No_Income] <- NA

#---------- Calculate Residual and Eqivalised incomes
output <- output %>% mutate(Equiv = 1 + 0.5*(Old-1) + 0.3*Young, # 1 for
1st adult, 0.5 for each other adult, 0.3 per child
                            Resid_income = Income - Housing_Cost,
                            Resid_Equiv = Resid_income / Equiv)

#---------- Calculate Nominal Affordability Thresholds
nom_thresholds <-
thresholds_2013$Threshold[thresholds_2013$Base_Year=="2012/13"]*fin$CPI[i]/
CPI2013

#---------- Calculate HAM Values
output <- output %>%
  mutate(HAM_Rent_Median = ifelse(Tenure == "Rent" & !is.na(Resid_Equiv),
                                  ifelse(Resid_Equiv < nom_thresholds[3],
"Below", "Above"),
                                  NA),
         HAM_Rent_10pc = ifelse(Tenure == "Rent" & !is.na(Resid_Equiv),
                                 ifelse(Resid_Equiv < nom_thresholds[1],
"Below", "Above"),
                                 NA),
         HAM_Rent_percent = ifelse(Tenure == "Rent",
                                    ifelse((Housing_Cost-AS_Income) /
(Income-AS_Income) < 0.3, "Less", "More"),
                                    NA),
         HAM_Buy_Median = ifelse(Tenure == "Rent" & !is.na(Resid_Equiv),
                                  ifelse((Income-Buy_Cost)/Equiv <
nom_thresholds[3], "Below", "Above"),
                                  NA),
         HAM_Buy_10pc = ifelse(Tenure == "Rent" & !is.na(Resid_Equiv),
                                ifelse((Income-Buy_Cost)/Equiv <
nom_thresholds[1], "Below", "Above"),
                                NA),
         HAM_Buy_percent = ifelse(Tenure == "Rent",
                                   ifelse((Buy_Cost-AS_Income) / (Income-
AS_Income) < 0.3, "Less", "More"),
                                   NA),
         HAM_Buy_old = ifelse(Tenure == "Rent",
                               ifelse((Buy_Cost_Old-AS_Income) / ((Income-
AS_Income)/Equiv) < 0.3, "Less", "More"),
                               NA),
         HAM_Buy_bad = ifelse(Tenure == "Rent",
                              ifelse((Buy_Cost_Old-AS_Income) / (Income-
AS_Income) < 0.3, "Less", "More"),
                              NA))

#---------- Remove Invalid Households
```

```
output$Valid_Data <- (!is.na(output$HAM_Rent_Median) |
!is.na(output$HAM_Buy_Median)) & output$Resid_income > 0

output <- filter(output,Valid_Data)

#---------- Apply Random Seeds
output <- left_join(output, seed_table, by = "address")

#---------- Write File
output <- ungroup(output)
if(write_permitted) {save(output, file = paste0("Unit Record/HAM Microdata
","Quarterly ",periods[i],".rda"))}
```

HAM Primary Table

```
#---------- HAM primary Table Aggregation Script
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Last Updated by James Kerr on 2019-05-03

# This script aggregates the HAM unit record files into the TA / Ward -
level HAM data
# It also performs the necessary confidentilaisation to make the data fit
for release

#---------- Identify unit-record files
  # Obtain list of Unit Record Files
  files <- list.files(path = "Unit Record")

  # Determine number of unit record files
  quarterly_files <- subset(files, grepl(".rda", files, fixed = TRUE) &
grepl("Quarterly", files, fixed = TRUE))

  #---------- Quarterly Aggregation
  raw_area <- data.frame(NULL)

  for(i in 1:length(quarterly_files)) {
    # Load File
    load(paste0("Unit Record/",quarterly_files[i]))

    # Aggregate
    quarter_label <- periods[i]

    region_quarterly <- output %>%
      filter(!is.na(Area)) %>%
      group_by(Area) %>%
      summarise(Quarter = quarter_label,
                Count_Rent = sum(!is.na(HAM_Rent_Median)),
                Count_Buy = sum(!is.na(HAM_Buy_Median)),

                HAM_Rent_Med = sum(HAM_Rent_Median == "Below", na.rm=TRUE),
                HAM_Rent_10 = sum(HAM_Rent_10pc == "Below", na.rm=TRUE),
                HAM_Rent_PC = sum(HAM_Rent_percent == "More", na.rm=TRUE),

                HAM_Buy_Med = sum(HAM_Buy_Median == "Below", na.rm=TRUE),
                HAM_Buy_10 = sum(HAM_Buy_10pc == "Below", na.rm=TRUE),
                HAM_Buy_PC = sum(HAM_Buy_percent == "More", na.rm=TRUE),

                MOD_Rent =
sum(ifelse(!is.na(HAM_Rent_Median),Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy = sum(ifelse(!is.na(HAM_Buy_Median),Random_Seed,0),
na.rm=TRUE) %% 1,

                MOD_Rent_Med = sum(ifelse(HAM_Rent_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_10 = sum(ifelse(HAM_Rent_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_PC = sum(ifelse(HAM_Rent_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1,

                MOD_Buy_Med = sum(ifelse(HAM_Buy_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_10 = sum(ifelse(HAM_Buy_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_PC = sum(ifelse(HAM_Buy_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1)

    akl_quarterly <- output %>%
      filter(grepl("Auckland",Area,fixed=TRUE)) %>%
      summarise(Area = "Auckland Total", Quarter = quarter_label,
                Count_Rent = sum(!is.na(HAM_Rent_Median)),
                Count_Buy = sum(!is.na(HAM_Buy_Median)),

                HAM_Rent_Med = sum(HAM_Rent_Median == "Below", na.rm=TRUE),
                HAM_Rent_10 = sum(HAM_Rent_10pc == "Below", na.rm=TRUE),
                HAM_Rent_PC = sum(HAM_Rent_percent == "More", na.rm=TRUE),

                HAM_Buy_Med = sum(HAM_Buy_Median == "Below", na.rm=TRUE),
                HAM_Buy_10 = sum(HAM_Buy_10pc == "Below", na.rm=TRUE),
                HAM_Buy_PC = sum(HAM_Buy_percent == "More", na.rm=TRUE),

                MOD_Rent =
sum(ifelse(!is.na(HAM_Rent_Median),Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy = sum(ifelse(!is.na(HAM_Buy_Median),Random_Seed,0),
na.rm=TRUE) %% 1,

                MOD_Rent_Med = sum(ifelse(HAM_Rent_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_10 = sum(ifelse(HAM_Rent_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_PC = sum(ifelse(HAM_Rent_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1,

                MOD_Buy_Med = sum(ifelse(HAM_Buy_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_10 = sum(ifelse(HAM_Buy_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_PC = sum(ifelse(HAM_Buy_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1)

    nz_quarterly <- output %>%
      summarise(Area = "National Total", Quarter = quarter_label,
                Count_Rent = sum(!is.na(HAM_Rent_Median)),
                Count_Buy = sum(!is.na(HAM_Buy_Median)),

                HAM_Rent_Med = sum(HAM_Rent_Median == "Below", na.rm=TRUE),
                HAM_Rent_10 = sum(HAM_Rent_10pc == "Below", na.rm=TRUE),
                HAM_Rent_PC = sum(HAM_Rent_percent == "More", na.rm=TRUE),

                HAM_Buy_Med = sum(HAM_Buy_Median == "Below", na.rm=TRUE),
                HAM_Buy_10 = sum(HAM_Buy_10pc == "Below", na.rm=TRUE),
                HAM_Buy_PC = sum(HAM_Buy_percent == "More", na.rm=TRUE),

                MOD_Rent =
sum(ifelse(!is.na(HAM_Rent_Median),Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy = sum(ifelse(!is.na(HAM_Buy_Median),Random_Seed,0),
na.rm=TRUE) %% 1,

                MOD_Rent_Med = sum(ifelse(HAM_Rent_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_10 = sum(ifelse(HAM_Rent_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Rent_PC = sum(ifelse(HAM_Rent_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_Med = sum(ifelse(HAM_Buy_Median ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_10 = sum(ifelse(HAM_Buy_10pc ==
"Below",Random_Seed,0), na.rm=TRUE) %% 1,
                MOD_Buy_PC = sum(ifelse(HAM_Buy_percent ==
"More",Random_Seed,0), na.rm=TRUE) %% 1)

    raw_area <- bind_rows(raw_area, region_quarterly, akl_quarterly,
nz_quarterly)
    print(c(quarter_label,as.character(Sys.time())))
  }

# Remove Chathan Islands Territory
  raw_area <- raw_area %>% filter(Area != "Chatham Islands Territory")

# Output File
  if(write_permitted) {write.csv(raw_area, file = "Output/raw_area.csv",
row.names = FALSE)}

#---------- CONFIDENTIALISATION
confid_area <- raw_area

# Identify column names for each type of HAM measure (controls which
columns get suppressed)
rent_columns <- confid_area %>% select(contains("Rent")) %>% names()
buy_columns <- confid_area %>% select(contains("Buy")) %>% names()

# Suppress where Count is less than 6
confid_area[confid_area$Count_Rent < 6, rent_columns] <- NA
confid_area[confid_area$Count_Buy < 6, buy_columns] <- NA

# Secondary suppression
time_series <- as.character(levels(factor(confid_area$Quarter)))
for(i in 1:length(time_series)) {
  period_set <- confid_area %>% filter(Quarter == periods[i])
  akl_set <- period_set %>%
filter(grepl("Auckland",period_set$Area,fixed=TRUE))
  NZ_suppress_Rent <- sum(is.na(period_set$Count_Rent)) == 1
  AKL_suppress_Rent <- sum(is.na(akl_set$Count_Rent)) == 1

  NZ_suppress_Buy <- sum(is.na(period_set$Count_Buy)) == 1
  AKL_suppress_Buy <- sum(is.na(akl_set$Count_Buy)) == 1

  # Identify largest unrepressed area in country and in Auckland
  NZ_target_Rent <- period_set$Area[period_set$Count_Rent ==
min(period_set$Count_Rent)]
  AKL_target_Rent <- akl_set$Area[akl_set$Count_Rent ==
min(akl_set$Count_Rent)]

  NZ_target_Buy <- period_set$Area[period_set$Count_Buy ==
min(period_set$Count_Buy)]
  AKL_target_Buy <- akl_set$Area[akl_set$Count_Buy ==
min(akl_set$Count_Buy)]

  # Apply Suppression, if required
  if(NZ_suppress_Rent){confid_area[confid_area$Area == NZ_target_Rent &
Quarter == periods[i],rent_columns] <- NA}
  if(AKL_suppress_Rent){confid_area[confid_area$Area == AKL_target_Rent &
Quarter == periods[i],rent_columns] <- NA}
```

```
  if(NZ_suppress_Buy){confid_area[confid_area$Area == NZ_target_Buy &
Quarter == periods[i],buy_columns] <- NA}
  if(AKL_suppress_Buy){confid_area[confid_area$Area == AKL_target_Buy &
Quarter == periods[i],buy_columns] <- NA}

}

# Random Base 3 Round counts - uses a seed to ensure consistency
base3_round <- function(data, seed){

  # Round Up and Down to nearest multiple of 3
  roundup <- ceiling(data/3)*3
  rounddown <- floor(data/3)*3

  # Is the nearest round up or down? (if value is multiple of 3
nearest_round_up and nearest_round_down will be TRUE)
  nearest_round_up <- abs(roundup - data) < 2
  nearest_round_down <- abs(rounddown - data) < 2

  # Identify the nearest and second-nearest round (if value is multiple of
3, both rounds will be the same)
  nearest_round <- ifelse(nearest_round_up, roundup, rounddown)
  second_round <- ifelse(nearest_round_up, rounddown, roundup)

  # 2/3 chance (based on seed of nearest round)
  rounded <- ifelse(seed < 2/3, nearest_round, second_round)

  return(rounded)

}

# Random Base 3 All Counts
confid_area <- confid_area %>%
  mutate(Count_Rent = base3_round(Count_Rent, MOD_Rent),
         Count_Buy = base3_round(Count_Buy, MOD_Buy),

         HAM_Rent_Med = base3_round(HAM_Rent_Med, MOD_Rent_Med),
         HAM_Rent_10 = base3_round(HAM_Rent_10, MOD_Rent_10),
         HAM_Rent_PC = base3_round(HAM_Rent_PC, MOD_Rent_PC),

         HAM_Buy_Med = base3_round(HAM_Buy_Med, MOD_Buy_Med),
         HAM_Buy_10 = base3_round(HAM_Buy_10, MOD_Buy_10),
         HAM_Buy_PC = base3_round(HAM_Buy_PC, MOD_Buy_PC))

# Remove MOD Columns
confid_area <- confid_area %>%
  select(-contains("MOD_"))

# Recalculate HAM as Proportions
confid_area <- mutate(confid_area,
                      HAM_Rent_Med = HAM_Rent_Med/Count_Rent,
                      HAM_Rent_10 = HAM_Rent_10/Count_Rent,
                      HAM_Rent_PC = HAM_Rent_PC/Count_Rent,

                      HAM_Buy_Med = HAM_Buy_Med/Count_Buy,
                      HAM_Buy_10 = HAM_Buy_10/Count_Buy,
                      HAM_Buy_PC = HAM_Buy_PC/Count_Buy)

# Refactor to force proportions above 1 back to 1
confid_area$HAM_Rent_Med[confid_area$HAM_Rent_Med > 1] <- 1
confid_area$HAM_Rent_10[confid_area$HAM_Rent_10 > 1] <- 1
```

```
confid_area$HAM_Rent_PC[confid_area$HAM_Rent_PC > 1] <- 1

confid_area$HAM_Buy_Med[confid_area$HAM_Buy_Med > 1] <- 1
confid_area$HAM_Buy_10[confid_area$HAM_Buy_10 > 1] <- 1
confid_area$HAM_Buy_PC[confid_area$HAM_Buy_PC > 1] <- 1


if(write_permitted) {write.csv(confid_area,"Output/HAM
Confidentialised.csv",row.names=FALSE)}
```

HAM Household Income

```
#---------- HAM Household Income
#---------- For HAM 1.4
#---------- NOTE - Requires HAM Master Script to Function
#---------- Last Updated by James Kerr on 2019-05-27

#---------- Identify unit-record files
# Obtain list of Unit Record Files
files <- list.files(path = "Unit Record")

# Determine number of household mapping files
hhold_files <- subset(files, grepl(".rda", files, fixed = TRUE) &
grepl("Household Map", files, fixed = TRUE))

#---------- Construct Household Incomes by quarter
# Create Empty table
hhold_income <- data.frame(NULL)

for(i in 1:length(periods)){

  #---------- Define required Dates and figures
  YearEnd <-
as.Date(paste(ifelse(months[i]==12,years[i]+1,years[i]),ifelse(months[i]==1
2,1,months[i]+1),1,sep="-")) # First day past end of year

  # Load File
  load(paste0("Unit Record/",hhold_files[i]))

  # Aggregate income to household level
  hhold_working <- person_char %>%
    filter(Pass_All_Filters) %>%
    group_by(snz_idi_address_register_uid, Meshblock) %>%
    summarise(Hhold_Size = n(),
              Income_Count = sum(Income > 0 & !is.na(Income)),
              Old = sum((Age >= 14) | (Income > 0 & !is.na(Income))),
              Young = sum(Age < 14),
              Income = sum(Income,na.rm=TRUE)) %>%
    filter(Hhold_Size <= 15, Old > 0, Income > 0) %>%
    rename(address = snz_idi_address_register_uid) %>%
    ungroup()

  # Apply Geography info
  hhold_working <- left_join(hhold_working, geo, by = "Meshblock")

  # Apply Random Seeds
  hhold_working <- left_join(hhold_working, seed_table, by = "address")

  # Collect Bond Data (for Tenure)
  latest_bond <- bonds %>% filter(lodged < YearEnd &
                                  (closed >= YearEnd | is.na(closed)) &
                                  (dbh_bond_tenancy_end_date >= YearEnd |
is.na(dbh_bond_tenancy_end_date))) %>%
    group_by(address) %>%
    summarise(lodged = max(lodged))

  hhold_working <- left_join(hhold_working,latest_bond,by="address") %>%
    mutate(Tenure = ifelse(is.na(lodged),"Own","Rent")) %>%
    select(-lodged)

  # Aggregate income to area level
  income_regional_all <- hhold_working %>%
    filter(!is.na(Income)) %>%
    group_by(Area) %>%
    summarise(Quarter = periods[i], Tenure = "All Households",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  income_regional_rent <- hhold_working %>%
    filter(!is.na(Income), Tenure == "Rent") %>%
    group_by(Area) %>%
    summarise(Quarter = periods[i], Tenure = "Renters",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  income_auckland_all <- hhold_working %>%
    filter(!is.na(Income), grepl("Auckland:", Area, fixed = TRUE)) %>%
    summarise(Quarter = periods[i], Tenure = "All Households", Area =
"Auckland Total",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  income_auckland_rent <- hhold_working %>%
    filter(!is.na(Income), Tenure == "Rent", grepl("Auckland:", Area, fixed
= TRUE)) %>%
    summarise(Quarter = periods[i], Tenure = "Renters", Area = "Auckland
Total",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  income_nz_all <- hhold_working %>%
    filter(!is.na(Income)) %>%
    summarise(Quarter = periods[i], Tenure = "All Households", Area =
"National Total",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  income_nz_rent <- hhold_working %>%
    filter(!is.na(Income), Tenure == "Rent") %>%
    summarise(Quarter = periods[i], Tenure = "Renters", Area = "National
Total",
              Count = n(),
              Median = quantile(Income, probs = 0.5, na.rm=TRUE),
              Percent_20 = quantile(Income, probs = 0.2, na.rm=TRUE),
              Percent_40 = quantile(Income, probs = 0.4, na.rm=TRUE),
              Percent_60 = quantile(Income, probs = 0.6, na.rm=TRUE),
              Percent_80 = quantile(Income, probs = 0.8, na.rm=TRUE),
              Mean = mean(Income, na.rm=TRUE),
              MOD = sum(Random_Seed) %% 1)

  hhold_income <- bind_rows(hhold_income,
                            income_regional_all, income_regional_rent,
                            income_auckland_all, income_auckland_rent,
                            income_nz_all, income_nz_rent)

  print(c(periods[i],as.character(Sys.time())))

}

hhold_income <- filter(hhold_income, !Area %in% c("Chatham Islands
Territory", "Area Outside territorial authority"), !is.na(Area))

if(write_permitted){write.csv(hhold_income, "Output/raw_income.csv",
row.names = FALSE)}

#---------- CONFIDENTIALISATION
confid_income <- hhold_income

# Suppress by column
confid_income <- confid_income %>%
  mutate(Mean = ifelse(Count < 20, NA, Mean), # Rule 5.4.2 suppress if
count is less than 20
         Median = ifelse(Count < 10, NA, Median), # Rule 5.5.1 suppress if
count is less than 10
         Percent_20 = ifelse(Count < 25, NA, Percent_20), # Rule 5.5.1
suppress if count is less than 25
         Percent_40 = ifelse(Count < 12.5, NA, Percent_40), # Rule 5.5.1
suppress if count is less than 12.5
         Percent_60 = ifelse(Count < 12.5, NA, Percent_60), # Rule 5.5.1
suppress if count is less than 12.5
         Percent_80 = ifelse(Count < 25, NA, Percent_80), # Rule 5.5.1
suppress if count is less than 25
         Count = ifelse(Count < 6, NA, Count) # Rule 5.15 suppress if count
is less than 6
         )

# Random Round Counts
confid_income$Count <- base3_round(confid_income$Count, confid_income$MOD)
confid_income$MOD <- NULL

if(write_permitted){write.csv(confid_income, "Output/Household Income
Confidentialised.csv", row.names = FALSE)}
```

Appendix D: Disclaimers

The results in this paper are an experimental series produced by the Ministry of Business Innovation and Employment. They are not a Tier One Official Statistic, nor have they been endorsed by Stats NZ. They have been created for research purposes from the Integrated Data Infrastructure (IDI), managed by Stats NZ.

The opinions, findings, recommendations, and conclusions expressed in this paper are those of the author(s), not Statistics NZ.

Access to the anonymised data used in this study was provided by Statistics NZ in accordance with security and confidentiality provisions of the Statistics Act 1975. Only people authorised by the Statistics Act 1975 are allowed to see data about a particular person, household, business, or organisation, and the results in this paper have been confidentialised to protect these groups from identification.

Careful consideration has been given to the privacy, security, and confidentiality issues associated with using administrative and survey data in the IDI. Further detail can be found in the Privacy impact assessment for the Integrated Data Infrastructure available from www.stats.govt.nz.

The results are based in part on tax data supplied by Inland Revenue to Statistics NZ under the Tax Administration Act 1994. This tax data must be used only for statistical purposes, and no individual information may be published or disclosed in any other form, or provided to Inland Revenue for administrative or regulatory purposes.

Any person who has had access to the unit record data has certified that they have been shown, have read, and have understood section 81 of the Tax Administration Act 1994, which relates to secrecy. Any discussion of data limitations or weaknesses is in the context of using the IDI for statistical purposes, and is not related to the data’s ability to support Inland Revenue’s core operational requirements