Table of Contents

1.0 Introduction

Prior to building a model for machine learning purposes, multiple steps must first be undertaken. Among the steps include the thorough understanding of the dataset employed as well as the preparing of the dataset so that further and meaningful analysis could be carried out from there onwards. The two steps mentioned are crucial and indispensable in the data analysis process because it is important to note that with garbage in comes garbage out. That said, this paper's primary objective is to perform the necessary initial data exploration process on the chosen dataset followed by the discussion on the concepts of data warehousing using real-life examples to better illustrate the discussion. The dataset employed in this paper is taken from Harlfoxem (2016) where the dataset contains the prices of houses sold within King Country, Washington as well as the respective features of those houses, during the period between May 2014 to May 2015.

The sections below follow the sequence of firstly, the assessment of the metadata of the chosen datatset, followed by the initial data exploration, the discussion on the concepts of data warehousing and lastly, a conclusion to sum the paper up. The SAS studio will be used extensively to achieve the said objective of this paper. Nevertheless, because the dataset obtained from Harlfoxem (2016) is too large (21613 observations with 21 attributes) for SAS Studio to handle, the dataset is hence reduced to only contain 3250 observations while maintaining all 21 attributes of the initial dataset. Also, because the dataset does not inherently contain any missing values, 2% out of the 3250 observations were then converted into missing values so as to be able demonstrate the application of the treatment of missing values within a dataset. The processes of reducing the size of the dataset as well as the introduction of missing values into the dataset were both done using another program called RStudio (*Refer to Appendix 1 for the RScript*).

2.0 Metadata of the Dataset

Given that metadata is data about data, looking into it is thereby essential because it provides necessary descriptions about the content of the particular dataset, which would then help users to better understand what the dataset represents, the structure of the dataset as well as its relevance. Besides that, knowing the metadata is also important in that when users understand the scale, units or limitations of the attributes within the dataset, it is only then that data could be used appropriately and efficiently to facilitate for accurate analyses.

Source Code:

```
14 *Printing the Metadata;
15 proc contents data = house;
16 run;
```

Output:

The CONTENTS Procedure

Data Set Name	WORK.HOUSE	Observations	3250
Member Type	DATA	Variables	21
Engine	V9	Indexes	0
Created	09/29/2023 22:47:42	Observation Length	176
Last Modified	09/29/2023 22:47:42	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	5
First Data Page	1
Max Obs per Page	743
Obs in First Data Page	718
Number of Data Set Repairs	0
Filename	/saswork/SAS_work29F00001051A_odaws01-apse1-2.oda.sas.com/SAS_work5C520001051A_odaws01-apse1-2.oda.sas.com/house.sas7bdat
Release Created	9.0401M7
Host Created	Linux
Inode Number	67108945
Access Permission	rw-r--r--
Owner Name	u63524281
File Size	768KB
File Size (bytes)	786432

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
5	bathrooms	Char	4	$4.	$4.
4	bedrooms	Num	8	BEST12.	BEST32.
11	condition	Num	8	BEST12.	BEST32.
2	date	Char	17	$17.	$17.
8	floors	Num	8	BEST12.	BEST32.
12	grade	Num	8	BEST12.	BEST32.
1	id	Num	8	BEST12.	BEST32.
18	lat	Num	8	BEST12.	BEST32.
19	long	Num	8	BEST12.	BEST32.
3	price	Num	8	BEST12.	BEST32.
13	sqft_above	Num	8	BEST12.	BEST32.
14	sqft_basement	Num	8	BEST12.	BEST32.
6	sqft_living	Num	8	BEST12.	BEST32.
20	sqft_living15	Num	8	BEST12.	BEST32.
7	sqft_lot	Num	8	BEST12.	BEST32.
21	sqft_lot15	Num	8	BEST12.	BEST32.
10	view	Num	8	BEST12.	BEST32.
9	waterfront	Num	8	BEST12.	BEST32.
15	yr_built	Num	8	BEST12.	BEST32.
16	yr_renovated	Num	8	BEST12.	BEST32.
17	zipcode	Num	8	BEST12.	BEST32.

From the metadata above, the name of the dataset is “HOUSE” and the dataset is located within the “WORK” library. There are in total 3250 observations (rows) and 21 variables (columns). In terms of the Engine/Host Dependent Information, it is an especially important piece of information when users are required to transfer datasets between different systems or when performing troubleshooting of issues related to data access or storage. Basically, such information provides insights regarding the physical and system-specific aspects of the dataset. The 21 variables within the dataset are namely *bathrooms*, *bedrooms*, *condition*, *date*, *floors*, *grade*, *id*, *lat*, *long*, *price*, *sqft_above*, *sqft_basement*, *sqft_living*, *sqft_living15*, *sqft_lot*, *sqft_lot15*, *view*, *waterfront*, *yr_built*, *yr_renovated*, and *zipcode*. The *bathroom* and *date* variables are of character type while the rest of the variables are of numeric type.

Having observed that, even though the metadata does tell users of the type of data for which each attribute belong to, users must know that it is possible however for the attributes to be categorized into the wrong data types. For instance, making reference back to the respective description of each variable (*Appendix 2*), intuition would suggest that the bathroom variable should be of numeric type rather than of character type. The *zipcode* variable on the other hand,

should be of character type instead of what was suggested by the metadata, on grounds that it does not have any mathematical meaning to it. Zip codes are there to only inform of a particular geographical area and does not contain any quantitative values to it. Hence, for the *zipcode* variable, conversion of its data type from numeric to character should be done. Similar reasonings apply to that of the *id* variable. Needless to say, because the paper is only concerned with the initial data exploration process, such conversion discussed would thereby not be discussed nor performed here. That said, it is still important to take note of the inaccuracies in the categorization of data types of the variables.

3.0 Initial Data Exploration

Variable	Level of Measurement
bathrooms	Ratio
bedrooms	Ratio
condition	Ordinal
date	Interval
floors	Ratio
grade	Ordinal
id	Nominal
lat	Interval
long	Interval
price	Ratio
sqft_above	Ratio
sqft_basement	Ratio
sqft_living	Ratio
sqft_living15	Ratio
sqft_lot	Ratio
sqft_lot15	Ratio
view	Ordinal
waterfront	Nominal
yr_built	Interval
yr_renovated	Interval
zipcode	Nominal

In this section, information on the characteristics and components of the data within the dataset will be visually represented. The types of data together with the summarizing properties namely the spread, distribution, median, mean, variance and percentile of each attribute within the dataset would also be pointed out. Detection of outliers, missing values together with any

inconsistencies within the dataset will too be documented under this section. The level of measurement for each variable is outlined in the table above too.

3.1 Summarizing Properties

Before diving into the respective summarizing properties of each variable, it is important to note that for the *id* (numeric type), *bathrooms* (character type) and *zipcode* (numeric type) variables, initial data exploration will not be performed because performing so would only lead to a misleading analysis as all three of these variables are still yet to be in the correct data type. That is, for instance, if the *id* variable is still in its initial numeric type, its summarizing properties such as mean would prove to be meaningless because one, the *id* variable is a unique value meant to represent a particular house that has been sold; and two, finding the mean of the variable which involves the adding up of all *id* values and then dividing it with the number of total instances would prove to be pointless because a mean value of 4591730690 for instance will not mean anything nor will it provide insights into the properties of the variable itself. With that, only variables with the correct data type would be examined under this section. Nevertheless, the three variables would still be converted into their rightful type followed by their respective initial data exploration when the process of feature engineering is done. Besides that, for the *yr_renovated* variable, there are inconsistencies in that for certain observations, the *yr_renovated* was indicated as '0'. Intuition tells that '0' was meant to indicate that no renovation was done, but '0' in the context of year would prove to be inappropriate. Hence, necessary feature selection must be done to correct this. With that, it is important to note that the necessary processes explained above for all five variables will be conducted and discussed in another separate paper and not in this paper instead.

price

Source Code:

```
22 /* price VARIABLE */
23 *Compute quartiles for the 'price' variable;
24 PROC UNIVARIATE DATA=house;
25     VAR price;
26     OUTPUT OUT=OutliersPrice (RENAME=(price=OriginalPrice))
27     Q1=Q1_price Q3=Q3_price;
28 RUN;
```

```
30 *Detect and store outliers for 'price' in the OutliersListPrice dataset;
31 DATA OutliersListPrice (keep=ObsNum OutlierValue);
32     SET house;
33     IF _N_ = 1 THEN SET OutliersPrice;
34 
35     IQR = Q3_price - Q1_price;
36     LowerBound = Q1_price - 1.5 * IQR;
37     UpperBound = Q3_price + 1.5 * IQR;
38 
39     /* Check if the price value is an outlier */
40     IF price < LowerBound OR price > UpperBound THEN DO;
41         ObsNum = _N_;
42         OutlierValue = price;
43         OUTPUT;
44     END;
45 
46     DROP IQR LowerBound UpperBound Q1_price Q3_price;
47 RUN;
48 
49 *Print detected outliers;
50 PROC PRINT DATA=OutliersListPrice;
51 RUN;
52 
53 *Visualize 'price' distribution with a boxplot;
54 PROC SGPLOT DATA=house;
55     VBOX price;
56 RUN;
```

Output:

Moments			
N	3248	Sum Weights	3248
Mean	543405.948	Sum Observations	1764982518
Std Deviation	368956.275	Variance	1.36129E11
Skewness	3.92496603	Kurtosis	29.3056602
Uncorrected SS	1.40111E15	Corrected SS	4.4201E14
Coeff Variation	67.8969886	Std Error Mean	6473.91347

Basic Statistical Measures			
Location		Variability	
Mean	543405.9	Std Deviation	368956
Median	459975.0	Variance	1.36129E11
Mode	375000.0	Range	5218000
		Interquartile Range	326600

Quantiles (Definition 5)	
Level	Quantile
100% Max	5300000
99%	1950000
95%	1160000
90%	880000
75% Q3	650000
50% Median	459975
25% Q1	323400
10%	245000
5%	210000
1%	160000
0% Min	82000

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
82000	271	3420000	2593
83000	377	4000000	2274
85000	2709	4210000	3226
102500	1361	5110000	235
110000	1931	5300000	432

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	2	0.06	100.00

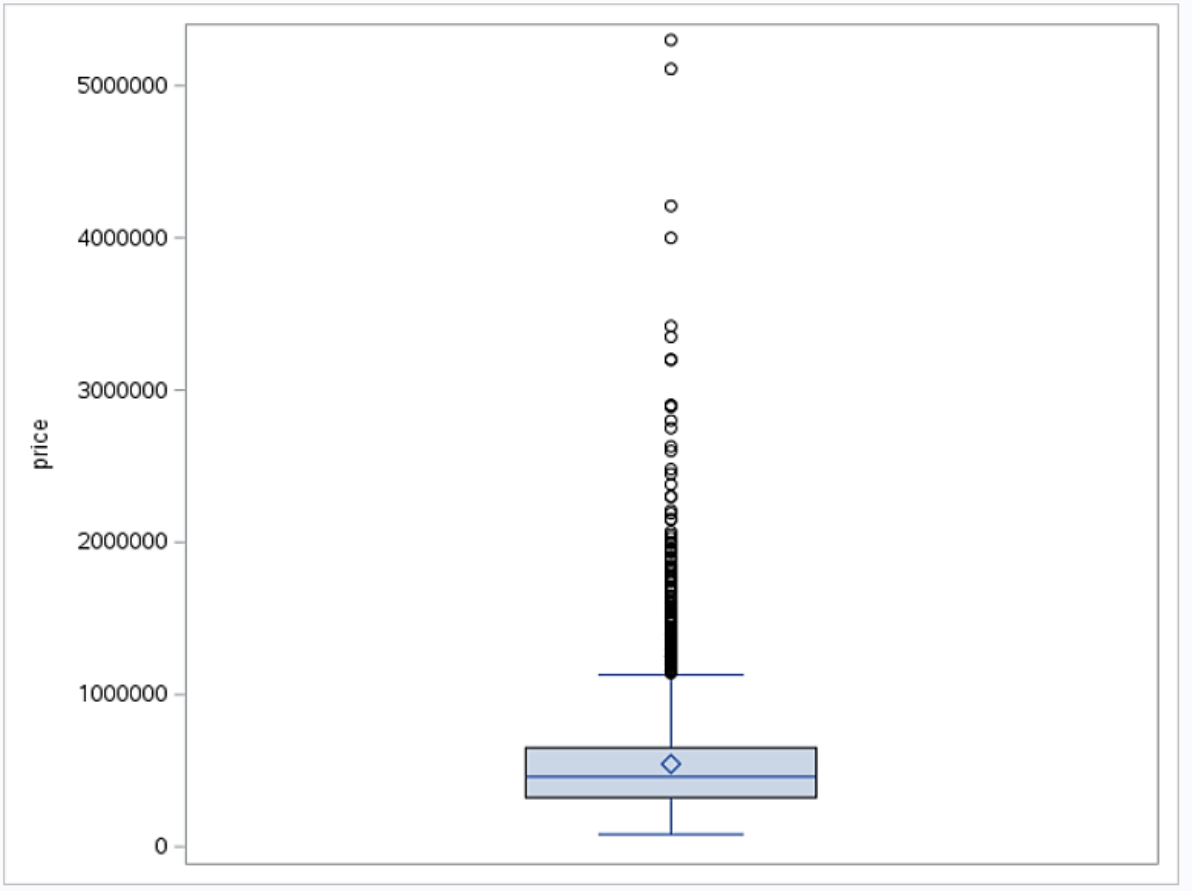
Summarizing Properties:

- There are 2 missing values.
- On average, the price of houses sold is $543405.85.
- There is a considerable spread among data points given the high standard deviation.
- The positive skewness indicates that the distribution is not symmetric and is skewed to the right, with several unusually high values. This is especially normal in the context

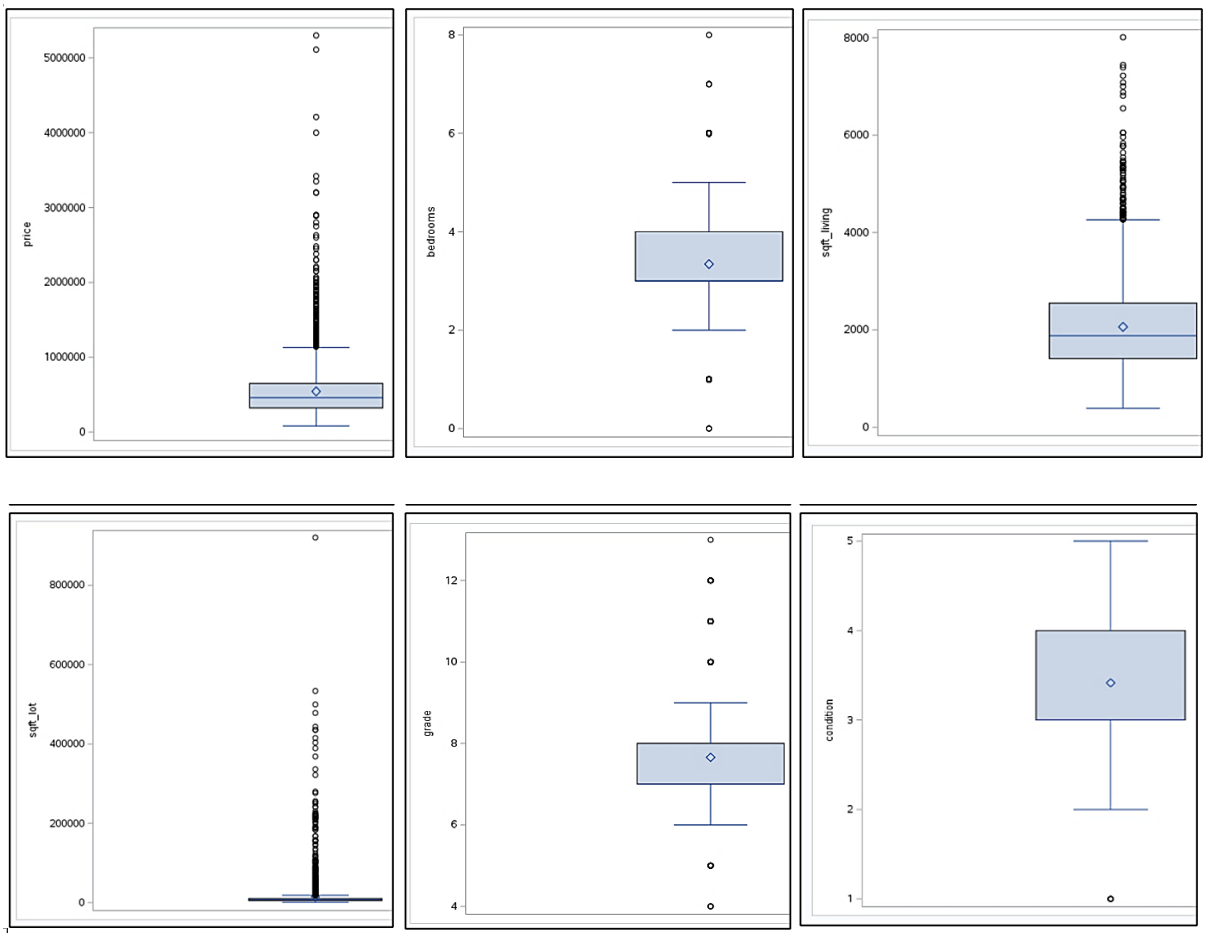
of housing prices due to the presence of luxury and high-end properties as well as the greater provision of affordable houses to cater for the middle class community.

- The high kurtosis suggests the presence of many extreme values or in other words outliers.
- No inconsistencies were detected.

Outliers:



From the plot above, there seems to be a considerable number of large outliers within the *price* variable. Nevertheless, as intuition suggests, housing prices are affected as well by factors such as the number of bedrooms, the square feet of the land space and the square feet of the living interior space to name a few. Hence, to determine if these outliers within the *price* variable are genuine or not, the three variables mentioned above will be examined and compared to the results of the *price* variable. Upon examination of the three variables, the results would once again be crossed check with the *condition*, *view* and *grade* variable for the checking of consistency purposes, as these factors would also influence the pricing of houses for sale.



Comparisons from the boxplots above demonstrate that the outliers for the *price* variable do actually correspond to the outliers for all 5 other variables plotted above. That said, it is reasonable to conclude that all of these outliers including the ones within the *price* variable are genuine outliers and hence must not be removed instead.

bedrooms

Source Code:

```
/* bedrooms VARIABLE */
/* Compute quartiles for the 'bathrooms' variable */
PROC UNIVARIATE DATA=house;
    VAR bedrooms;
    OUTPUT OUT=OutliersBedrooms (RENAME=(bedrooms=OriginalBedrooms))
    Q1=Q1_bedrooms Q3=Q3_bedrooms;
RUN;
```

```
/* Detect and store outliers for 'bathrooms' in a new dataset */
DATA OutliersListBedrooms (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersBedrooms;
    IQR = Q3_bedrooms - Q1_bedrooms;
    LowerBound = Q1_bedrooms - 1.5 * IQR;
    UpperBound = Q3_bedrooms + 1.5 * IQR;
    IF bedrooms < LowerBound OR bedrooms > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = bedrooms;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_bedrooms Q3_bedrooms;
RUN;

/* Print detected outliers */
PROC PRINT DATA=OutliersListBedrooms;
RUN;

/* Visualize 'bedrooms' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX bedrooms;
RUN;
```

```
/* Create a new dataset with the filtered observations */
data filtered_bedrooms;
    set house;
    if bedrooms in (6, 7, 8);
run;

/* Print the filtered observations */
proc print data=filtered_bedrooms;
run;
```

Output:

The UNIVARIATE Procedure
Variable: bedrooms

Moments			
N	3249	Sum Weights	3249
Mean	3.34195137	Sum Observations	10858
Std Deviation	0.89178863	Variance	0.79528695
Skewness	0.33325935	Kurtosis	0.67676398
Uncorrected SS	38870	Corrected SS	2583.09203
Coeff Variation	26.684668	Std Error Mean	0.01564541

Basic Statistical Measures			
Location		Variability	
Mean	3.341951	Std Deviation	0.89179
Median	3.000000	Variance	0.79529
Mode	3.000000	Range	8.00000
		Interquartile Range	1.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	8
99%	6
95%	5
90%	4
75% Q3	4
50% Median	3
25% Q1	3
10%	2
5%	2
1%	2
0% Min	0

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0	2679	6	3174
0	1239	7	7
1	3172	7	1637
1	3159	7	2178
1	3145	8	268

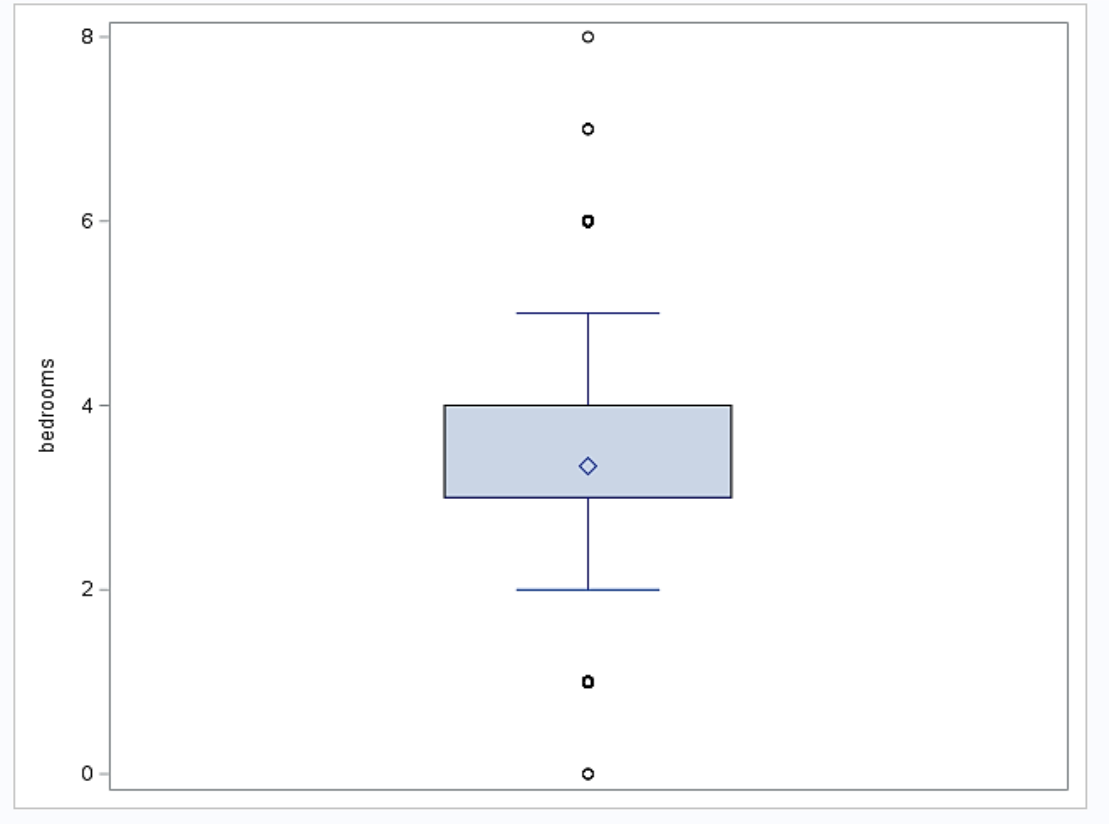
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	1	0.03	100.00

Summarizing Properties:

- There is 1 missing value.
- The mean value indicates that on average, most houses within the dataset has around 3 bedrooms in total.

- The standard deviation of approximately 0.8918 suggests moderate variability in the number of bedrooms among the houses in the dataset
- The positive skewness value of 0.3333 indicates that the distribution is slightly skewed to the right.
- The kurtosis suggests that while most houses have an average number of bedrooms, there are however those that have exceptionally more bedrooms in their houses. Nevertheless, since the kurtosis value is low, these extreme values are though present, are not many in numbers.
- Inconsistency has been detected in that there are 2 observations where the number of bedrooms is 0. Considering that it is unconventional to have houses with no bedrooms, hence making these 2 observations to be illogical, removing them would then appropriate.

Outliers:



Obs	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
1	4178300130	20150413T000000	950000	7	3.5	3470	16264	2	0	0	4	9	3470	0	1980	0	98007	47.6203	-122.149	3040	13500
2	4174600331	20140717T000000	384000	6	3	2320	4502	1	0	0	4	7	1200	1120	1987	0	98108	47.5552	-122.3	1160	5628
3	3693900885	20141202T000000	1020000	6	2.25	2550	5000	2	0	0	4	7	2550	0	1907	0	98117	47.6785	-122.396	1480	5000
4	922059169	20141201T000000	800000	6	4.25	5480	189050	2	0	0	4	10	5140	340	1991	0	98031	47.412	-122.168	2470	10429
5	2024059052	20140814T000000	975000	6	3	3420	22421	1	0	0	5	9	2270	1150	1948	0	98006	47.5508	-122.189	2430	15560
6	913000315	20140605T000000	1300000	6	4.5	3902	3880	3	0	4	4	9	2782	1120	1977	0	98116	47.5837	-122.399	1100	3870
7	2769600590	20141016T000000	900000	8	4	4020	7500	1	0	0	3	8	2010	2010	1968	0	98107	47.6732	-122.363	1560	3737
8	7558700030	20150413T000000	5300000	6	6	7390	24829	2	1	4	4	12	5000	2390	1991	0	98040	47.5631	-122.21	4320	24619
9	125059178	20140722T000000	510000	6	4.5	3300	7480	2	0	0	3	8	3300	0	1980	0	98052	47.6796	-122.104	2470	7561
10	8576400110	20150317T000000	580000	6	2.5	3596	13700	1	0	0	5	8	1798	1798	1964	0	98166	47.4388	-122.339	1894	10500
11	4027700853	20150428T000000	387500	6	2	2400	7684	1	0	0	3	6	1200	1200	1932	2005	98028	47.7705	-122.269	1290	9800
12	1644500050	20150312T000000	875000	6	3.5	4430	11453	2	0	0	3	9	3000	1430	2001	0	98056	47.5156	-122.204	2730	5661
13	6383900090	20140904T000000	838300	6	2.5	3760	12978	1	0	0	3	9	2360	1400	1967	0	98117	47.6976	-122.381	2300	7362
14	8650000250	20140607T000000	611000	6	2.5	3820	53173	1	0	0	4	9	2040	1780	1974	0	98027	47.5209	-122.052	2510	15314
15	7129300400	20140814T000000	400000	6	2	2350	6554	2	0	1	3	8	2000	350	1905	0	98178	47.5115	-122.256	1560	6554
16	9187200285	20140505T000000	823000	6	1.75	2920	5000	2.5	0	0	4	9	2780	140	1908	0	98122	47.6024	-122.295	2020	5000
17	2887703155	20150225T000000	642000	6	1	1530	4305	1.5	0	0	4	7	1530	0	1921	0	98115	47.6862	-122.31	1530	3800
18	7896300592	20150114T000000	303500	6	4.5	3390	7200	2	0	0	3	8	2440	950	2007	0	98118	47.5205	-122.288	2040	7214
19	2224700045	20140804T000000	375000	6	2	1900	8057	1	0	0	4	7	1170	730	1959	0	98133	47.762	-122.335	2090	8626
20	7338402690	20150401T000000	335000	6	2	2020	7071	1	0	0	3	7	1010	1010	1979	0	98108	47.5329	-122.294	2020	5000
21	1562100220	20150501T000000	605000	6	2	2610	9132	1	0	0	4	8	1320	1290	1965	0	98007	47.622	-122.14	2170	8000
22	9265200060	20141007T000000	650000	6	4.5	3900	9100	2	0	0	3	8	2870	1030	1979	0	98052	47.6612	-122.137	2080	9216
23	3834500195	20140707T000000	527000	6	3.5	3000	8401	1	0	0	3	8	1500	1500	1979	0	98125	47.7226	-122.299	1400	8403
24	3225079035	20140618T000000	1600000	6	5	6050	230652	2	0	3	3	11	6050	0	2001	0	98024	47.6033	-121.943	4210	233971
25	2862100366	20141015T000000	730000	7	2.75	3110	4400	1.5	0	0	5	7	2010	1100	1914	0	98105	47.6684	-122.319	1240	4280
26	7856700060	20150413T000000	893880	6	2.5	2820	8600	1	0	0	5	8	1430	1390	1967	0	98006	47.565	-122.144	2070	8900
27	3972900025	20150313T000000	499000	6	1.75	2400	7500	1.5	0	0	3	7	1400	1000	1975	0	98155	47.7661	-122.313	1980	7500
28	9198600035	20140805T000000	240000	6	1.75	2210	8594	1	0	0	3	7	1310	900	1959	0	98188	47.4594	-122.273	1850	8594
29	2771604190	20140617T000000	824000	7	4.25	3670	4000	2	0	1	3	8	2800	870	1964	0	98199	47.6375	-122.388	2010	4000
30	3232200085	20150428T000000	1500000	6	3.5	3670	3959	2	0	0	3	10	2410	1260	2008	0	98119	47.6356	-122.373	2060	3625
31	8941100095	20140923T000000	1110000	6	4	3600	6224	2	0	0	3	9	2610	990	1945	2006	98199	47.6531	-122.405	1430	6224
32	4037200075	20140911T000000	662500	6	2.25	2450	25600	1	0	2	3	7	1340	1110	1957	0	98008	47.6061	-122.117	1850	10230
33	8820903380	20140728T000000	452000	6	2.25	2660	13579	2	0	0	3	7	2660	0	1937	1990	98125	47.7142	-122.286	1120	8242
34	1994200260	20140819T000000	869900	6	4.5	2750	4400	2	0	0	3	8	1770	980	1987	0	98103	47.6883	-122.335	1860	4400
35	1169000057	20141006T000000	1130000	6	4.25	3100	9378	3	0	2	3	11	3100	0	1978	0	98112	47.6381	-122.314	3270	6334
36	8917100180	20140604T000000	583000	6	2.75	2630	16411	1	0	0	4	8	1650	980	1974	0	98052	47.6309	-122.093	2250	12255
37	1568100295	20150202T000000	592500	6	4.5	3500	8504	2	0	0	3	7	3500	0	1980	0	98155	47.7349	-122.295	1550	8460
38	1959701745	20141107T000000	1680000	6	2.25	4910	6600	2.5	0	0	5	10	3580	1330	1910	0	98102	47.6458	-122.32	3280	5500
39	2412600070	20141030T000000	230000	6	3	2180	7220	2	0	0	3	7	2180	0	1980	0	98003	47.3046	-122.305	2260	7344
40	1066000290	20141117T000000	600000	6	3	2600	9350	1	0	0	4	8	1340	1260	1963	0	98008	47.6198	-122.105	2090	9102
41	5132000140	20140618T000000	175000	6	1	1370	5080	1.5	0	0	3	6	1120	250	1931	0	98106	47.5238	-122.35	1020	5080
42	2397101606	20141208T000000	2630000	6	4.75	5540	7200	2.5	0	2	4	11	3950	1590	1909	0	98119	47.6361	-122.366	2930	7200
43	6071800100	20150327T000000	815000	6	3	2860	17853	1	0	0	3	8	1430	1430	1962	2015	98006	47.546	-122.175	1920	13452

The boxplot above indicates that there do exist outliers within the *bedrooms* variable. That said, when comparing the number of bedrooms of these outliers with their other respective variables namely *sqft_lot*, *sqft_living*, *sqft_lot15*, *sqft_living15* and *bathrooms*, consistency among the number of bedrooms and these variables has been found to be maintained. Hence, because the higher number of bedrooms has been justified, implying that they are indeed genuine outliers, none of them would then be removed but kept within the dataset instead.

sqft_living

Source Code:

```
/* sqft_living VARIABLE */
* Compute quartiles for the 'sqft_living' variable;
PROC UNIVARIATE DATA=house;
    VAR sqft_living;
    OUTPUT OUT=OutliersSqftLiving (RENAME=(sqft_living=OriginalSqftLiving))
    Q1=Q1_sqft_living Q3=Q3_sqft_living;
RUN;

* Detect and store outliers for 'sqft_living' in a new dataset;
DATA OutliersListSqftLiving (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftLiving;
    IQR = Q3_sqft_living - Q1_sqft_living;
    LowerBound = Q1_sqft_living - 1.5 * IQR;
    UpperBound = Q3_sqft_living + 1.5 * IQR;
    IF sqft_living < LowerBound OR sqft_living > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_living;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_living Q3_sqft_living;
RUN;

* Print detected outliers for 'sqft_living';
PROC PRINT DATA=OutliersListSqftLiving;
RUN;

* Visualize 'sqft_living' distribution with a boxplot;
PROC SGPLOT DATA=house;
    VBOX sqft_living;
RUN;
```

```
/* Create a new dataset with the filtered observations */
data filtered_living;
    set house;
    if sqft_living in (5480, 8010, 5780, 5120, 5770, 7390, 5320, 5640, 5220, 5050,
                       6810, 5830, 6550, 5960, 7000, 6050, 7220, 5080, 5270, 6880,
                       5400, 5280, 5360, 7080, 5040, 5450, 5180, 5330, 5350, 5440,
                       6050, 5540, 7440);
run;

/* Print the filtered observations */
proc print data=filtered_living;
run;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_living

Moments			
N	3247	Sum Weights	3247
Mean	2061.44133	Sum Observations	6693500
Std Deviation	915.700697	Variance	838507.767
Skewness	1.33457476	Kurtosis	3.28126712
Uncorrected SS	1.65201E10	Corrected SS	2721796211
Coeff Variation	44.4204103	Std Error Mean	16.0698714

Basic Statistical Measures			
Location		Variability	
Mean	2061.441	Std Deviation	915.70070
Median	1880.000	Variance	838508
Mode	1640.000	Range	7620
		Interquartile Range	1140

Quantiles (Definition 5)	
Level	Quantile
100% Max	8010
99%	5040
95%	3740
90%	3240
75% Q3	2550
50% Median	1880
25% Q1	1410
10%	1060
5%	920
1%	720
0% Min	390

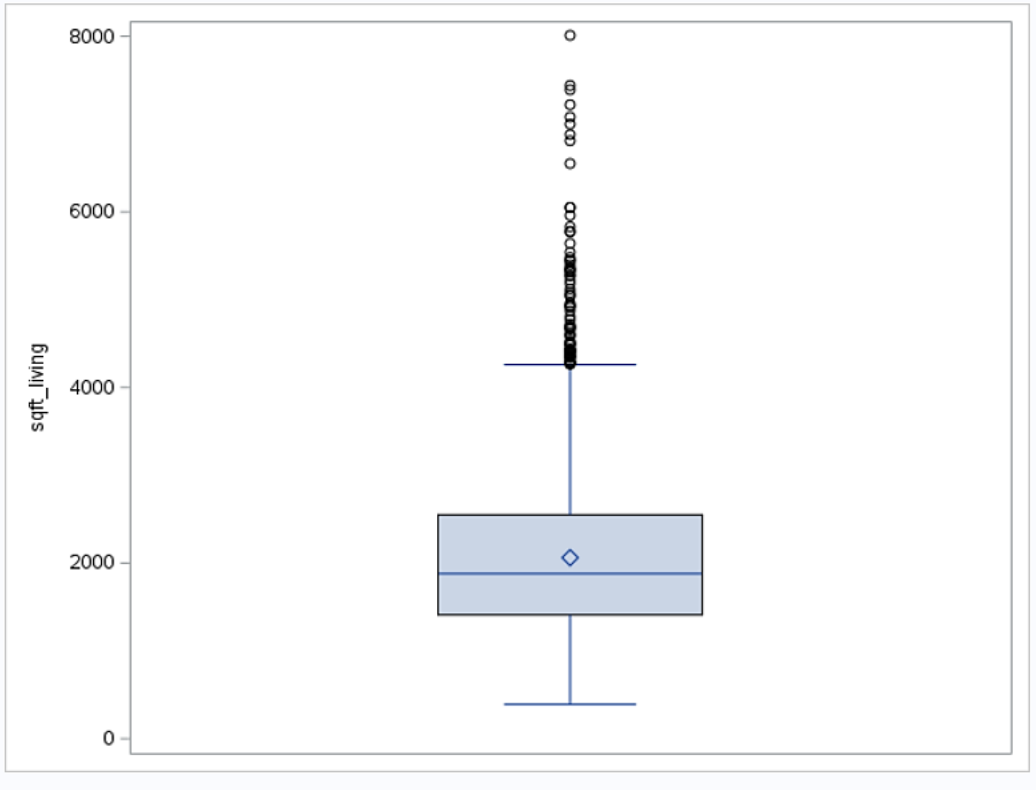
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
390	1239	7080	2274
520	2946	7220	1581
520	1753	7390	432
520	19	7440	3226
560	394	8010	235

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	3	0.09	100.00

Summarizing Properties:

- There are 3 missing values.
- On average, the houses within the dataset have an average interior living space of 2061.44 square feet.
- The standard deviation of approximately 915.7 square feet implies that there is a considerable dispersion between the interior living space of the houses within the dataset.
- The positive skewness indicates that the distribution is skewed to the right and that while most houses have typical interior living areas, a few however do have exceptionally large spaces.
- The slightly elevated kurtosis value suggests that even though many houses have interior living areas that are close to the average size, there are however properties with significantly different living areas, either those that are much larger or much smaller. In other words, there are potential outliers present.
- No inconsistencies were detected.

Outliers:



Obs	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
1	922059169	20141201T000000	800000	6	4.25	5480	189050	2	0	0	4	10	5140	340	1991	0	98031	47.412	-122.168	2470	10429
2	1247600105	20141020T000000	5110000	5	5.25	8010	45517	2	1	4	3	12	5990	2020	1999	0	98033	47.6767	-122.211	3430	26788
3	1118000301	20141219T000000	2890000	4	4	5780	7173	2	0	0	3	11	4130	1650	2008	0	98112	47.6374	-122.288	3930	7994
4	3754501240	20150206T000000	1550000	3	4	5120	4600	3	0	2	3	11	4490	630	2008	0	98034	47.7052	-122.223	2510	5918
5	8924600020	20141114T000000	1540000	4	4.5	5770	10050	1	0	3	5	9	3160	2610	1949	0	98115	47.677	-122.275	2950	6700
6	7558700030	20150413T000000	5300000	6	6	7390	24829	2	1	4	4	12	5000	2390	1991	0	98040	47.5631	-122.21	4320	24619
7	98000060	20140714T000000	1060000	4	4	5320	20041	2	0	0	3	11	5320	0	2003	0	98075	47.5852	-121.966	4640	17268
8	2424059174	20150508T000000	2000000	4	3.25	5640	35006	2	0	2	3	11	4900	740	2015	0	98006	47.5491	-122.104	4920	35033
9	8920100066	20140820T000000	1480000	4	3.5	5220	15411	2	0	3	3	11	3550	1670	2006	0	98075	47.592	-122.085	3110	14124
10	7960900060	20150504T000000	2900000	4	3.25	5050	20100	1.5	0	2	3	11	4750	300	1982	2008	98004	47.6312	-122.223	3890	20060
11	9831200500	20150304T000000	2480000	5	3.75	6810	7500	2.5	0	0	3	13	6110	700	1922	0	98102	47.6285	-122.322	2660	7500
12	8678500020	20141213T000000	1580000	4	3.5	5830	131116	2	0	0	3	11	5830	0	2005	0	98024	47.5986	-121.949	5340	207206
13	622069006	20140820T000000	1500000	4	5.5	6550	217374	1	0	0	3	11	5400	1150	2006	0	98058	47.4302	-122.095	4110	50378
14	4039800080	20140529T000000	1360000	5	3.5	5960	13703	2	0	2	3	10	4770	1190	1984	0	98008	47.6151	-122.107	2810	17320
15	624069108	20140812T000000	3200000	4	3.25	7000	28206	1	1	4	4	12	3500	3500	1991	0	98075	47.5928	-122.086	4913	14663
16	3225079035	20140618T000000	1600000	6	5	6050	230652	2	0	3	3	11	6050	0	2001	0	98024	47.6033	-121.943	4210	233971
17	2626069030	20150209T000000	1940000	4	5.75	7220	223462	2	0	4	3	12	6220	1000	2000	0	98053	47.7097	-122.013	2680	7593
18	3303850330	20141216T000000	1900000	4	3.25	5080	27755	2	0	0	3	11	5080	0	2001	0	98006	47.5423	-122.111	4730	22326
19	2923500230	20141216T000000	2600000	4	4.5	5270	12195	2	1	4	3	11	3400	1870	1979	0	98027	47.5696	-122.09	3390	9905
20	8835770170	20140822T000000	1490000	5	6	6880	279968	2	0	3	3	12	4070	2810	2007	0	98045	47.4624	-121.779	4690	256803
21	203100440	20140911T000000	1210000	3	3.75	5400	24740	2	0	0	3	11	5400	0	1997	0	98053	47.6426	-121.955	1690	20000
22	5451300117	20150422T000000	1550000	4	4	5280	17677	2	0	3	3	11	3220	2060	1978	0	98040	47.5323	-122.238	3470	17474
23	7964410100	20150504T000000	700000	4	3.5	5360	25800	1	0	0	3	9	3270	2090	1971	0	98074	47.6099	-122.054	2650	21781
24	6447300265	20141014T000000	4000000	4	5.5	7080	16573	2	0	0	3	12	5760	1320	2008	0	98039	47.6151	-122.224	3140	15996
25	7237500650	20150213T000000	1280000	5	4.25	5040	9466	2	0	0	3	11	5040	0	2004	0	98059	47.5282	-122.133	4300	9417
26	2525049148	20141007T000000	3420000	5	5	5450	20412	2	0	0	3	11	5450	0	2014	0	98039	47.6209	-122.237	3160	17825
27	7851980100	20140605T000000	1080000	5	4.75	5180	17811	2	0	2	3	11	4070	1110	2001	0	98065	47.5405	-121.868	3960	15103
28	8562710550	20140521T000000	950000	5	3.75	5330	6000	2	0	2	3	10	3570	1760	2006	0	98027	47.5401	-122.073	4420	5797
29	9808100150	20150402T000000	3350000	5	3.75	5350	15360	1	0	1	3	11	3040	2310	2008	0	98004	47.648	-122.218	3740	15940
30	2954400310	20140915T000000	1770000	4	3.5	5440	38900	2	0	0	3	12	5440	0	1990	0	98053	47.6605	-122.069	4830	41313
31	1526069135	20141211T000000	930000	4	4	6050	84942	2.5	0	2	3	9	4150	1900	2009	0	98077	47.7466	-122.029	2700	199504
32	2397101606	20141208T000000	2630000	6	4.75	5540	7200	2.5	0	2	4	11	3950	1590	1909	0	98119	47.6361	-122.366	2930	7200
33	6065300370	20150506T000000	4210000	5	6	7440	21540	2	0	0	3	12	5550	1890	2003	0	98006	47.5692	-122.189	4740	19329

The boxplot above suggests that there is a considerable number of outliers present within the *sqft_living* variable. Nevertheless, when these outliers are evaluated in accordance to other variables such as the number of bedrooms, the number of bathrooms, the number of floors in the house as well as the square footage of interior housing living space for the nearest 15 neighbours, the higher values of *sqft_living* of these outliers do however appear to be adequately reasonable and justifiable, which is unlike what was suggested by the boxplot. Hence, no outliers will be removed here.

sqft_lot

Source Code:

```
/* sqft_lot VARIABLE */
/* Compute quartiles for the 'sqft_lot' variable */
PROC UNIVARIATE DATA=house;
    VAR sqft_lot;
    OUTPUT OUT=OutliersSqftLot (RENAME=(sqft_lot=OriginalSqftLot))
    Q1=Q1_sqft_lot Q3=Q3_sqft_lot;
RUN;

/* Detect and store outliers for 'sqft_lot' in a new dataset */
DATA OutliersListSqftLot (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftLot;
    IQR = Q3_sqft_lot - Q1_sqft_lot;
    LowerBound = Q1_sqft_lot - 1.5 * IQR;
    UpperBound = Q3_sqft_lot + 1.5 * IQR;
    IF sqft_lot < LowerBound OR sqft_lot > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_lot;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_lot Q3_sqft_lot;
RUN;

/* Print detected outliers for 'sqft_lot' */
PROC PRINT DATA=OutliersListSqftLot;
RUN;

/* Visualize 'sqft_lot' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX sqft_lot;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_lot

Moments			
N	3248	Sum Weights	3248
Mean	15002.5034	Sum Observations	48728131
Std Deviation	39196.2315	Variance	1536344565
Skewness	9.88388029	Kurtosis	142.063131
Uncorrected SS	5.71955E12	Corrected SS	4.98851E12
Coeff Variation	261.264607	Std Error Mean	687.759034

Basic Statistical Measures			
Location		Variability	
Mean	15002.50	Std Deviation	39196
Median	7560.00	Variance	1536344565
Mode	5000.00	Range	919733
		Interquartile Range	5500

Quantiles (Definition 5)	
Level	Quantile
100% Max	920423
99%	212137
95%	45613
90%	20954
75% Q3	10500
50% Median	7560
25% Q1	5000
10%	3150
5%	1598
1%	1007
0% Min	690

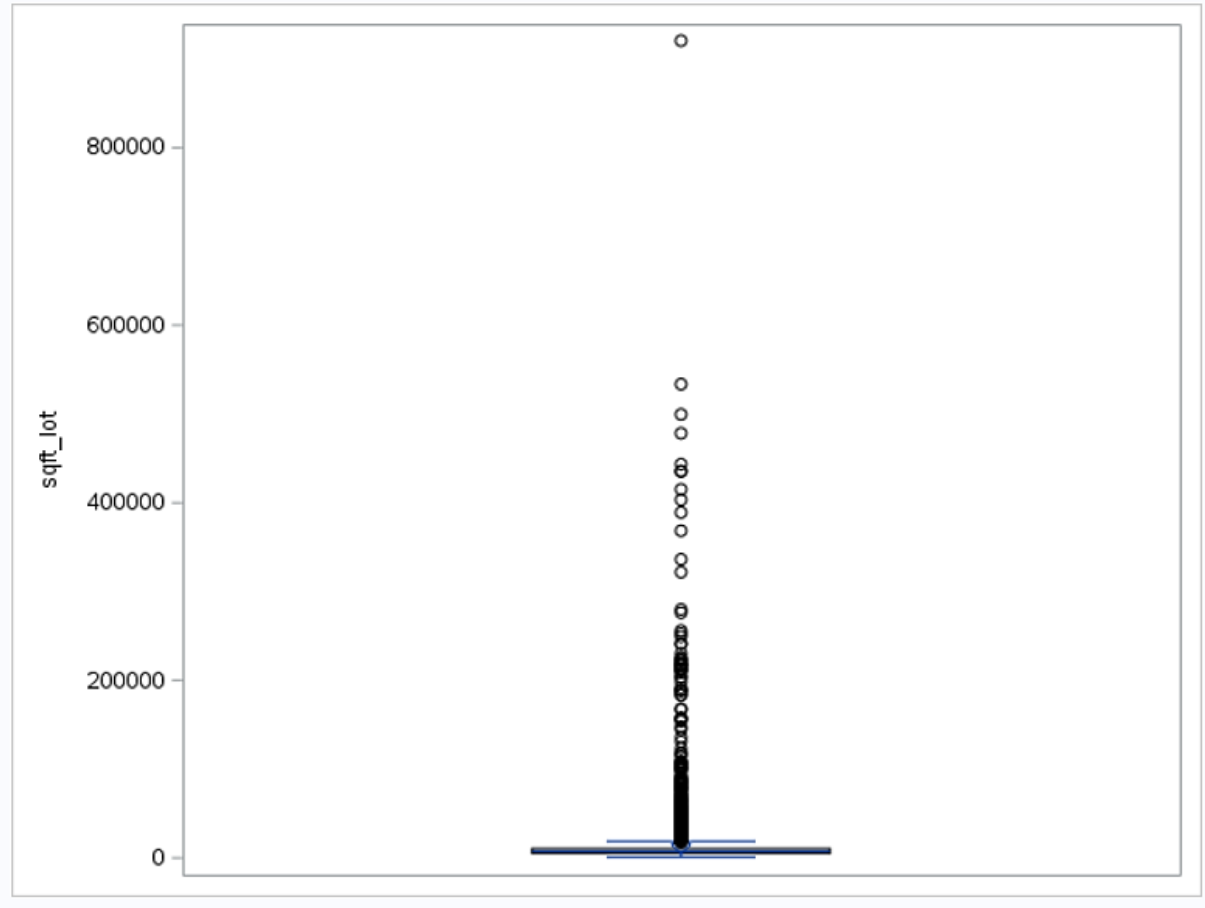
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
690	94	443440	1957
704	2406	478288	713
713	606	499571	2571
779	2210	533610	2627
780	3076	920423	1212

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	2	0.06	100.00

Summarizing Properties:

- There are 2 missing values.
- On average, the houses within the dataset have an average land space of 15002.50 square feet.
- The standard deviation of approximately 39,196 square feet is notably large, more so when it is compared to the mean lot size of about 15,002.5 square feet. This high standard deviation suggests that there is a diverse range of property lot sizes within the dataset, ranging from small urban plots to potentially expansive rural or suburban properties.
- A skewness of 9.88 suggests that the data distribution is positively skewed, meaning most houses have lot sizes that are average sized, but at the same time, there are a few properties with exceptionally huge lot sizes as depicted by the long right tail.
- The large positive kurtosis informs of the potentiality of outliers within the dataset.
- No inconsistencies were detected.

Outlier



Obs	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
1	3520069033	20140623T000000	230000	3	1	1530	389126	1.5	0	0	4	7	1530	0	1919	0	98022	47.1776	-122.011	1768	42148
2	720079001	20140626T000000	667000	3	1.75	3320	478288	1.5	0	3	4	8	2260	1060	1933	1982	98022	47.2407	-121.953	2960	217800
3	2523089025	20150210T000000	1080000	3	3	4020	435600	1.5	0	2	3	10	4020	0	1999	0	98045	47.4418	-121.731	2590	283140
4	2624089007	20150320T000000	2000000	2	2.5	3900	920423	2	0	0	3	12	3900	0	2009	0	98065	47.5371	-121.756	2720	411962
5	826069002	20141029T000000	355000	2	1	1350	368517	1	0	0	3	6	1350	0	1947	0	98077	47.7617	-122.061	2330	104108
6	1623069046	20150312T000000	1700000	4	3.5	4070	336283	2	0	0	3	11	4070	0	2006	0	98027	47.478	-122.038	3020	44613
7	621069057	20150323T000000	569950	4	3.5	2700	443440	1.5	0	0	3	8	2700	0	1948	1997	98042	47.333	-122.098	3210	298182
8	3420069055	20141203T000000	350000	4	2.25	1570	499571	1	0	3	4	7	1570	0	1972	0	98022	47.1808	-122.023	1700	181708
9	1222069089	20140904T000000	375000	1	1	800	533610	1.5	0	0	5	5	800	0	1950	0	98038	47.4134	-121.986	1790	216057
10	3624079067	20140508T000000	330000	2	2	1550	435600	1.5	0	0	2	7	1550	0	1972	0	98065	47.5145	-121.853	1600	217800
11	1422069070	20150507T000000	472000	3	2.5	1860	415126	2	0	0	3	7	1860	0	2006	0	98038	47.3974	-122.005	2070	54014
12	1720069006	20140812T000000	474000	2	1	1050	403365	1	0	3	5	6	1050	0	1905	0	98022	47.2221	-122.059	1760	108900
13	1622069127	20141118T000000	525000	5	3.25	3960	321908	2	0	0	4	9	2690	1270	1989	0	98038	47.3984	-122.055	2360	96703

From the boxplot given above and consistent with the implication of the high positive kurtosis, there are indeed outliers present. Upon further investigation into the observations with extremely high lot sizes, observation number 9 with house id 1222069089 is the only one that does not seem to not make sense. Reason being there is only one bedroom and one bathroom in the house and that when comparing the square footage of the interior living space of the said house with that of its nearest 15 neighbours, the former was only 800 square feet while the latter was 1790 square feet, more than double of the size of the former. Intuitively, a land lot of 533610 square feet for the given house features discussed above is just illogical. Hence, observation number 9 is identified as the only outlier that needs to be removed.

grade

Source Code:

```
/* grade VARIABLE */
/* Compute quartiles for the 'grade' variable */
PROC UNIVARIATE DATA=house;
     VAR grade;
     OUTPUT OUT=OutliersGrade (RENAME=(grade=OriginalGrade))
     Q1=Q1_grade Q3=Q3_grade;
RUN;

/* Detect and store outliers for 'grade' in a new dataset */
DATA OutliersListGrade (keep=ObsNum OutlierValue);
     SET house;
     IF _N_ = 1 THEN SET OutliersGrade;
     IQR = Q3_grade - Q1_grade;
     LowerBound = Q1_grade - 1.5 * IQR;
     UpperBound = Q3_grade + 1.5 * IQR;
     IF grade < LowerBound OR grade > UpperBound THEN DO;
          ObsNum = _N_;
          OutlierValue = grade;
          OUTPUT;
     END;
     DROP IQR LowerBound UpperBound Q1_grade Q3_grade;
RUN;

/* Print detected outliers for 'grade' */
PROC PRINT DATA=OutliersListGrade;
RUN;
```

```
/* Visualize 'grade' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX grade;
RUN;


/* Create a new dataset with the filtered observations */
data filtered_data;
    set house;
    if sqft_lot in (389126, 478288, 435600, 920423, 368517, 336283, 443440, 499571, 533610, 415126, 403365, 321908);
run;

/* Print the filtered observations */
proc print data=filtered_data;
run;
```

Output:

The UNIVARIATE Procedure
Variable: grade

Moments			
N	3249	Sum Weights	3249
Mean	7.65835642	Sum Observations	24882
Std Deviation	1.18003397	Variance	1.39248018
Skewness	0.78089457	Kurtosis	1.07123553
Uncorrected SS	195078	Corrected SS	4522.77562
Coeff Variation	15.4084494	Std Error Mean	0.02070235

Basic Statistical Measures			
Location		Variability	
Mean	7.658356	Std Deviation	1.18003
Median	7.000000	Variance	1.39248
Mode	7.000000	Range	9.00000
		Interquartile Range	1.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	13
99%	11
95%	10
90%	9
75% Q3	8
50% Median	7
25% Q1	7
10%	6
5%	6
1%	5
0% Min	4

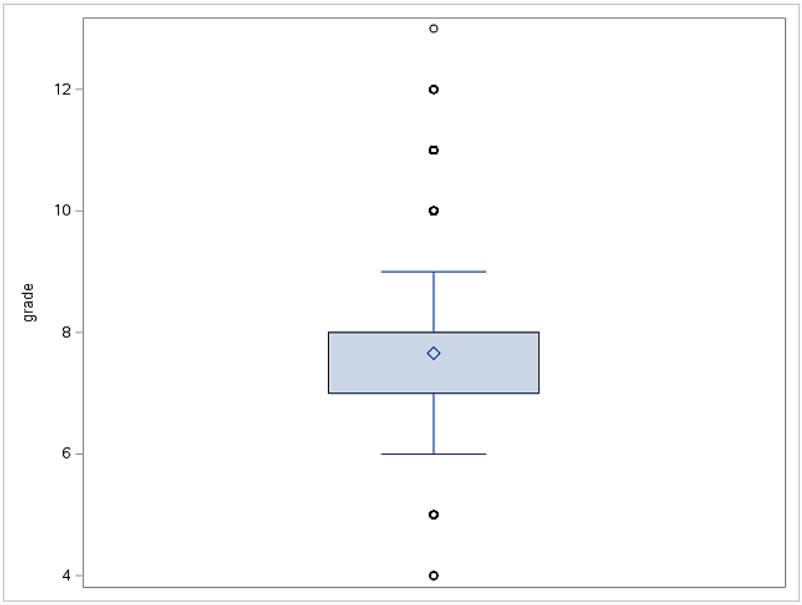
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
4	3145	12	2855
4	2804	12	3011
4	1239	12	3155
4	19	12	3226
5	3025	13	867

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	1	0.03	100.00

Summarizing Properties:

- There is 1 missing value.
- On average, the houses within the dataset have an average index grade of 7.66, indicating that they are of an average level in terms of construction and design.
- A standard deviation of 1.1800 suggests that the spread in terms of the grades of houses within the dataset is moderate
- The data distribution is slightly skewed to the right, implying that while most houses have an average grade, there are however a few with exceptionally high grades.
- The positive kurtosis suggests that extreme values are present within the dataset in terms of either very high or/and very low grades.
- No inconsistencies were detected.

Outliers:



The boxplot above suggests that there are indeed both large and small outliers among the grading index of the houses. Having said that, because these outliers identified do still fall within the range of what is acceptable, that is between 1 to 13 as pointed out in *Appendix 2*, they are then considered genuine and would be kept within the dataset.

condition

Source Code:

```
/* condition VARIABLE */
* Compute quartiles for the 'condition' variable;
PROC UNIVARIATE DATA=house;
    VAR condition;
    OUTPUT OUT=OutliersCondition (RENAME=(condition=OriginalCondition))
    Q1=Q1_condition Q3=Q3_condition;
RUN;

* Detect and store outliers for 'condition' in a new dataset;
DATA OutliersListCondition (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersCondition;
    IQR = Q3_condition - Q1_condition;
    LowerBound = Q1_condition - 1.5 * IQR;
    UpperBound = Q3_condition + 1.5 * IQR;
    IF condition < LowerBound OR condition > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = condition;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_condition Q3_condition;
RUN;

* Print detected outliers for 'condition';
PROC PRINT DATA=OutliersListCondition;
RUN;

* Visualize 'condition' distribution with a boxplot;
PROC SGPLOT DATA=house;
    VBOX condition;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: condition

Moments			
N	3249	Sum Weights	3249
Mean	3.41520468	Sum Observations	11096
Std Deviation	0.64872196	Variance	0.42084018
Skewness	1.01087615	Kurtosis	0.50269757
Uncorrected SS	39262	Corrected SS	1366.88889
Coeff Variation	18.995112	Std Error Mean	0.01138109

Basic Statistical Measures			
Location		Variability	
Mean	3.415205	Std Deviation	0.64872
Median	3.000000	Variance	0.42084
Mode	3.000000	Range	4.00000
		Interquartile Range	1.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	5
99%	5
95%	5
90%	4
75% Q3	4
50% Median	3
25% Q1	3
10%	3
5%	3
1%	3
0% Min	1

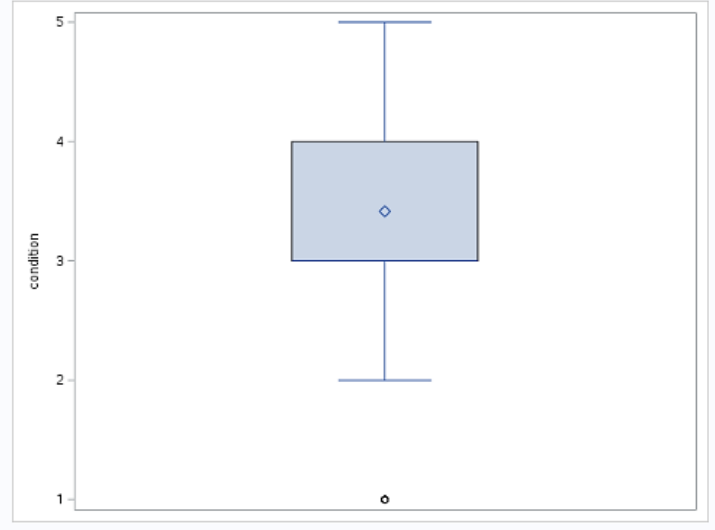
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
1	2159	5	3217
1	2016	5	3220
1	1883	5	3228
1	1687	5	3247
1	1058	5	3250

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	1	0.03	100.00

Summarizing Properties:

- There is 1 missing value.
- On average, the condition of houses within the dataset has an index of 3.42 which indicates that the condition is of an average one.
- There is a moderate spread among data points given a standard deviation of 0.6487.
- The distribution is positively-skewed, suggesting that most houses have a condition rating on the lower end.
- The kurtosis suggests that the distribution has a sharper peak and slightly heavier tails than a normal distribution, essentially implying the presence of many extreme values or outliers.
- No inconsistencies were detected.

Outliers:



There seems to be some outliers taking the value of 1 for the condition index. Nevertheless, because these outliers do still fall within the given range of the *condition* variable, they are thereby considered legitimate data points and will not be removed.

view

Source Code:

```
/* view VARIABLE */
/* Compute quartiles for the 'view' variable */
PROC UNIVARIATE DATA=house;
    VAR view;
    OUTPUT OUT=OutliersView (RENAME=(view=OriginalView))
    Q1=Q1_view Q3=Q3_view;
RUN;

/* Detect and store outliers for 'view' in a new dataset */
DATA OutliersListView (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersView;
    IQR = Q3_view - Q1_view;
    LowerBound = Q1_view - 1.5 * IQR;
    UpperBound = Q3_view + 1.5 * IQR;
    IF view < LowerBound OR view > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = view;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_view Q3_view;
RUN;
```

```
/* Print detected outliers for 'view' */
PROC PRINT DATA=OutliersListView;
RUN;

/* Visualize 'view' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX view;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: view

Moments			
N	3248	Sum Weights	3248
Mean	0.23337438	Sum Observations	758
Std Deviation	0.76879033	Variance	0.59103856
Skewness	3.44875012	Kurtosis	11.3275038
Uncorrected SS	2096	Corrected SS	1919.10222
Coeff Variation	329.423612	Std Error Mean	0.01348963

Basic Statistical Measures			
Location		Variability	
Mean	0.233374	Std Deviation	0.76879
Median	0.000000	Variance	0.59104
Mode	0.000000	Range	4.00000
		Interquartile Range	0

Quantiles (Definition 5)	
Level	Quantile
100% Max	4
99%	4
95%	2
90%	0
75% Q3	0
50% Median	0
25% Q1	0
10%	0
5%	0
1%	0
0% Min	0

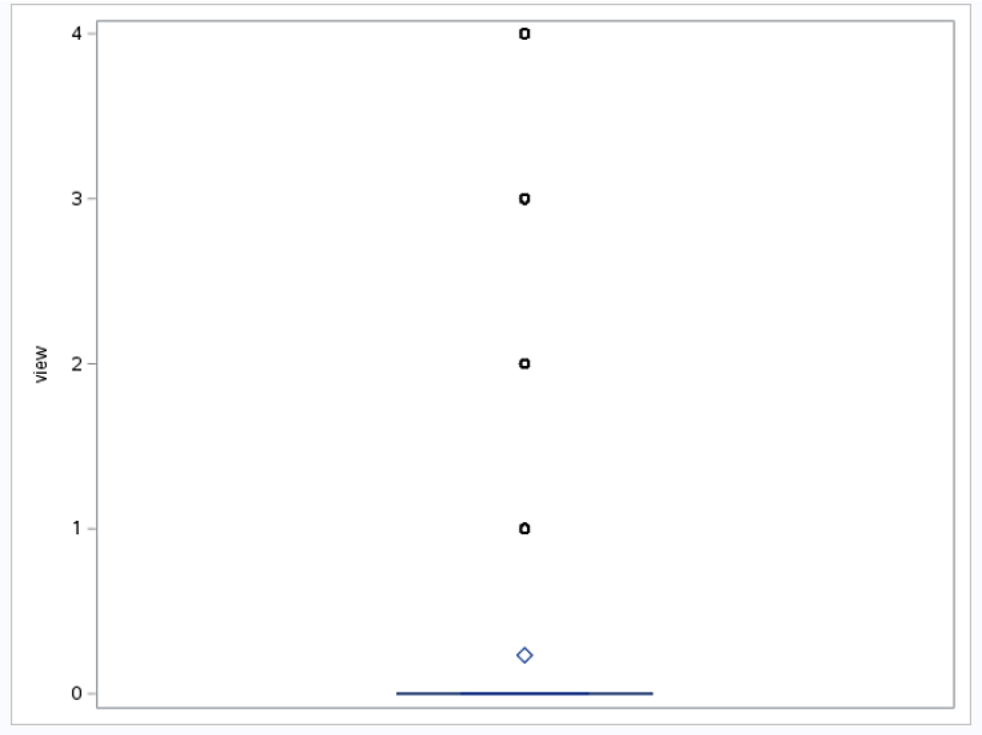
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0	3250	4	2965
0	3248	4	2974
0	3246	4	3029
0	3245	4	3053
0	3243	4	3079

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	2	0.06	100.00

Summarizing Properties:

- There are 2 missing values.
- On average, the view of houses within the dataset has an index of 0.2334 which indicates that the view limited.
- There is a moderate spread among data points given a standard deviation of 0.7688.
- The high skewness value suggests a pronounced right skewed distribution, essentially implying that there exists a significant imbalance between the number of houses with average or below-average views and those with exceptional views.
- The high kurtosis value suggests that there are outliers within the view variable.
- No inconsistencies were detected.

Outliers:



Even though the boxplot above suggests that there are outliers present, yet because all these outliers have values that fall within the given range as specified in *Appendix 2*, they are thereby considered genuine and will not be removed.

floors

Source Code:

```
/* floors VARIABLE */
/* Compute quartiles for the 'floors' variable */
PROC UNIVARIATE DATA=house;
    VAR floors;
    OUTPUT OUT=OutliersFloors (RENAME=(floors=OriginalFloors))
    Q1=Q1_floors Q3=Q3_floors;
RUN;

/* Detect and store outliers for 'floors' in a new dataset */
DATA OutliersListFloors (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersFloors;
    IQR = Q3_floors - Q1_floors;
    LowerBound = Q1_floors - 1.5 * IQR;
    UpperBound = Q3_floors + 1.5 * IQR;
    IF floors < LowerBound OR floors > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = floors;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_floors Q3_floors;
RUN;

/* Print detected outliers for 'floors' */
PROC PRINT DATA=OutliersListFloors;
RUN;

/* Visualize 'floors' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX floors;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: floors

Moments			
N	3246	Sum Weights	3246
Mean	1.49199014	Sum Observations	4843
Std Deviation	0.5432856	Variance	0.29515924
Skewness	0.66200882	Kurtosis	-0.3912708
Uncorrected SS	8183.5	Corrected SS	957.791744
Coeff Variation	36.4134846	Std Error Mean	0.00953573

Basic Statistical Measures			
Location		Variability	
Mean	1.491990	Std Deviation	0.54329
Median	1.500000	Variance	0.29516
Mode	1.000000	Range	2.50000
		Interquartile Range	1.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	3.5
99%	3.0
95%	2.0
90%	2.0
75% Q3	2.0
50% Median	1.5
25% Q1	1.0
10%	1.0
5%	1.0
1%	1.0
0% Min	1.0

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
1	3250	3.0	3123
1	3249	3.0	3142
1	3246	3.0	3205
1	3240	3.0	3233
1	3239	3.5	367

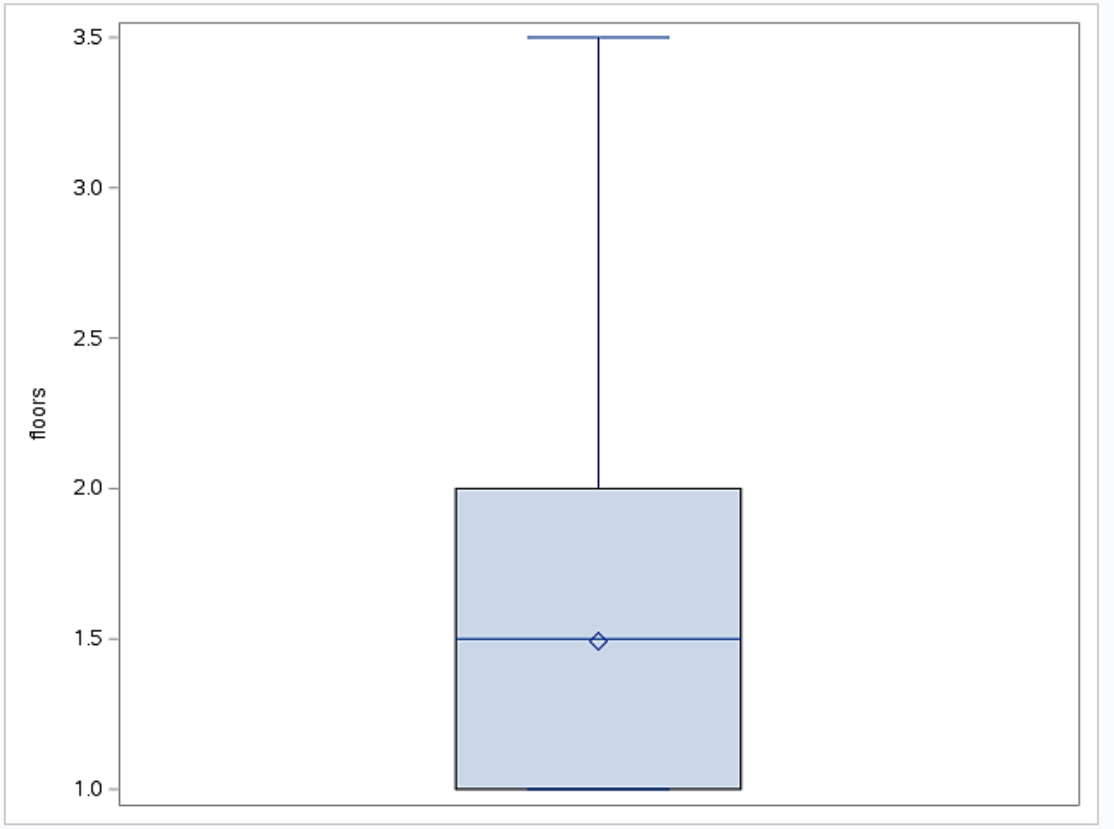
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	4	0.12	100.00

Summarizing Properties:

- There are 4 missing values.
- The mean value suggests that most houses within the dataset are single-storied.

- There is a moderate degree of variability in the number of floors among the houses within the dataset, given a standard deviation of 0.7688.
- The positive skewness suggests that even though majority of the houses are single-storied, there are however few houses with a significantly higher number of floors (like three-story or more), which have caused the right tail to be elongated.
- The negative kurtosis value indicates that the dataset has fewer extreme values in terms of the number of floors than a normal distribution would. In other words, most houses to conform to the typical designs.
- No inconsistencies were detected.

Outliers:



No outliers have been detected according to the boxplot above.

waterfront

Source Code:

```
/* waterfront VARIABLE */
/* Compute quartiles for the 'waterfront' variable */
PROC UNIVARIATE DATA=house;
    VAR waterfront;
    OUTPUT OUT=OutliersWaterfront (RENAME=(waterfront=OriginalWaterfront))
    Q1=Q1_waterfront Q3=Q3_waterfront;
RUN;

/* Detect and store outliers for 'waterfront' in a new dataset */
DATA OutliersListWaterfront (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersWaterfront;
    IQR = Q3_waterfront - Q1_waterfront;
    LowerBound = Q1_waterfront - 1.5 * IQR;
    UpperBound = Q3_waterfront + 1.5 * IQR;
    IF waterfront < LowerBound OR waterfront > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = waterfront;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_waterfront Q3_waterfront;
RUN;

/* Print detected outliers for 'waterfront' */
PROC PRINT DATA=OutliersListWaterfront;
RUN;

/* Visualize 'waterfront' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX waterfront;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: waterfront

Moments			
N	3247	Sum Weights	3247
Mean	0.00954727	Sum Observations	31
Std Deviation	0.09725758	Variance	0.00945904
Skewness	10.0918612	Kurtosis	99.9072012
Uncorrected SS	31	Corrected SS	30.7040345
Coeff Variation	1018.69473	Std Error Mean	0.0017068

Basic Statistical Measures			
Location		Variability	
Mean	0.009547	Std Deviation	0.09726
Median	0.000000	Variance	0.00946
Mode	0.000000	Range	1.00000
		Interquartile Range	0

Quantiles (Definition 5)	
Level	Quantile
100% Max	1
99%	0
95%	0
90%	0
75% Q3	0
50% Median	0
25% Q1	0
10%	0
5%	0
1%	0
0% Min	0

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0	3250	1	2534
0	3249	1	2539
0	3248	1	2827
0	3247	1	2965
0	3246	1	3029

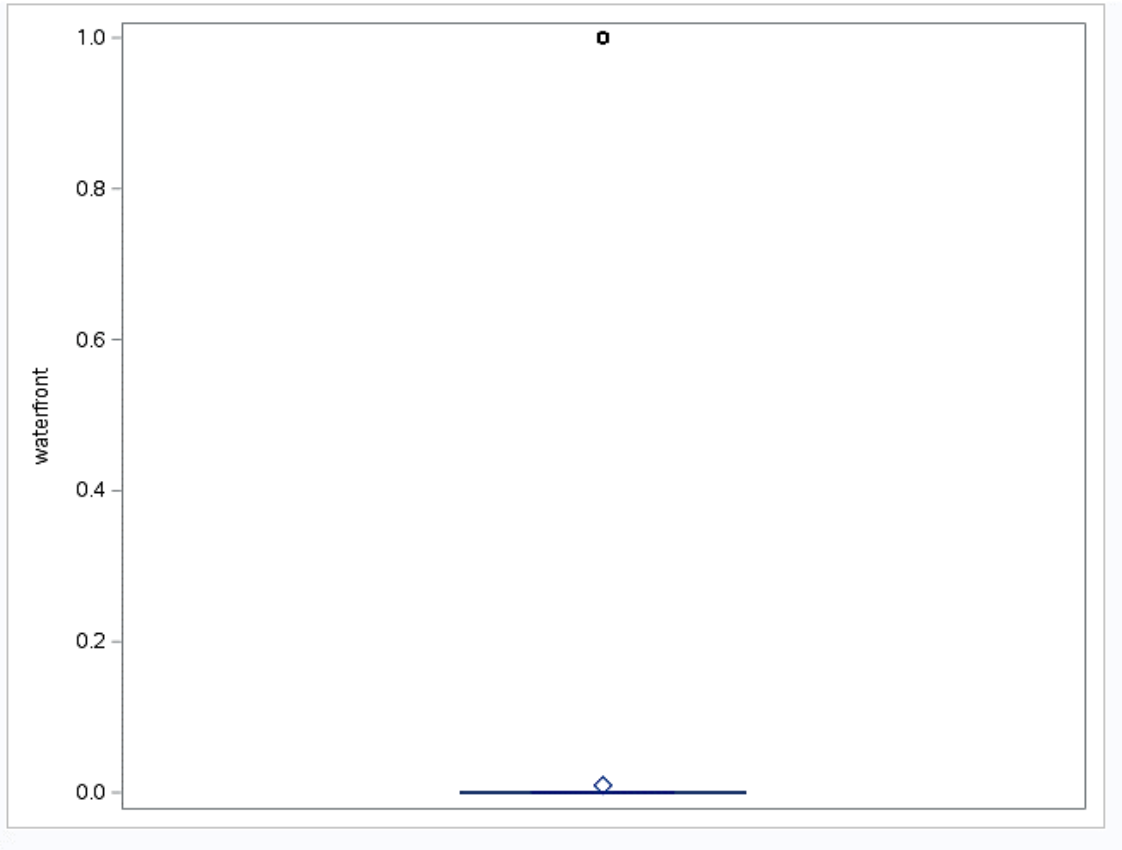
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	3	0.09	100.00

Summarizing Properties:

- There are 3 missing values.
- Given that the waterfront variable is a binary variable, where '1' indicates that the house is overlooking a waterfront while '0' represents otherwise, the low mean value suggests that only a very small fraction of houses within the dataset is actually overlooking a waterfront.

- The small standard deviation indicates that the majority of houses in the dataset do not have a waterfront,.
- The high positive skewness suggests a pronounced rightly skewed distribution. The distribution indicates that most houses in the dataset do not overlook a waterfront but at the same time, there are a few houses that do overlook a waterfront instead.
- The extremely high kurtosis value of 99.9072 indicates that the majority of the houses have a very common characteristic of not overlooking a waterfront, thereby resulting in a sharp peak in the distribution.
- No inconsistencies were detected.

Outliers:



No outliers were detected as all data points fall within the specified values of either ‘0’ or ‘1’ as indicated in *Appendix 2*.

sqft_above

Source Code:

```
/* sqft_above VARIABLE */
/* Compute quartiles for the 'sqft_above' variable */
PROC UNIVARIATE DATA=house;
    VAR sqft_above;
    OUTPUT OUT=OutliersSqftAbove (RENAME=(sqft_above=OriginalSqftAbove))
    Q1=Q1_sqft_above Q3=Q3_sqft_above;
RUN;

/* Detect and store outliers for 'sqft_above' in a new dataset */
DATA OutliersListSqftAbove (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftAbove;
    IQR = Q3_sqft_above - Q1_sqft_above;
    LowerBound = Q1_sqft_above - 1.5 * IQR;
    UpperBound = Q3_sqft_above + 1.5 * IQR;
    IF sqft_above < LowerBound OR sqft_above > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_above;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_above Q3_sqft_above;
RUN;

/* Print detected outliers for 'sqft_above' */
PROC PRINT DATA=OutliersListSqftAbove;
RUN;

/* Visualize 'sqft_above' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX sqft_above;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_above

Moments			
N	3247	Sum Weights	3247
Mean	1771.19218	Sum Observations	5751061
Std Deviation	818.210298	Variance	669468.092
Skewness	1.36572808	Kurtosis	2.41400228
Uncorrected SS	1.23593E10	Corrected SS	2173093426
Coeff Variation	46.1954557	Std Error Mean	14.3589868

Basic Statistical Measures			
Location		Variability	
Mean	1771.192	Std Deviation	818.21030
Median	1540.000	Variance	669468
Mode	1300.000	Range	5830
		Interquartile Range	1010

Quantiles (Definition 5)	
Level	Quantile
100% Max	6220
99%	4350
95%	3361
90%	2920
75% Q3	2190
50% Median	1540
25% Q1	1180
10%	950
5%	840
1%	700
0% Min	390

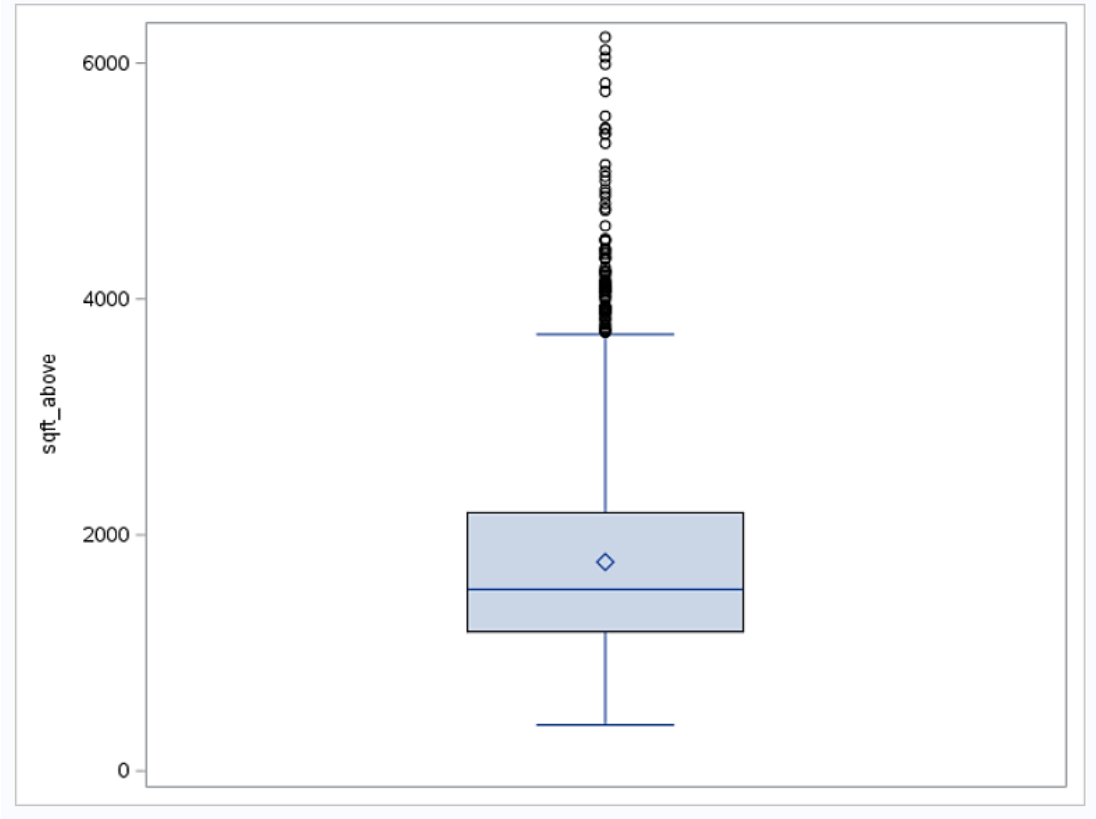
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
390	1239	5830	917
480	3003	5990	235
490	1377	6050	1526
520	2946	6110	867
520	1753	6220	1581

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	3	0.09	100.00

Summarizing Properties:

- There are 3 missing values.
- On average, the square footage of the interior housing space that is above ground level of the houses within the dataset is 1771.192 square feet.
- Given that the mean is 1,771.19 square feet, a standard deviation of 818.21 thereby suggests a considerable variation in the sizes of houses in the dataset.
- The positive skewness suggests that the distribution is skewed to the right.
- A kurtosis value of 2.4140 implies that there exist a diverse set of properties in the dataset, from average-sized houses to those with either exceptionally large or small above-ground living spaces.
- No inconsistencies were detected.

Outliers:



The boxplot above indicates that there are outliers within the *sqft_above* variable. Despite that and upon inspection, outliers within the boxplot for the *sqft_above* variable does in fact correspond to that done for the *sqft_living* variable. Hence, the outliers outlined above are considered genuine and will not be removed.

sqft_basement

Source Code:

```
/* sqft_basement VARIABLE */
/* Compute quartiles for the 'sqft_basement' variable */
PROC UNIVARIATE DATA=house;
    VAR sqft_basement;
    OUTPUT OUT=OutliersSqftBasement (RENAME=(sqft_basement=OriginalSqftBasement))
    Q1=Q1_sqft_basement Q3=Q3_sqft_basement;
RUN;

/* Detect and store outliers for 'sqft_basement' in a new dataset */
DATA OutliersListSqftBasement (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftBasement;
    IQR = Q3_sqft_basement - Q1_sqft_basement;
    LowerBound = Q1_sqft_basement - 1.5 * IQR;
    UpperBound = Q3_sqft_basement + 1.5 * IQR;
    IF sqft_basement < LowerBound OR sqft_basement > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_basement;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_basement Q3_sqft_basement;
RUN;

/* Print detected outliers for 'sqft_basement' */
PROC PRINT DATA=OutliersListSqftBasement;
RUN;

/* Visualize 'sqft_basement' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX sqft_basement;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_basement

Moments			
N	3246	Sum Weights	3246
Mean	289.16297	Sum Observations	938623
Std Deviation	443.11169	Variance	196347.969
Skewness	1.59888598	Kurtosis	2.57164866
Uncorrected SS	908564175	Corrected SS	637149161
Coeff Variation	153.239431	Std Error Mean	7.77748083

Basic Statistical Measures			
Location		Variability	
Mean	289.1630	Std Deviation	443.11169
Median	0.0000	Variance	196348
Mode	0.0000	Range	3500
		Interquartile Range	550.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	3500
99%	1670
95%	1190
90%	970
75% Q3	550
50% Median	0
25% Q1	0
10%	0
5%	0
1%	0
0% Min	0

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0	3248	2310	2822
0	3245	2390	432
0	3243	2610	406
0	3242	2810	2088
0	3241	3500	1450

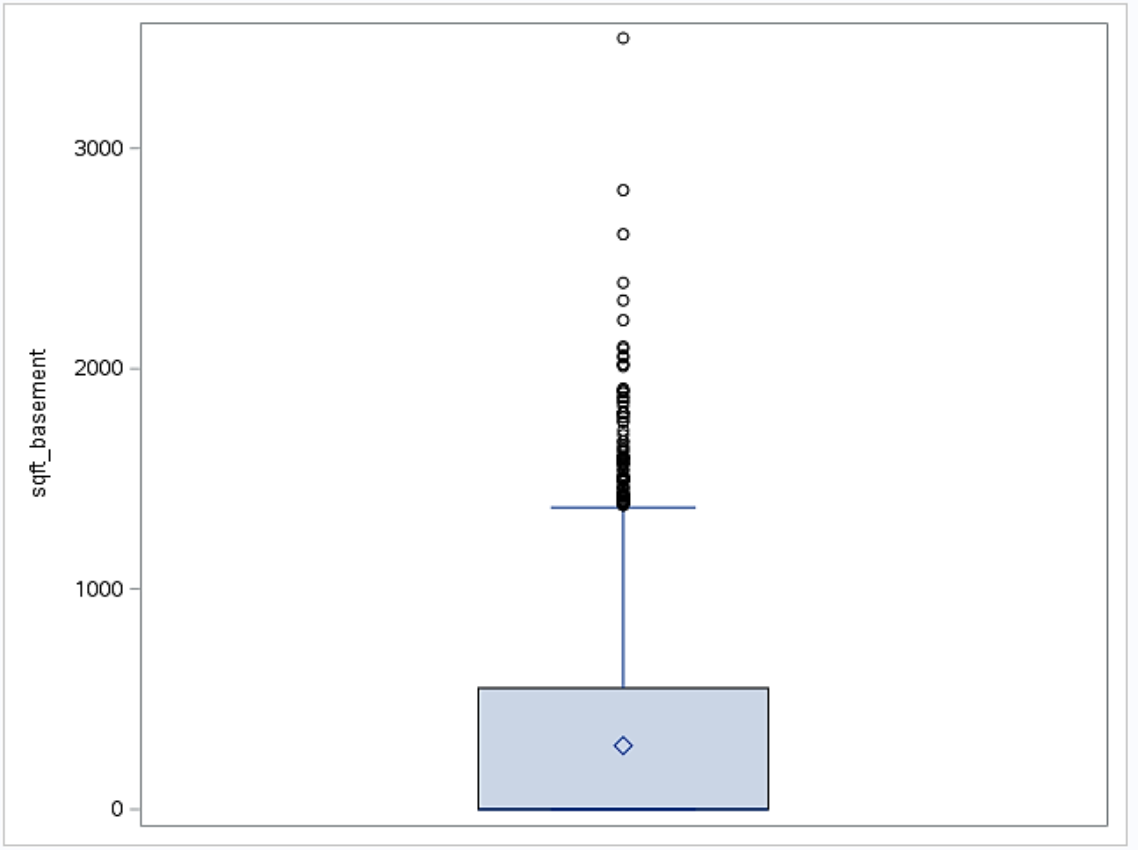
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	4	0.12	100.00

Summarizing Properties:

- There are 4 missing values.
- On average, the square footage of the interior housing space that is below ground level of the houses within the dataset is 289.16 square feet.

- Given that the mean is 289.16 square feet, a standard deviation of 443.11 thereby suggests a considerable variation in the sizes of houses in the dataset.
- The positive skewness of 1.5989 suggests that the distribution is highly skewed to the right, implying that while majority of the houses have smaller or no basements at all, there are however houses with exceptionally large basements still.
- The kurtosis value implies that there is potentially the presence of extreme values within the dataset in terms of the *sqft_basement* variable.
- No inconsistencies were detected.

Outliers:



The boxplot above indicates that there are outliers within the *sqft_basement* variable. Similar to the explanation provided for the *sqft_above* variable, even though there are outliers as indicated by the boxplot, these outliers do in fact correspond to that done for the *sqft_living* variable, once again making them genuine extreme values which must not be removed.

yr_built

Source Code:

```
/* yr_built VARIABLE */
/* Compute quartiles for the 'yr_built' variable */
PROC UNIVARIATE DATA=house;
    VAR yr_built;
    OUTPUT OUT=OutliersYrBuilt (RENAME=(yr_built=OriginalYrBuilt))
    Q1=Q1_yr_built Q3=Q3_yr_built;
RUN;

/* Detect and store outliers for 'yr_built' in a new dataset */
DATA OutliersListYrBuilt (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersYrBuilt;
    IQR = Q3_yr_built - Q1_yr_built;
    LowerBound = Q1_yr_built - 1.5 * IQR;
    UpperBound = Q3_yr_built + 1.5 * IQR;
    IF yr_built < LowerBound OR yr_built > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = yr_built;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_yr_built Q3_yr_built;
RUN;

/* Print detected outliers for 'yr_built' */
PROC PRINT DATA=OutliersListYrBuilt;
RUN;

/* Visualize 'yr_built' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX yr_built;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: yr_built

Moments			
N	3246	Sum Weights	3246
Mean	1970.69409	Sum Observations	6396873
Std Deviation	29.5840149	Variance	875.213937
Skewness	-0.4594976	Kurtosis	-0.7145666
Uncorrected SS	1.26091E10	Corrected SS	2840069.23
Coeff Variation	1.50119773	Std Error Mean	0.51925759

Basic Statistical Measures			
Location		Variability	
Mean	1970.694	Std Deviation	29.58401
Median	1975.000	Variance	875.21394
Mode	2014.000	Range	115.00000
		Interquartile Range	46.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	2015
99%	2014
95%	2011
90%	2007
75% Q3	1996
50% Median	1975
25% Q1	1950
10%	1925
5%	1915
1%	1905
0% Min	1900

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
1900	3161	2015	1391
1900	2275	2015	1842
1900	1536	2015	2249
1900	1057	2015	2834
1900	1005	2015	2948

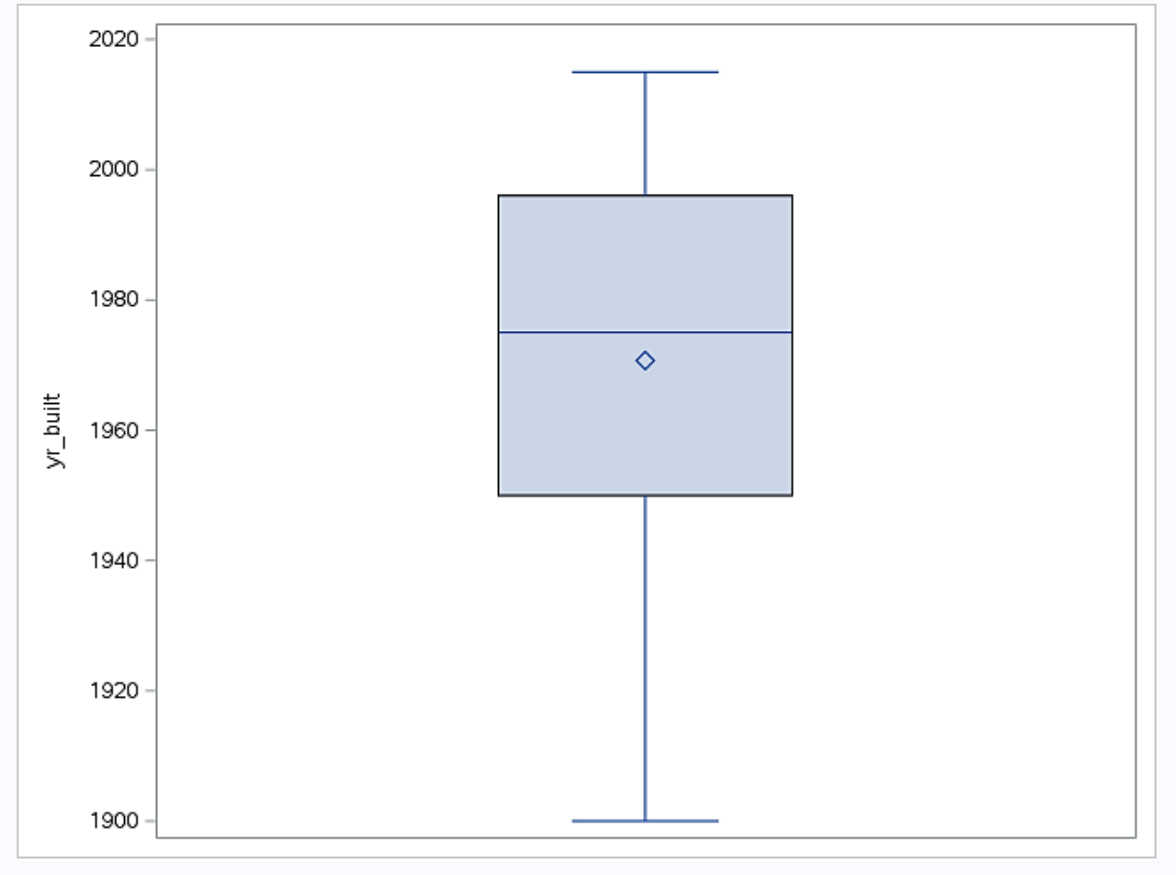
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	4	0.12	100.00

Summarizing Properties:

- There are 4 missing values.
- The mean value suggests that the majority of houses within the dataset were built around the 1970s.

- The standard deviation of roughly 29.58 years indicates that the years in which houses were built vary, on average, by about 30 years from the mean year of 1970.69. Meaning, houses within the dataset are not just built in the 1970s but also in the decades prior and post that period.
- The distribution is skewed to the left as indicated by the negative skewness value. This left skew suggests that the distribution has a longer tail on the left side which meant that there are fewer older houses, but instead a larger number of houses were built more recently.
- The negative kurtosis value implies that the years when houses were constructed are relatively uniformly spread out. That is, there are no extreme concentrations of houses built in a specific year or period, and there are not many extreme outliers in terms of very old or very new houses.
- No inconsistencies were detected.

Outliers:



No outliers have been detected according to the boxplot above.

lat

Source Code:

```
452 /* lat VARIABLE */
453 /* Compute quartiles for the 'lat' variable */
454 PROC UNIVARIATE DATA=house;
455     VAR lat;
456     OUTPUT OUT=OutliersLat (RENAME=(lat=OriginalLat))
457     Q1=Q1_lat Q3=Q3_lat;
458 RUN;
459
460 /* Detect and store outliers for 'lat' in a new dataset */
461 DATA OutliersListLat (keep=ObsNum OutlierValue);
462     SET house;
463     IF _N_ = 1 THEN SET OutliersLat;
464     IQR = Q3_lat - Q1_lat;
465     LowerBound = Q1_lat - 1.5 * IQR;
466     UpperBound = Q3_lat + 1.5 * IQR;
467     IF lat < LowerBound OR lat > UpperBound THEN DO;
468         ObsNum = _N_;
469         OutlierValue = lat;
470         OUTPUT;
471     END;
472     DROP IQR LowerBound UpperBound Q1_lat Q3_lat;
473 RUN;

475 /* Print detected outliers for 'lat' */
476 PROC PRINT DATA=OutliersListLat;
477 RUN;
478
479 /* Visualize 'lat' distribution with a boxplot */
480 PROC SGPLOT DATA=house;
481     VBOX lat;
482 RUN;
```

Output:

The UNIVARIATE Procedure
Variable: lat

Moments			
N	3247	Sum Weights	3247
Mean	47.5636364	Sum Observations	154439.128
Std Deviation	0.13837421	Variance	0.01914742
Skewness	-0.5199184	Kurtosis	-0.6735391
Uncorrected SS	7345748.66	Corrected SS	62.152531
Coeff Variation	0.29092437	Std Error Mean	0.00242837

Basic Statistical Measures			
Location		Variability	
Mean	47.56364	Std Deviation	0.13837
Median	47.57760	Variance	0.01915
Mode	47.68450	Range	0.59990
		Interquartile Range	0.21150

Quantiles (Definition 5)	
Level	Quantile
100% Max	47.7775
99%	47.7718
95%	47.7493
90%	47.7266
75% Q3	47.6831
50% Median	47.5776
25% Q1	47.4716
10%	47.3538
5%	47.3122
1%	47.2584
0% Min	47.1776

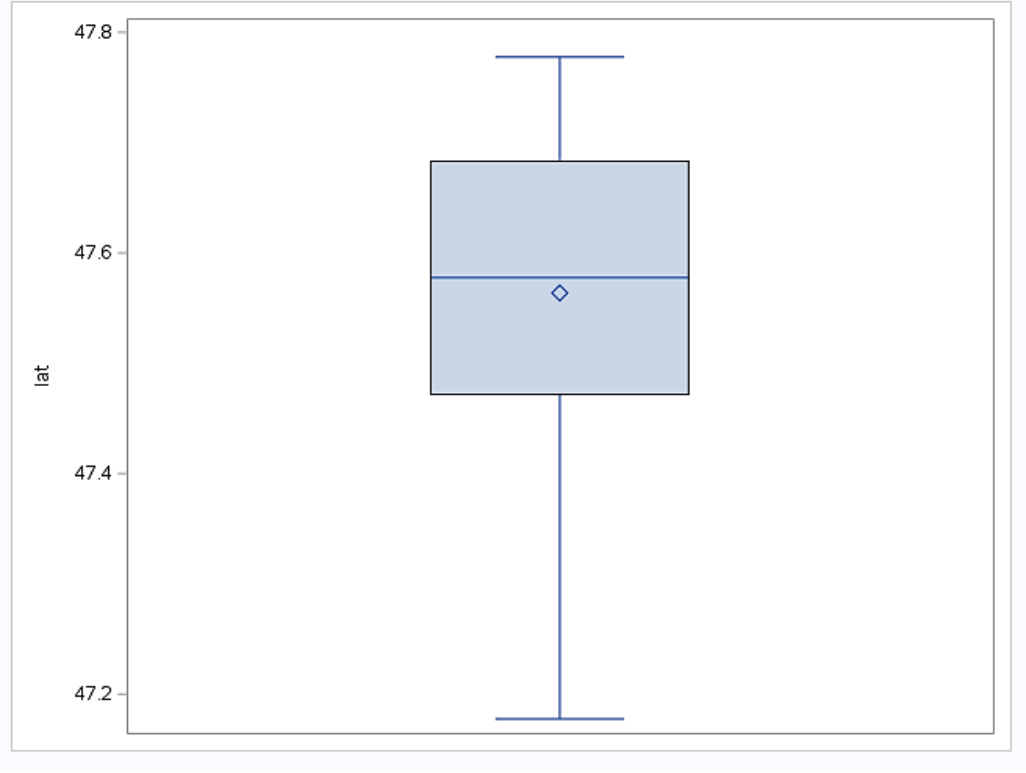
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
47.1776	147	47.7758	2016
47.1808	2571	47.7759	1617
47.1900	977	47.7762	3048
47.1913	2603	47.7770	1856
47.1927	2395	47.7775	150

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	3	0.09	100.00

Summarizing Properties:

- There are 3 missing values.
- The mean value suggests that most houses are geographically close to each other, centred around a latitude of approximately 47.56.
- The standard deviation proposes that most of the latitude values for the properties are closely clustered around the average latitude of 47.56. In other words, the properties are geographically concentrated in a specific region, with only slight variations in latitude.
- The negative skewness value of -0.5199 suggests that there are slightly more properties located to the south of the average latitude (47.56) than to the north.
- The negative kurtosis value implies that there are fewer extreme outliers in the dataset than would there be in a normal distribution. This is consistent with the idea that most properties are geographically close to each other, and that there are not many extreme latitudinal values.
- No inconsistencies were detected.

Outliers:



No outliers have been detected according to the boxplot above.

long

Source Code:

```
/* long VARIABLE */
/* Compute quartiles for the 'long' variable */
PROC UNIVARIATE DATA=house;
    VAR long;
    OUTPUT OUT=OutliersLong (RENAME=(long=OriginalLong))
    Q1=Q1_long Q3=Q3_long;
RUN;
```

```
/* Detect and store outliers for 'long' in a new dataset */
DATA OutliersListLong (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersLong;
    IQR = Q3_long - Q1_long;
    LowerBound = Q1_long - 1.5 * IQR;
    UpperBound = Q3_long + 1.5 * IQR;
    IF long < LowerBound OR long > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = long;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_long Q3_long;
RUN;

/* Print detected outliers for 'long' */
PROC PRINT DATA=OutliersListLong;
RUN;

/* Visualize 'long' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX long;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: long

Moments			
N	3245	Sum Weights	3245
Mean	-122.21463	Sum Observations	-396586.48
Std Deviation	0.1403142	Variance	0.01968807
Skewness	0.91859492	Kurtosis	1.38316073
Uncorrected SS	48468733.5	Corrected SS	63.868112
Coeff Variation	-0.1148097	Std Error Mean	0.00246317

Basic Statistical Measures			
Location		Variability	
Mean	-122.215	Std Deviation	0.14031
Median	-122.229	Variance	0.01969
Mode	-122.365	Range	1.19000
		Interquartile Range	0.20700

Quantiles (Definition 5)	
Level	Quantile
100% Max	-121.316
99%	-121.792
95%	-121.986
90%	-122.024
75% Q3	-122.123
50% Median	-122.229
25% Q1	-122.330
10%	-122.372
5%	-122.387
1%	-122.406
0% Min	-122.506

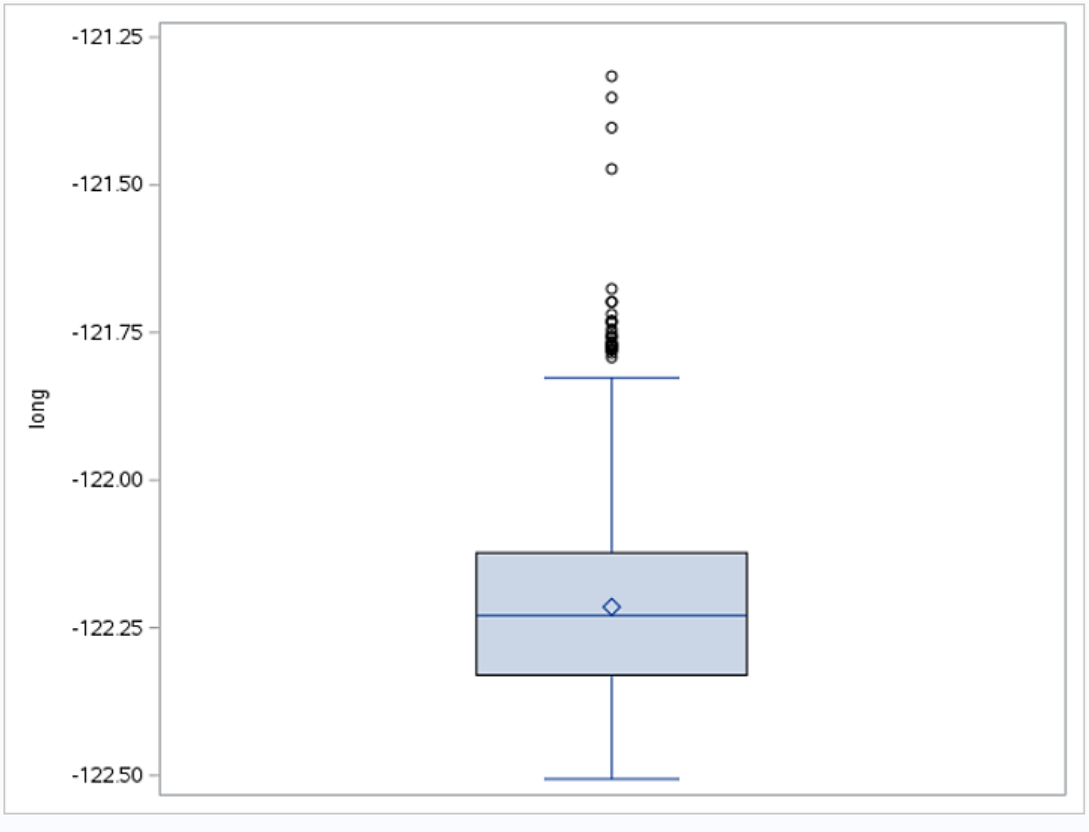
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
-122.506	132	-121.676	965
-122.504	1595	-121.473	939
-122.503	428	-121.403	1231
-122.486	2614	-121.352	623
-122.472	2673	-121.316	1809

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	5	0.15	100.00

Summarizing Properties:

- There are 5 missing values.
- The average longitude of the properties in the dataset is approximately -122.21, which is a longitude commonly associated with the western part of the United States, especially the Pacific Northwest.
- The standard deviation proposes that most of the longitude values of the properties are closely clustered around the mean longitude of -122.21. In other words, the properties are geographically concentrated in a specific longitudinal band, with only slight variations in longitude.
- The positive skewness value of 0.9186 suggests that the distribution is skewed to the right. This suggests that there are slightly more properties located to the east of the average longitude (-122.21) than to the west.
- The kurtosis value implies that there is a lower likelihood of extreme outliers in the longitude values.
- No inconsistencies were detected.

Outliers:



The boxplot shows that there are some outliers present in the *long* variable. Despite that, one would notice that the outliers outlined above do not deviate too far from the other non-outliers data points. That said, none of them would be removed.

sqft_living15

Source Code:

```
/* sqft_living15 VARIABLE */
/* Compute quartiles for the 'sqft_living15' variable */
PROC UNIVARIATE DATA=house;
    VAR sqft_living15;
    OUTPUT OUT=OutliersSqftLiving15 (RENAME=(sqft_living15=OriginalSqftLiving15))
    Q1=Q1_sqft_living15 Q3=Q3_sqft_living15;
RUN;

/* Detect and store outliers for 'sqft_living15' in a new dataset */
DATA OutliersListSqftLiving15 (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftLiving15;
    IQR = Q3_sqft_living15 - Q1_sqft_living15;
    LowerBound = Q1_sqft_living15 - 1.5 * IQR;
    UpperBound = Q3_sqft_living15 + 1.5 * IQR;
    IF sqft_living15 < LowerBound OR sqft_living15 > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_living15;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_living15 Q3_sqft_living15;
RUN;

/* Print detected outliers for 'sqft_living15' */
PROC PRINT DATA=OutliersListSqftLiving15;
RUN;

/* Visualize 'sqft_living15' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX sqft_living15;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_living15

Moments			
N	3245	Sum Weights	3245
Mean	1984.13005	Sum Observations	6438502
Std Deviation	687.931872	Variance	473250.261
Skewness	1.09188649	Kurtosis	1.52049061
Uncorrected SS	1.431E10	Corrected SS	1535223847
Coeff Variation	34.6717129	Std Error Mean	12.0764164

Basic Statistical Measures			
Location		Variability	
Mean	1984.130	Std Deviation	687.93187
Median	1840.000	Variance	473250
Mode	1440.000	Range	5391
		Interquartile Range	890.00000

Quantiles (Definition 5)	
Level	Quantile
100% Max	5790
99%	4080
95%	3280
90%	2920
75% Q3	2370
50% Median	1840
25% Q1	1480
10%	1250
5%	1140
1%	940
0% Min	399

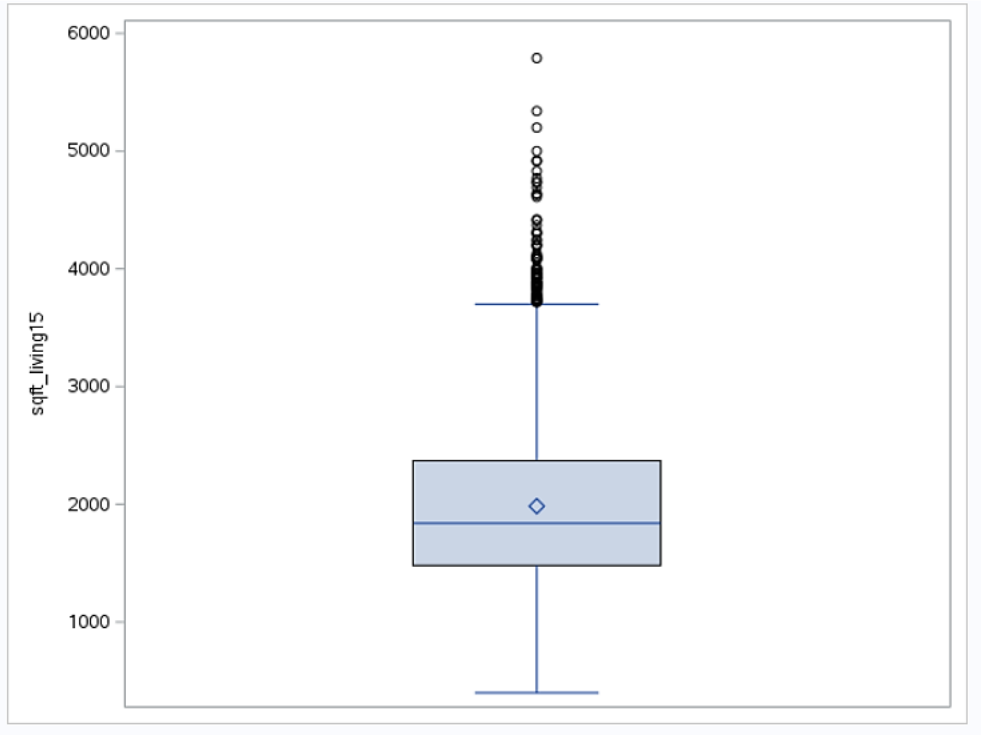
Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
399	2508	4920	435
670	401	5000	2027
710	488	5200	2938
740	2337	5340	917
740	1475	5790	854

Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	5	0.15	100.00

Summarizing Properties:

- There are 5 missing values.
- The average square footage of interior housing living space for the nearest 15 neighbours of the houses within the dataset is 1984.13 square feet.
- The standard deviation proposes that the neighbourhoods might be mixed in terms of house sizes, with both smaller and larger homes present in close proximity.
- The positive skewness value of 1.09188649 suggests that the distribution is skewed to the right. Meaning, while many neighbourhoods might have homes with typical sizes, there are certain neighbourhoods or areas where houses tend to be larger, essentially pulling the tail of the distribution to the right.
- The kurtosis value proposes a distribution with light tail and a relatively flat peak, meaning that there are fewer extreme outliers and a broader central range of values.
- No inconsistencies were detected.

Outliers:



The boxplot above suggests the presence of outliers within the *sqft_living15* variable. Despite that and upon inspection, outliers within the boxplot for the *sqft_living15* variable does in fact correspond to that done for the *sqft_living* variable. Hence, the outliers outlined above are considered genuine and will not be removed.

sqft_lot15

Source Code:

```
/* sqft_lot15 VARIABLE */
/* Compute quartiles for the 'sqft_lot15' variable */
PROC UNIVARIATE DATA=house;
    VAR sqft_lot15;
    OUTPUT OUT=OutliersSqftLot15 (RENAME=(sqft_lot15=OriginalSqftLot15))
    Q1=Q1_sqft_lot15 Q3=Q3_sqft_lot15;
RUN;

/* Detect and store outliers for 'sqft_lot15' in a new dataset */
DATA OutliersListSqftLot15 (keep=ObsNum OutlierValue);
    SET house;
    IF _N_ = 1 THEN SET OutliersSqftLot15;
    IQR = Q3_sqft_lot15 - Q1_sqft_lot15;
    LowerBound = Q1_sqft_lot15 - 1.5 * IQR;
    UpperBound = Q3_sqft_lot15 + 1.5 * IQR;
    IF sqft_lot15 < LowerBound OR sqft_lot15 > UpperBound THEN DO;
        ObsNum = _N_;
        OutlierValue = sqft_lot15;
        OUTPUT;
    END;
    DROP IQR LowerBound UpperBound Q1_sqft_lot15 Q3_sqft_lot15;
RUN;

/* Print detected outliers for 'sqft_lot15' */
PROC PRINT DATA=OutliersListSqftLot15;
RUN;

/* Visualize 'sqft_lot15' distribution with a boxplot */
PROC SGPLOT DATA=house;
    VBOX sqft_lot15;
RUN;
```

Output:

The UNIVARIATE Procedure
Variable: sqft_lot15

Moments			
N	3248	Sum Weights	3248
Mean	12710.7694	Sum Observations	41284579
Std Deviation	26707.22	Variance	713275602
Skewness	8.06072448	Kurtosis	82.5909775
Uncorrected SS	2.84076E12	Corrected SS	2.31601E12
Coeff Variation	210.114897	Std Error Mean	468.619843

Basic Statistical Measures			
Location		Variability	
Mean	12710.77	Std Deviation	26707
Median	7563.50	Variance	713275602
Mode	5000.00	Range	434069
		Interquartile Range	4976

Quantiles (Definition 5)	
Level	Quantile
100% Max	434728.0
99%	155073.0
95%	36567.0
90%	17859.0
75% Q3	10016.0
50% Median	7563.5
25% Q1	5040.0
10%	3501.0
5%	1789.0
1%	1173.0
0% Min	659.0

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
659	191	298182	1957
817	475	310582	2259
915	2667	380279	132
942	1267	411962	1212
967	2208	434728	1566

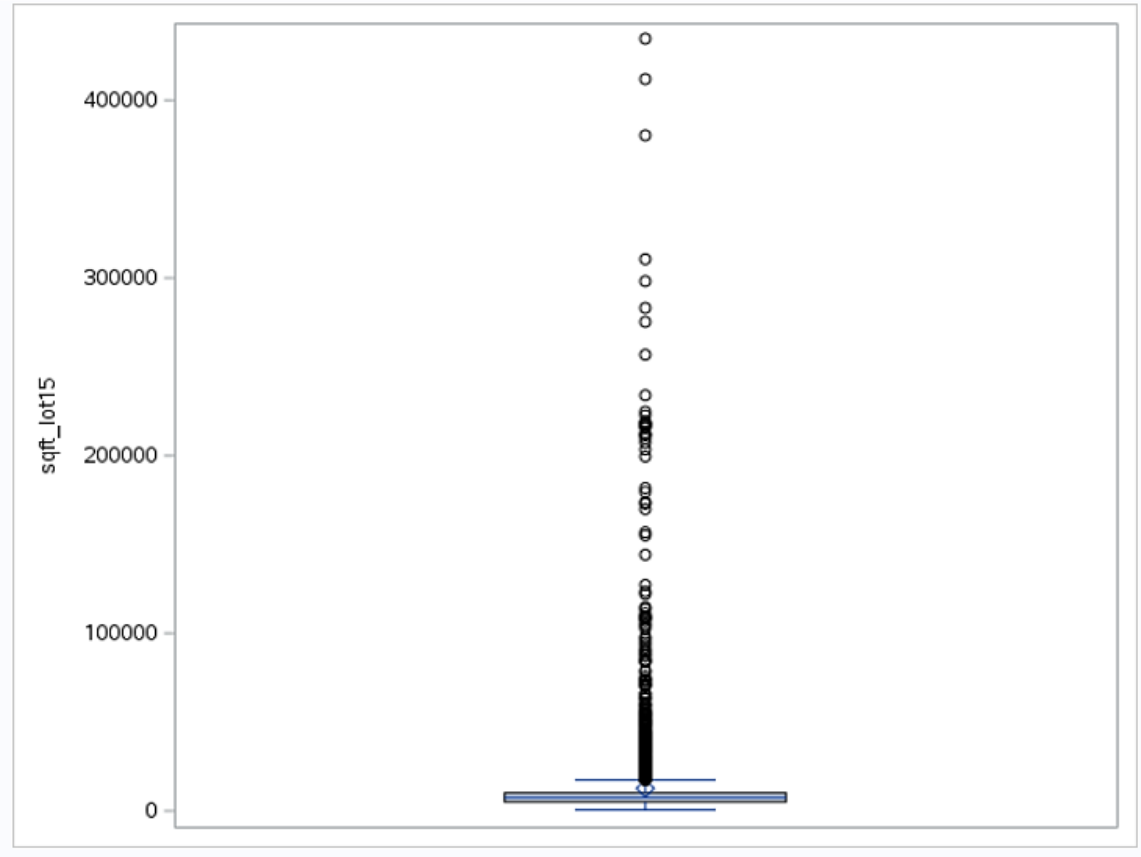
Missing Values			
Missing Value	Count	Percent Of	
		All Obs	Missing Obs
.	2	0.06	100.00

Summarizing Properties:

- There are 2 missing values.
- The average square footage of the land lots of the nearest 15 neighbours of the houses within the dataset is 12710.77 square feet.

- The high standard deviation proposes that there is a wide variation in the sizes of lot spaces for the nearest 15 neighbours as compared to the houses within the dataset.
- The high positive skewness value suggests that the distribution is skewed to the right. Meaning, there are a few lot sizes which are much larger size than the rest, essentially pulling the average up.
- The high kurtosis value suggests that the distribution has very heavy tails. In other words, there are extreme values present in the distribution.
- No inconsistencies were detected.

Outliers:



The boxplot above suggests that there is the issue of outliers within the distribution. Nevertheless, when examined further, the outliers identified above does in fact correspond to those identified for the *sqft_lot* variable. With that, these outliers will be treated as legitimate data points and hence, will be retained within the dataset.

4.0 Data Warehouse Concepts

A data warehouse is a type of data management system that aggregates data from various disparate sources for instance, those from operational databases, transactional systems, web logs as well as external sources from third parties to name a few (IBM, n.d.-a). Through the consolidation of these data which prove to be sizable in volume and diverse in origins, users would then be able to benefit from it, in terms of both the quality and depth of their analysis. Valuable business insights could then be derived, essentially enriching the decision-making processes of businesses. That being said, 5 distinct criteria which aim to serve as a baseline checklist when evaluating the various data warehousing concepts will be discussed below. In doing that, Landbay which is a United Kingdom-based buy-to-let mortgage lending platform will be used as an example to illustrate the 5 criteria. Before that however, it is important to note that even though there do exist dissimilarity between Landbay and the above chosen dataset, where the former is of fintech nature while the latter does not have such component attached to it, relevancy between the company and the dataset is still present because both are still within the same domain of property market.

4.1 Five Criteria Checklist

The 5 criteria are *Scalability and Performance, Data Integration, Data Governance*, *Flexibility and Adaptability*, and lastly, *Cost Efficiency*.

4.1.1 Criteria 1 – Scalability and Performance

The first criterion of *Scalability and Performance* asserts that as data increases in volume, the data warehouse should be able to handle this increased volume, without having to compromise on performance (StoneFly, n.d.). The architecture of the chosen data warehouse should thereby be one that is prepared to provide for either vertical and/or horizontal scaling so as to accommodate for the future data growth of the business. Having said that, scalability does vary with the choice of either an on-premise, cloud-based or hybrid data warehouse solution. Cloud-based data warehouses for example, provide an almost immediate scalability as compared to the other two options (Mariani, 2019). Performance on the other hand refers not just only in terms of the warehouse's ability to handle large data, but also in terms of its ability to deliver quick query responses, more so when analytical tasks are complex. Throughput, latency and concurrency are metrics used to measure the performance of a data warehouse. Throughput refers to the number of jobs or requests that can be handled per unit of time (Burke, n.d.); Latency refers to the amount of time required to perform a single job or

request (Amazon Web Services, n.d.-a); and, Concurrency refers to how well the system performs when two or more users are making multiple requests at the same time (Massachusetts Institute of Technology, 2016).

4.1.2 Criteria 2 – Data Integration

The second criterion of *Data Integration* relates to the data warehouse's ability to support diverse data sources, ranging from those that are structured to those that are semi-structured. There are two choices of data integration pipelines namely the ETL (Extract, Transform, Load) technique and the ELT (Extract, Load, Transform) technique (Amazon Web Services, n.d.-b).. ETL is a conventional data integration technique, where user would extract data from disparate sources, transform the data into a standardized format and structure, and finally load the transformed data into a destination database or into a data warehouse. ETL is appropriate for when users have a predefined data model already, an established set of queries and reports, and a limited volume of data (IBM, n.d.-b). The three main advantages of ETL pipelines are enhanced data quality, optimized performance and heightened security. As data is cleaned and standardized before it is being stored, the ETL technique is thereby able to ensure that data quality is being maintained. Transforming data before storing them also implies that performance of the database or data warehouse will be optimized for querying and reporting to be carried out. Lastly, because transformation is done before data is being loaded, data that are confidential and sensitive could then be masked or encrypted before finally being stored. Given these advantages, there are however some downsides to the ETL data integration technique. One of the downsides being that, because ETL pipelines are generally designed for batch processing, it is thus unsuitable to accommodate for real-time data needs. Besides, since a predefined model is necessary for the implementation of the pipeline, flexibility and scalability would then be restricted, more so when attempting to accommodate for new sources, formats and/or requirements.

ELT on the other hand, is a newer technique of data integration. The ELT pipeline begins when users extract data from different sources, load the data into a destination database or data warehouse with no transformation done at this point, and finally performing the necessary transformation on demand using the destination's processing power (IBM, n.d.-c). Such type of data integration is suited best for when data models are dynamic and that various queries and analytical works are required to be performed on a huge volume of data. Comparing ETL to ELT, the latter is more able to ingest data faster because, unlike the former which performs an additional step of transformation before loading the data, the latter can instead load

the data directly and transform the data in parallel. Greater flexibility and scalability, as well as the ability for more complex data exploration are also features that set ELT apart from ETL. Cost wise, and when dealing with cloud-based or hybrid architectures, ELT has the upper hand too because even though contemporary ETL pipelines has been embracing cloud-based data warehouses, such pipelines still do require a separate engine to perform data transformation before storing it into the cloud. ELT on the flip side, eliminates the need for an intermediate processing engine to carry out the transformation task. This is because after extracting data from the relevant sources, these data are then loaded directly into the destination database or data warehouse, and the database or data warehouse would then perform the transformation using its inherent processing capabilities from there onwards, essentially eliminating the additional layer that the ETL possesses. While more businesses are opening their doors to ELT, there are nevertheless still some limitations to it. One of it being that to be able to effectively perform transformations on the data within the ELT data integration pipeline, it is necessary for company's personnels to possess specialized expertise and knowledge of the employed data warehouse's processing capabilities. Besides, since transformation is done concurrently while loading the data, redundancy may occur in that, raw data and the transformed data may coexist unnecessarily, ultimately resulting in an increased of storage requirements. Having said the above, regardless of the choice of pipelines chosen, the bottom line is that the process should be seamless, and backed with specific support for data ingestion that is required by the particular business.

4.1.3 Criteria 3 – Data Governance

The third criterion of *Data Governance* asserts that apart from making sure that data is accurate, consistent, and reliable, data warehouses should also ensure that data integrity and security is maintained either through audit trails, data lineage and/or access controls to name a few (IBM, n.d.-d).. The chosen data warehouse should be one that support compliance needs and is in line with regulatory bodies for instance the California's Consumer Privacy Act (CCPA) and the European Union's General Data Protection Regulation (GDPR) to name a few. Apart from the more apparent reason of avoiding legal and financial repercussions, a compliant vendor will ensure that while regulatory landscapes are constantly changing, their customer to whom they provide data warehousing solutions to will still remain compliant without needing to invest in additional resources for any monitoring or adaption purposes. Besides staying abreast of the latest regulatory changes and updates, choosing a vendor which prioritizes regulatory compliance will also boost stakeholder's confidence in that, it is indication of a

business' responsible and future-proof operations. Also, because regulatory compliance often mandates for stringent data security and privacy standards, a compliant vendor is thus more likely to have had a comprehensive security mechanism, ultimately lowering the risk of data breaches, unauthorized access or corruption to the data.

4.1.4 Criteria 4 – Flexibility and Adaptability

The fourth criterion of *Flexibility and Adaptability* suggests that when choosing a data warehouse, the choice should be one that not only provides support for conventional Business Intelligence tasks, but also for newer analytical needs, for instance, predictive modeling, machine learning and real-time analytics just to mention a few. Integration between existing tools, platforms and softwares with newer and contemporary ones should be done in a seamless manner (Twilio Inc, n.d.). Another indicator of flexibility is the availability of Application Programming Interfaces (APIs) or more commonly known as connectors. These connectors are what allows the data warehouse to communicate with various other systems when performing data exchange and integration. Similarly, being flexible and adaptable also means that as data needs changes with time, in terms of its nature and/or structure, the chosen data warehouse should be able to facilitate for schema alterations, without having to involve enormous rework, downtime or overheads. Essentially, the idea of flexibility and adaptability in the context of a data warehouse must not merely be about the warehouse's ability to house more data or different types of data. Instead, the grander scheme of both idea is about making sure that the chosen data warehouse remains a valuable asset for the business, even as the business' analytical needs, tools and/or data sources advance.

4.1.5 Criteria 5 – Cost Efficiency

The fifth and last criterion which is *Cost Efficiency* asserts that, while the initial setup cost of the data warehouse is one factor to be considered, the total expenditure of ownership which includes those of the maintenance work, scaling and operations should also be taken into account (Bhanot, 2019). The pricing models of the data warehouses considered should be compared and contrasted with one another so as to ensure that, budget planning and operational strategies are aligned with the pricing models of the chosen data warehouse. Alignment with the business' budget planning is important because knowing how the business will be charged enables the business to not just allocate resources accordingly but also to appropriately predict the expenses it bears. This would ensure that the business is able to maintain a healthy financial standing without any unexpected surprises on its bill. Similarly, ensuring that operational strategy is in line with the pricing model is important because decisions about of how frequently

will the data warehouse be used, what datasets will be stored and how long these data will be retained are all factors which will be influenced by the pricing models of the chosen warehousing solution. Given that, being fully aware of the costs associated with the various data operations of the business would then help the business in making informed and strategic choices. In fact, most contemporary cloud-based data warehousing solutions are changing the way businesses think about costs. Rather than a huge initial outlay, cloud solutions are basing their pricing models on the concept of operational expenditure. Under such pricing models, businesses would only pay for what they have used. In essence and having reviewed all five criteria above, it is important to understand that there is really just no one-size-fits-all approach when evaluating which data warehouse solution to go with. Instead, every business's individual requirement, current tech stack, budget constraints and future growth should direct their ultimate decision.

4.2 Case Study on Landbay Based on the Five Criteria

4.2.1 Landbay's Company Background

Landbay, as previously mentioned, is a fintech mortgage lending platform that allows institutional investors to fund buy-to-let residential mortgages (Landbay, n.d.). The company has an extensive funding base with Retail Banks, Investment Banks and Asset Managers, where loans are funded through, both the securitization process in the capital markets as well as through bank deposits. Landbay's business model involves two sets of customers, namely the institutional lenders (investors) and borrowers. Institutional lenders are those large entities with investment goals and risk appetites that they are looking to meet within their portfolios. These investors look to Landbay to provide for decision insights. The borrowers on the other hand, are those who possess varying degree of credit risk and are those who would come to Landbay to seek loans. Landbay then plays the intermediary to match the goals of both sets of customers. Institutional investors could fund loans to their preferred timelines and outcomes, while borrowers would in return receive the fund that they require.

4.2.2 Evaluation on Landbay's Existing Data Warehouse

Based on the most recent knowledge which was according to an article published on the 27th of July 2020, Landbay has been engaging with Amazon Redshift as its data warehouse provider at that point of time. Recall that, that period in time was also when the COVID-19 pandemic was still ongoing. Similar to all other sectors, the pandemic has also adversely

affected the buy-to-let mortgage market such that originations were lower, capital market funding were tightened and risk appetite of investors has significantly shrunk due to the uncertainty surrounding the pandemic (Amazon Web Services, n.d.-c). Movement controls during the lockdown has also caused property viewings to not be able to take place, essentially resulting in a halt in the issuing of new mortgage offers on the part of Landbay. Government's initiative in easing the impact of the pandemic includes the introduction of mortgage holidays as well as special servicing. The real challenge then was to implement all of these government initiatives in an environment where all employees were 100% working remotely. Hence, to be successful in doing that, Landbay needed additional data and insights to firstly, put in place the processes necessary to facilitate mortgage holidays for its borrowers while still ensuring adherence to the regulatory guidelines outlined by the government; at the same time, additional data and insights are required to enable Landbay to continually make timely and accurate decisions for both the request for payment holidays as well as plans to exit payment holidays, essentially making sure that its customers had the best outcomes; lastly, additional data and insights is required to compare the performance of the resulting mortgage holidays against Landbay's industry peers.

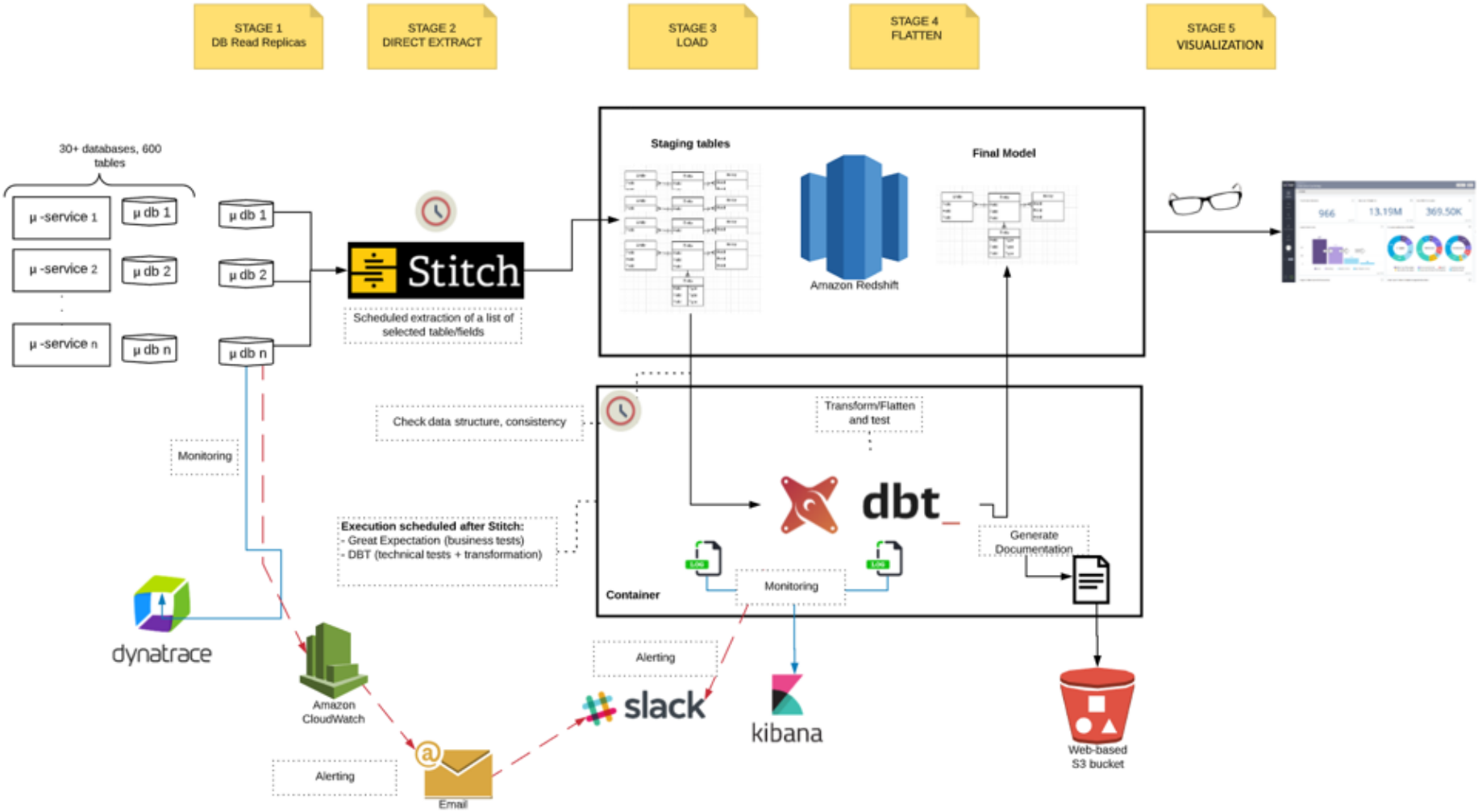


Figure 1: : Initial Data Pipeline at Landbay (Amazon Web Service, n.d.-c)

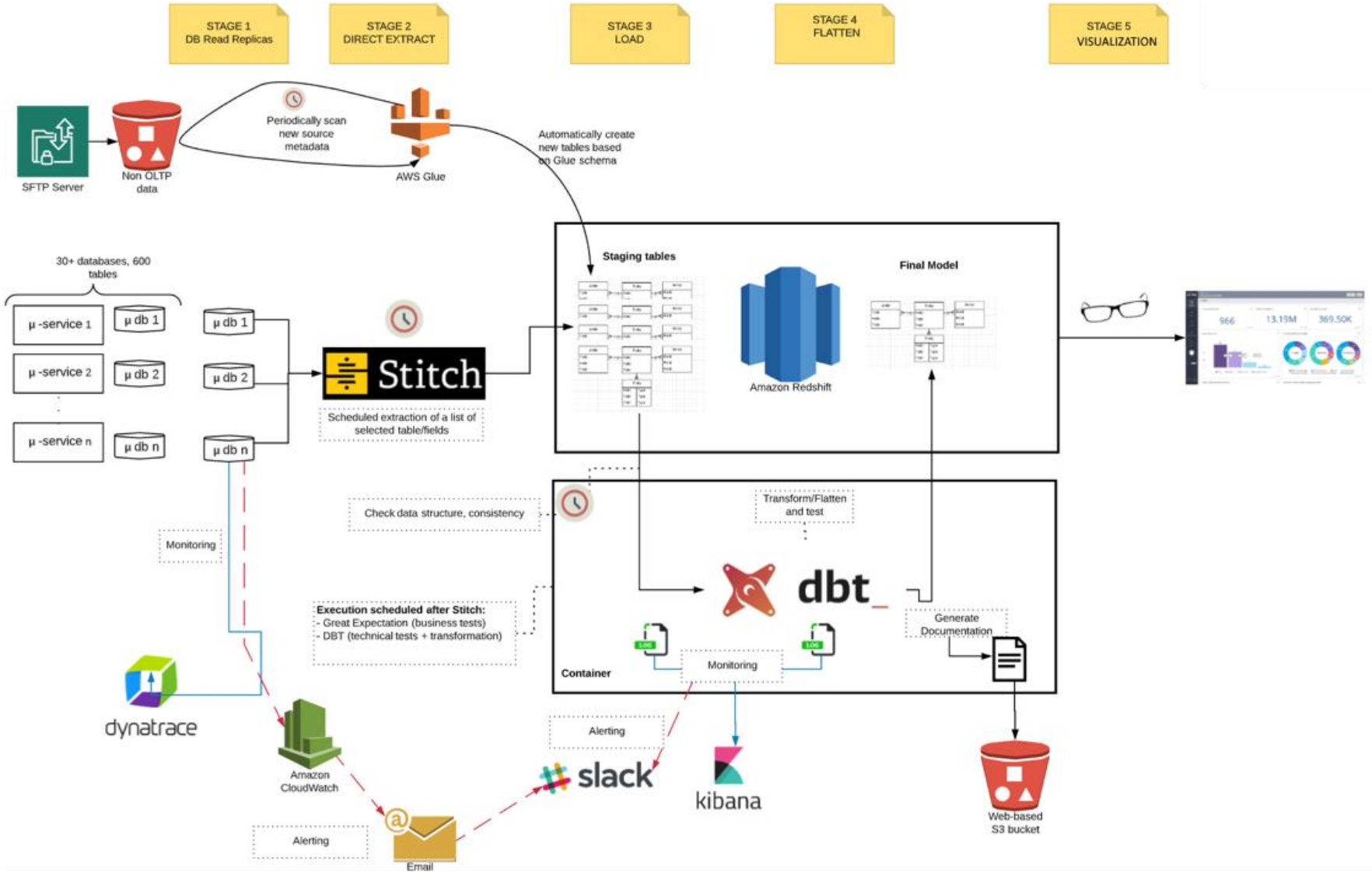


Figure 2: : Latest Data Pipeline at Landbay (Amazon Web Service, n.d.-c)

The diagram above shows how Landbay's data pipeline works and how each of its components has been instrumental in implementing the said requirements above (Amazon Web Services, n.d.-c). The data pipeline is built on top of a microservice architecture and it uses the ELT concepts (*Criterion 2 – Data Integration*) to send all of its operational data into its data warehouse. There are three fundamental elements to the data pipelines namely the source data, where MariaDB RDS and read-replicas are used to offload the data extraction outside of any OLTP or real-time operations traffic; Amazon Redshift, which is the cloud-based data repository (*Criterion 1 – Scalability and Performance*) for all of the extracted data from the microservices; and lastly, DBT which is a SQL-based data tool, used to transform the source data for end-user visualization purposes.

As one could see from the diagram above, Landbay has been utilizing AWS Glue and Amazon Redshift to load its data. That is, through Glue's crawler, Landbay could automatically produce a catalogue of data tables simply by going through a set of S3 buckets. Once the scanning is done, the process will then prompt Glue to uncover the schema and semantics of the data held within each respective bucket. When the crawler is running, it will then convert the insights about the data into a table in the Glue data catalogue. While Glue configured these data as Spectrum tables into Landbay's data warehouse, data could then be transformed and visualized whenever needed. Put simply, the entire process starts with the creating of a bucket, the configuring of the bucket in Glue and subsequently the uploading of some data into the bucket. Following that, the tables will appear in Redshift automatically. SQL query will be used to then transform the incoming data into the necessary fact table using DBT. Visualization could then be done after all that and end users will now be able to get their data within a short period of time, somewhere between a mere 2 to 4 hours. On a side note, working on the background is one of Redshift's features called the Redshift Spectrum, which was used to make S3 files accessible simply through a create table statement. This remarkable feature was what makes the onboarding process of new data sources straightforward. Instead of having to build an entirely new microservice, data could now be transferred swiftly onto the platform with the Redshift Spectrum feature.

Building on that, as Redshift Spectrum bills its customers on a "per byte scanned" basis, storage in the form of columnar format is thereby to Landbay's advantage in that, it allows Landbay to optimize both performance and cost all at once. In fact, Landbay found out that by configuring their larger data sets to use the Parquet format, the company was actually able to

achieve a 30x cost reduction on some of their data sources. Such cost reduction was possible because unlike before, only a fraction of the CSV file needs to be read now for each query, essentially bring the cost down. Similarly, in terms of performance, an increase of 3 to 5x was observed as well, proving that the Redshift Spectrum was really a game changer for Landbay. From here, even though it was not explicitly mentioned, it is very likely that the choice of Redshift as its data warehouse provider was influenced by the cost reduction (*Criterion 5 – Cost Efficiency*) and boosted performance (*Criterion 1 – Scalability and Performance*) that comes with the warehousing solution. Again, while it was not explicitly mentioned, the choice of an ELT data pipeline is likely due to Landbay's heavy reliance on real-time analytics so as to provide institutional lenders with timely insights when spotting trends, essentially enabling investors to strategize and optimize their portfolio performance, once again illustrating how *Criterion 2 – Data Integration* is at play. The attribute of being flexible and adaptable (*Criterion 4*) was also demonstrated by Redshift in that, even when Landbay had implemented a change in tools from initially the use of Stitch (a third-party provider) to AWS Glue, because the latter was more time-efficient in terms of extracting data into the data warehouse as compared to the former, Redshift still remained a valuable asset to Landbay nonetheless. Lastly, in terms of Data Governance (*Criterion 3 – Data Governance*) , Amazon Redshift prided itself as a provider that maintains more security standards and compliance certifications than any other provider, and that includes ISO 27001, SOC, HIPAA/HITECH, and FedRAMP (Amazon Web Services, n.d.-d). Also, security features offered by Redshift are at no additional cost for all of its consumers. Even though not specifically expressed, these two perks from Redshift could potentially be one of the contributing factors in Landbay's decision to go with Redshift instead of with any other data warehouse provider.

5.0 Conclusion

This paper consists of two parts namely the exploration of the dataset on housing prices within King Country, Washington, and the evaluation of data warehousing concepts using Landbay for the case study. Exploration of the dataset includes that on the metadata, the summarizing properties of all attributes within the dataset, as well as the detection of inconsistencies on each attribute. In terms of the data warehousing concepts, 5 criteria were identified and discussed so that businesses could use them as a checklist to guide their decision when choosing a data warehousing provider. The five criteria are namely *Scalability and Performance*, *Data Integration*, *Data Governance*, *Flexibility and Adaptability*, and lastly, *Cost Efficiency*. Each criterion was examined and cross-referenced to the components of the data pipeline within Landbay.

Appendices

Appendix 1

RScript

```r
# Load necessary library
install.packages("dplyr")
library(dplyr)

# Read the dataset
df <- read.csv("House.csv")

# Set seed for reproducibility
set.seed(123)

# Reduce dataset to 3,250 observations
df_reduced <- df %>% sample_n(3250)

# Introduce 60 random missing values
for(i in 1:60) {
  # Select a random row and column
  row <- sample(1:nrow(df_reduced), 1)
  col <- sample(2:ncol(df_reduced), 1)  # starting from 2 to avoid the 'id' column

  # Introduce a missing value
  df_reduced[row, col] <- NA
}

# Check for missing values within the entire dataset
total_missing_values <- sum(is.na(df_reduced))
print(total_missing_values)
```

```r

# Check for missing values for each column separately
missing_values_per_column <- colSums(is.na(df_reduced))
print(missing_values_per_column)

# Save the modified dataset as House_Dataset
write.csv(df_reduced, "House_Dataset.csv", row.names = FALSE)

```

Appendix 2

Description of All Attributes

Variable Name	Descriptions
id	Unique ID for each home sold
date	Date of the home sale
price	Price of each home sold
bedrooms	Number of bedrooms
bathrooms	Number of bathrooms, where .5 accounts for a room with a toilet but no shower
sqft_living	Square footage of the apartments interior living space
sqft_lot	Square footage of the land space
floors	Number of floors
waterfront	A dummy variable for whether the apartment was overlooking the waterfront or not
view	An index from 0 to 4 of how good the view of the property was
condition	An index from 1 to 5 on the condition of the apartment. (1 – Poor; 2 – Fair; 3 – Average; 4 – Good; 5 – Very Good)
grade	An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design. (1 – Cabin; 2- Substandard; 3 – Poor; 4 – Low; 5 – Fair; 6 – Low Average; 7 – Average; 8 – Good; 9 – Better; 10 – Very Good; 11 – Excellent; 12 – Luxury; 13 – Mansion)
sqft_above	The square footage of the interior housing space that is above ground level
sqft_basement	The square footage of the interior housing space that is below ground level
yr_built	The year the house was initially built
yr_renovated	The year of the house's last renovation
zipcode	What zip code area the house is in
lat	Latitude
long	Longitude
sqft_living15	The square footage of interior housing living space for the nearest 15 neighbours
sqft_lot15	The square footage of the land lots of the nearest 15 neighbours

References

Amazon Web Services. (n.d.-a). What is network latency?. Retrieved from https://aws.amazon.com/what-is/latency/

Amazon Web Services. (n.d.-b). What is data integration?. Retrieved from https://aws.amazon.com/what-is/data-integration/

Amazon Web Services. (n.d.-c). Using Amazon Redshift & AWS Glue: How Landbay pivoted to provide mortgage payment holidays. Retrieved from https://aws.amazon.com/blogs/startups/how-landbay-uses-amazon-redshift-glue-for-mortgage-payment-holidays/

Amazon Web Services. (n.d.-d). Amazon Redshift security & governance. Retrieved from https://aws.amazon.com/redshift/security/

Bhanot, P. (2019). A cost/benefit guide to the data warehouse. Retrieved from https://www.actian.com/blog/cloud-data-warehouse/a-cost-benefit-guide-to-the-data-warehouse/

Burke, J. (n.d.). Throughput. Retrieved from https://www.techtarget.com/searchnetworking/definition/throughput

Harlfoxem. (2016). *House sales in King County, USA* [Data set]. Retrieved from https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/discussion

IBM. (n.d.-a). What is a data warehouse?. Retrieved from https://www.ibm.com/topics/data-warehouse

IBM. (n.d.-b). ETL (extract, transform, load). Retrieved from https://www.ibm.com/topics/etl

IBM. (n.d.-c). What is ELT (extract, load, transform)?. Retrieved from https://www.ibm.com/topics/elt

IBM. (n.d.-d). What is data governance?. Retrieved from https://www.ibm.com/topics/data-governance

Landbay. (n.d.). Your lending partner. Retrieved from https://landbay.co.uk/about-us/

Mariani, D. (2019). What is a cloud data warehouse?. Retrieved from https://www.atscale.com/blog/what-is-a-cloud-data-warehouse/

Massachusetts Institute of Technology. (2016). Reading 19: Concurrency. Retrieved from https://ocw.mit.edu/ans7870/6/6.005/s16/classes/19-concurrency/index.html

StoneFly. (n.d.). Understanding scalability in data storage. Retrieved from https://stonefly.com/blog/understanding-scalability-in-data-storage/

Twilio Inc. (n.d.). How to choose the right data warehouse. Retrieved from https://segment.com/academy/choosing-stack/how-to-choose-the-right-data-warehouse/