

STAT 506 Final Project

For this project, the dataset that I chose to utilize was real Airbnb listings, which I found on InsideAirbnb, a website that lists all of the Airbnb datasets and the connected listings. In this dataset, it has information on the property listings, which include the listing ID, the host's information, the types of rooms in the property, the cost, how many bedrooms there are, the neighborhood that it is in, and any review metrics. I imported the dataset from a CSV file called Listings.csv, and I cleaned and analyzed the data in SAS. It had approximately 181,928 observations, along with multiple numeric variables and character variables for my demonstration.

The topics that I completed are as follows (including the questions answered):

1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 24, 25, 26

1. PG1.02 — Accessing Data (Import, LIBNAME, PROC CONTENTS)
 - a. 1 – Importing data
 - b. 2 – Access SAS tables in a library
2. PG1.03 — Exploring and Validating Data
 - a. 3 – Print SAS tables
 - b. 4 – Identify and remove duplicates
3. PG1.04 — Preparing Data
 - a. 5 – DATA step creation
 - b. 6 - Three IF-THEN statements
4. PG1.05 — Analyzing and Reporting on Data
 - a. 7 – Titles and footnotes
 - b. 8 – Frequency reports
 - c. 9 – Summary statistics with PROC MEANS
5. PG1.07 — Using SQL in SAS
 - a. 12 – PROC SQL
 - b. 13 – SQL merge using aliases
6. PG2.01 — Controlling DATA Step Processing
 - a. 14 – SELECT group and processing options
7. PG2.02 — Summarizing Data
 - a. 15 - Data summarization
8. PG2.03 — Manipulating Data with Functions
 - a. 16 – CALL routines
 - b. 17 – Date functions
 - c. 18 – Character functions
 - d. 19 – Search/Replace/Cleaning functions
 - e. 20 – Numeric/stat functions
 - f. 21 – Converting data types
9. PG2.05 — Combining Tables
 - a. 24 – Concatenation and simple merge
 - b. 25 – Merging with non-matches
10. PG2.06 — Processing Repetitive Code (DO Loops)
 - a. 26 – DO loops

```

1a)
%let projpath = /home/u64313069/pg3;          /* (setup, not graded) */
%let csvpath = &projpath./Listings.csv;        /* (setup, not graded) */

libname project "&projpath.";                  /* 2a */

/*
PG1.02 - Accessing Data
Topic 1: Importing data (1a, 1b)
Topic 2: Access SAS tables in a library (2a, 2b, 2c)
*/
/* 1a: PROC IMPORT step */
proc import datafile="&csvpath."
  out=project.airbnb_raw
  dbms=CSV
  replace;
run;                                         /* 1a */

1b)
/* 1b: PROC CONTENTS for imported table, VARNUM option */
proc contents data=project.airbnb_raw varnum;
  title "Contents of Airbnb Raw Listing Data (VarNum Order)";
run;                                         /* 1b */

```

Contents of Airbnb Raw Listing Data (VarNum Order)

The CONTENTS Procedure

Data Set Name	PROJECT.AIRBNB_RAW	Observations	279712
Member Type	DATA	Variables	33
Engine	V9	Indexes	0
Created	11/29/2025 12:34:33	Observation Length	744
Last Modified	11/29/2025 12:34:33	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	1590
First Data Page	1
Max Obs per Page	176
Obs in First Data Page	167
Number of Data Set Repairs	0
Filename	/home/u64313069/pg3/airbnb_raw.sas7bdat
Release Created	9.0401M8
Host Created	Linux
Inode Number	15233973039
Access Permission	rw-r--r--
Owner Name	u64313069
File Size	199MB
File Size (bytes)	208535552

Variables in Creation Order					
#	Variable	Type	Len	Format	Informat
1	listing_id	Num	8	BEST12.	BEST32.
2	name	Char	53	\$53.	\$53.
3	host_id	Num	8	BEST12.	BEST32.
4	host_since	Num	8	YYMMDD10.	YYMMDD10.
5	host_location	Char	30	\$30.	\$30.
6	host_response_time	Char	1	\$1.	\$1.
7	host_response_rate	Char	1	\$1.	\$1.
8	host_acceptance_rate	Char	1	\$1.	\$1.
9	host_is_superhost	Char	1	\$1.	\$1.
10	host_total_listings_count	Num	8	BEST12.	BEST32.
11	host_has_profile_pic	Char	1	\$1.	\$1.
12	host_identity_verified	Char	1	\$1.	\$1.
13	neighbourhood	Char	19	\$19.	\$19.
14	district	Char	1	\$1.	\$1.
15	city	Char	5	\$5.	\$5.
16	latitude	Num	8	BEST12.	BEST32.
17	longitude	Num	8	BEST12.	BEST32.
18	property_type	Char	16	\$16.	\$16.
19	room_type	Char	12	\$12.	\$12.
20	accommodates	Num	8	BEST12.	BEST32.
21	bedrooms	Num	8	BEST12.	BEST32.
22	amenities	Char	456	\$456.	\$456.
23	price	Num	8	BEST12.	BEST32.

24	minimum_nights	Num	8	BEST12.	BEST32.
25	maximum_nights	Num	8	BEST12.	BEST32.
26	review_scores_rating	Num	8	BEST12.	BEST32.
27	review_scores_accuracy	Num	8	BEST12.	BEST32.
28	review_scores_cleanliness	Num	8	BEST12.	BEST32.
29	review_scores_checkin	Num	8	BEST12.	BEST32.
30	review_scores_communication	Num	8	BEST12.	BEST32.
31	review_scores_location	Num	8	BEST12.	BEST32.
32	review_scores_value	Num	8	BEST12.	BEST32.
33	instant_bookable	Char	1	\$1.	\$1.

2a)

```
/* 2a: LIBNAME statement (already above, repeated here as explicit use) */
libname project "&projpath."; /* 2a */

1      OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69      /* 2a: LIBNAME statement (already above, repeated here as explicit use) */
70      libname project "&projpath.";

NOTE: Libref PROJECT was successfully assigned as follows:
      Engine:      V9
      Physical Name: /home/u64313069/pg3
70      !
71
72      OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
82
```

2b)

```
/* 2b: PROC CONTENTS for the library using _ALL_ and NODS */
proc contents data=project._all_nods;
  title "Contents of PROJECT Library (All Tables)";
run; /* 2b */
```

Contents of PROJECT Library (All Tables)

The CONTENTS Procedure

Directory	
Libref	PROJECT
Engine	V9
Physical Name	/home/u64313069/pg3
Filename	/home/u64313069/pg3
Inode Number	15233971423
Access Permission	rwxr-xr-x
Owner Name	u64313069
File Size	0KB
File Size (bytes)	156

#	Name	Member Type	File Size	Last Modified
1	AIRBNB_RAW	DATA	199MB	11/29/2025 18:38:10

2c)

```
/* 2c: PROC CONTENTS for a single table in the library */
proc contents data=project.airbnb_raw;
  title "Contents of project.airbnb_raw";
run; /* 2c */

title; /* clear titles (housekeeping
```

Contents of project.airbnb_raw

The CONTENTS Procedure

Data Set Name	PROJECT.AIRBNB_RAW	Observations	279712
Member Type	DATA	Variables	33
Engine	V9	Indexes	0
Created	11/29/2025 12:38:09	Observation Length	744
Last Modified	11/29/2025 12:38:09	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	1590
First Data Page	1
Max Obs per Page	176
Obs in First Data Page	167
Number of Data Set Repairs	0
Filename	/home/u64313069/pg3/airbnb_raw.sas7bdat
Release Created	9.0401M8
Host Created	Linux
Inode Number	15233973039
Access Permission	rw-r--r--
Owner Name	u64313069
File Size	199MB
File Size (bytes)	208535552

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
20	accommodates	Num	8	BEST12.	BEST32.
22	amenities	Char	456	\$456.	\$456.
21	bedrooms	Num	8	BEST12.	BEST32.
15	city	Char	5	\$5.	\$5.
14	district	Char	1	\$1.	\$1.
8	host_acceptance_rate	Char	1	\$1.	\$1.
11	host_has_profile_pic	Char	1	\$1.	\$1.
3	host_id	Num	8	BEST12.	BEST32.
12	host_identity_verified	Char	1	\$1.	\$1.
9	host_is_superhost	Char	1	\$1.	\$1.
5	host_location	Char	30	\$30.	\$30.
7	host_response_rate	Char	1	\$1.	\$1.
6	host_response_time	Char	1	\$1.	\$1.
4	host_since	Num	8	YYMMDD10.	YYMMDD10.
10	host_total_listings_count	Num	8	BEST12.	BEST32.
33	instant_bookable	Char	1	\$1.	\$1.
16	latitude	Num	8	BEST12.	BEST32.
1	listing_id	Num	8	BEST12.	BEST32.
17	longitude	Num	8	BEST12.	BEST32.
25	maximum_nights	Num	8	BEST12.	BEST32.
24	minimum_nights	Num	8	BEST12.	BEST32.
2	name	Char	53	\$53.	\$53.
13	neighbourhood	Char	19	\$19.	\$19.

23	price	Num	8	BEST12.	BEST32.
18	property_type	Char	16	\$16.	\$16.
27	review_scores_accuracy	Num	8	BEST12.	BEST32.
29	review_scores_checkin	Num	8	BEST12.	BEST32.
28	review_scores_cleanliness	Num	8	BEST12.	BEST32.
30	review_scores_communication	Num	8	BEST12.	BEST32.
31	review_scores_location	Num	8	BEST12.	BEST32.
26	review_scores_rating	Num	8	BEST12.	BEST32.
32	review_scores_value	Num	8	BEST12.	BEST32.
19	room_type	Char	12	\$12.	\$12.

3)

```
/* Create a small cleaned subset just for exploration */
data project.airbnb_explore;
  set project.airbnb_raw;
  /* Keep some key columns, including host_id now */
  keep listing_id host_id name city district neighbourhood
    property_type room_type accommodates bedrooms
    price review_scores_rating;
run;

/* Formats used in question 3 (e) */
proc format;
  value pricefmt
    low-50      = "Budget (<=50)"
    50<-150    = "Mid-range (50-150)"
    150<-high  = "High-end (>150)";
  value ratingfmt
    0-80       = "Low"
    80<-95     = "Medium"
    95<-100    = "High";
run;
```

3a, b, c, d, e)

```
/* 3a-3e: PROC PRINT with OBS=, VAR, WHERE IN, formats */
proc print data=project.airbnb_explore (obs=10); /* 3a, 3b */
  var listing_id name city neighbourhood room_type
    price review_scores_rating; /* 3c */
  where room_type in ("Entire home/apt","Private room"); /* 3d */
  format price pricefmt.
    review_scores_rating ratingfmt.; /* 3e */
  title "First 10 Airbnb Listings - Selected Rooms and Formats";
run;
title; /* housekeeping */
```

First 10 Airbnb Listings – Selected Rooms and Formats

Obs	listing_id	name	city	neighbourhood	room_type	price	review_scores_rating
78269	718761	2 private BRs, private bath - Like your OWN APT	New Y	Williamsburg	Private room	Mid-range (50-150)	Medium
78272	1755844	Brooklyn Stunning Event Space.	New Y	Crown Heights	Private room	High-end (>150)	Medium
78281	17605478	NANCY Room Private Key lock Private entrance	New Y	Bushwick	Private room	Budget (<=50)	Medium
78283	19362649	upstairs single with queens bed and all decorative	New Y	Briarwood	Private room	Mid-range (50-150)	Medium
78286	32562943	94TH STREET 2BR-2BA SUITE WITH DOORMAN-GYM	New Y	Upper East Side	Private room	High-end (>150)	Low
78287	36911220	Kitty à»éçàà§à¥—æ^çì¼private ba	New Y	Flushing	Private room	Mid-range (50-150)	Low
78306	32173127	Lovely Studio @ BTS Nana, Bumrungrad Hospital	Bangk	Ratchathewi	Private room	High-end (>150)	Low
78316	5971310	Single room (shared bathroom) (3)å–@àºæ^ç	Hong	Yau Tsim Mong	Private room	Mid-range (50-150)	Medium
78317	10743742	New modern Studio #A	Hong	Yau Tsim Mong	Private room	High-end (>150)	Low
78321	19429191	Single Room Shared Bathroom (Monthly Rental)	Hong	Yau Tsim Mong	Private room	High-end (>150)	Medium

4a)

```
/* 4a: Use PROC FREQ to identify duplicate HOST IDs */
proc freq data=project.airbnb_explore;
  tables host_id / nocum nopercnt;
  title "Frequency of Host IDs (Checking for Duplicate Hosts)";
run;
title;
```

Frequency of Host IDs (Checking for Duplicate Hosts)

The FREQ Procedure

host_id	Frequency
1822	1
1944	11
2330	3
2353	1
2438	1
2571	1
2626	2
2782	3
2787	8
2827	1
2845	2
2868	1
2881	2
3047	5
3085	1
3280	3
3504	7
3607	2

(continues on for more, but chose to not show them all for space)

4b)

```
/* 4b: Use PROC SORT to remove unwanted duplicate rows */
proc sort data=project.airbnb_explore
           out=project.airbnb_nodups nodupkey;
by host_id;
run; /* 4b */
```

```
1      OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69      proc sort data=project.airbnb_explore
70              out=project.airbnb_nodups nodupkey;
71          by host_id;
72      run;
```

NOTE: There were 279712 observations read from the data set PROJECT.AIRBNB_EXPLORE.

NOTE: 97688 observations with duplicate key values were deleted.

NOTE: The data set PROJECT.AIRBNB_NODUPS has 182024 observations and 12 variables.

5a, b, c, d, e, f)

```
data project.airbnb_prep;
set project.airbnb_nodups;                                /* 5a */

where city ne "" and price > 0;                            /* 5b */

/* Keep only relevant columns in final output */
keep listing_id name city neighbourhood district property_type room_type
    accommodates bedrooms price minimum_nights maximum_nights
    review_scores_rating instant_bookable avg_price_per_night
    stay_category long_stay_flag;                           /* 5c */

length stay_category $12 long_stay_flag $3;                  /* 5e */

format price dollar8.2;                                     /* 5d */

/* Create new columns using assignment statements */
avg_price_per_night = price / max(1, minimum_nights); /* 5f (assignment #1) */

if maximum_nights > 28 then                                /* 5f (assignment #2 inside :*/
    stay_category = "Long stay";
else stay_category = "Short stay";
```

6a, b, c)

7a, b, c, d)

```
/* Macro vars for titles/footnotes (used in 7a-7d) */
%let projcity    = Airbnb City Listings;                      /* 7d (macro var) */
%let datasource = Public Airbnb CSV;                          /* 7d (macro var) */

data project.airbnb_labeled;
  set project.airbnb_preppe;
  label
    listing_id      = "Listing ID"
    city            = "City"
    neighbourhood   = "Neighbourhood"
    room_type       = "Room Type"
    price           = "Total Price"
    avg_price_per_night = "Average Price per Night";
run;                                                       /* 7e (labels defined) */

/* 7a: TITLE using macro variable */
/* 7b: TITLE2 subtitle */
/* 7c: FOOTNOTE */
/* 7d: Macro variables used in title/footnote */
title1 "&projcity.";                                     /* 7a, 7d */
title2 "Selected Listings Summary";                     /* 7b */
footnote "Data source: &datasource.";                  /* 7c, 7d */
```

7e)

```
/* 7e: Display labels in PROC PRINT using LABEL option */
proc print data=project.airbnb_labeled (obs=15) label;
  var listing_id city neighbourhood room_type
      price avg_price_per_night;
run;                                         /* 7e */
```

Airbnb City Listings Selected Listings Summary

Obs	Listing ID	City	Neighbourhood	Room Type	Total Price	Average Price per Night
1	23537464	Rome	V Prenestino/Centoc	Entire place	\$52.00	52
2	31647336	Rome	X Ostia/Acilia	Private room	\$40.00	40
3	5148970	Rome	IX Eur	Private room	\$60.00	60
4	9587470	Rome	I Centro Storico	Entire place	\$80.00	80
5	23669201	New Y	Williamsburg	Entire place	\$95.00	95
6	1101224	New Y	Bedford-Stuyvesant	Entire place	\$117.00	117
7	43559737	Paris	Opera	Entire place	\$46.00	46
8	43564916	New Y	Upper East Side	Entire place	\$95.00	95
9	10267242	New Y	Gravesend	Private room	\$149.00	149
10	2577	Paris	Enclos-St-Laurent	Entire place	\$125.00	125
11	2595	New Y	Midtown	Entire place	\$100.00	100
12	13234457	New Y	Clinton Hill	Entire place	\$60.00	60
13	6027345	New Y	Bedford-Stuyvesant	Private room	\$39.00	39
14	31476548	Rome	VIII Appia Antica	Private room	\$22.00	22
15	538460	Rome	XII Monte Verde	Entire place	\$30.00	30

Data source: Public Airbnb CSV

7f)

```
/* 7f: Clear titles and footnote */
title;                                         /* 7f */
footnote;                                       /* 7f */
```

8a)

```
/* ---- Question 8: Frequency reports ---- */

/* 8a: One-way frequency with ORDER= and output to table */
proc freq data=project.airbnb_preped order=freq;
  tables room_type / out=project.freq_roomtype;
  title "Frequency of Room Type (Most Common First)";
run;                                         /* 8a */
title;

proc format;
  value accomfmt
    low-2    = "1-2 guests"
    3-4      = "3-4 guests"
    5-high   = "5+ guests";
run;                                         /* 8b (for 8c) */


```

Frequency of Room Type (Most Common First)

The FREQ Procedure

room_type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Entire place	126109	69.32	126109	69.32
Private room	52056	28.61	178165	97.93
Shared room	2623	1.44	180788	99.37
Hotel room	1140	0.63	181928	100.00

8b)

```

/* 8b: Two-way frequency, formats, suppress stats */
proc freq data=project.airbnb_prep;
  tables room_type * accommodates / norow nocol nopercent;
  format accommodates accomfmt.;
  title "Cross-tab of Room Type by Accommodates";
run;
title;

```

Cross-tab of Room Type by Accommodates				
The FREQ Procedure				
Frequency	Table of room_type by accommodates			
	accommodates			
room_type	1-2 guests	3-4 guests	5+ guests	Total
Entire place	47252	49977	28880	126109
Hotel room	629	403	108	1140
Private room	44788	5789	1479	52056
Shared room	2030	321	272	2623
Total	94699	56490	30739	181928

```

9a,b,c,d)
/* ---- Question 9: PROC MEANS summary stats ---- */

/* 9a: Non-default stats (mean, median, std, min, max)
/* 9b: VAR statement
/* 9c: CLASS + WAYS
/* 9d: Output to new table */
proc means data=project.airbnb_preppe
            mean median std min max;
    var price avg_price_per_night review_scores_rating; /* 9b */
    class room_type; /* 9c */
    ways 1; /* 9c */
    output out=project.means_roomtype
        mean(price avg_price_per_night review_scores_rating)=
        median(price)=MedianPrice; /* 9d */
    title "Summary Statistics of Prices and Ratings by Room Type";
run; /* 9a-9d */
title;

```

Summary Statistics of Prices and Ratings by Room Type

The MEANS Procedure

room_type	N Obs	Variable	Mean	Median	Std Dev	Minimum	Maximum
Entire place	126109	price	551.510033	144.000000	2408.31	1.000000	300000.00
		avg_price_per_night	551.510033	144.000000	2408.31	1.000000	300000.00
		review_scores_rating	94.1547949	97.000000	8.8976175	20.000000	100.0000000
Hotel room	1140	price	836.8289474	250.0000000	2930.83	12.000000	72999.00
		avg_price_per_night	836.8289474	250.0000000	2930.83	12.000000	72999.00
		review_scores_rating	90.4674479	95.0000000	13.6688907	20.000000	100.0000000
Private room	52056	price	402.1092477	90.000000	4013.83	8.000000	499000.00
		avg_price_per_night	402.1092477	90.000000	4013.83	8.000000	499000.00
		review_scores_rating	93.2924499	97.000000	11.9373891	20.000000	100.0000000
Shared room	2623	price	662.2436142	89.000000	12407.37	5.000000	625216.00
		avg_price_per_night	662.2436142	89.000000	12407.37	5.000000	625216.00
		review_scores_rating	90.3697318	96.000000	16.1050984	20.000000	100.0000000

12a)

```

/* 12a: Create a new table with subset of columns */
proc sql;
    create table project.sql_subset as
    select listing_id, city, neighbourhood, room_type,
           price, avg_price_per_night, review_scores_rating
    from project.airbnb_preppe;
quit; /* 12a */

```

12b)

```
/* 12b: Display subset based on WHERE clause */
proc sql;
    select listing_id, city, room_type, price, review_scores_rating
    from project.sql_subset
    where room_type = "Entire place"
        and price between 50 and 200;
quit;                                         /* 12b */
```

listing_id	city	room_type	price	review_scores_rating
23537464	Rome	Entire place	\$52.00	90
9587470	Rome	Entire place	\$80.00	91
23669201	New Y	Entire place	\$95.00	.
1101224	New Y	Entire place	\$117.00	98
43564916	New Y	Entire place	\$95.00	.
2577	Paris	Entire place	\$125.00	100
2595	New Y	Entire place	\$100.00	94
13234457	New Y	Entire place	\$60.00	80
3801938	Rome	Entire place	\$105.00	.
3109	Paris	Entire place	\$60.00	100
246916	New Y	Entire place	\$96.00	96
3831	New Y	Entire place	\$73.00	90
2075600	New Y	Entire place	\$139.00	94
38569438	Paris	Entire place	\$65.00	97
6751450	New Y	Entire place	\$190.00	94
14869215	New Y	Entire place	\$135.00	96
5396	Paris	Entire place	\$81.00	90
46544	New Y	Entire place	\$150.00	94
302758	New Y	Entire place	\$150.00	90
36025823	Paris	Entire place	\$180.00	100
18873947	Rome	Entire place	\$100.00	90
23135	New Y	Entire place	\$125.00	94

(this is only a portion of the rows, chose not to display them all because of size)

12c)

```
/* 12c: Display table sorted by two columns */
proc sql;
    select city, room_type, price, review_scores_rating
    from project.sql_subset
    order by city, price desc;
quit; /* 12c */
```

city	room_type	price	review_scores_rating
Bangk	Private room	300177.0	95
Bangk	Private room	300000.0	.
Bangk	Entire place	300000.0	.
Bangk	Entire place	180124.0	.
Bangk	Entire place	160000.0	89
Bangk	Private room	137031.0	80
Bangk	Entire place	100000.0	.
Bangk	Entire place	100000.0	98
Bangk	Private room	100000.0	92
Bangk	Entire place	99285.00	.
Bangk	Entire place	99000.00	100
Bangk	Private room	98000.00	100
Bangk	Entire place	89000.00	.
Bangk	Entire place	75000.00	.
Bangk	Private room	75000.00	.
Bangk	Private room	70000.00	.
Bangk	Private room	70000.00	98
Bangk	Private room	65000.00	.
Bangk	Entire place	65000.00	.
Bangk	Private room	60000.00	.
Bangk	Private room	60000.00	.
Bangk	Private room	60000.00	.
Bangk	Private room	54135.00	.

(this is only a portion of the rows, chose not to display them all because of size)

13)

```

/* Helper tables for 13a-13b (join) */
proc sql;
    create table project.city_avg_price as
    select city,
        mean(price) as city_avg_price
    from project.airbnb_prep
    group by city;
quit;

proc sql;
    create table project.roomtype_avg_rating as
    select room_type,
        mean(review_scores_rating) as rt_avg_rating
    from project.airbnb_prep
    group by room_type;
quit;

13a, b)
/* 13a: Display result of merging at least three tables
   13b: Use aliases */
proc sql;
    create table project.join3 as
    select a.listing_id,
        a.city,
        a.room_type,
        a.price,
        b.city_avg_price,
        c.rt_avg_rating
    from project.airbnb_prep      as a
        left join project.city_avg_price      as b
            on a.city = b.city
        left join project.roomtype_avg_rating as c
            on a.room_type = c.room_type;
quit;                                /* 13a, 13b */

proc print data=project.join3 (obs=10);
run;

```

Obs	listing_id	city	room_type	price	city_avg_price	rt_avg_rating
1	29185610	Sydney	Entire place	\$400.00	207.236	94.1548
2	43526850	Sydney	Entire place	\$145.00	207.236	94.1548
3	8831946	Sydney	Entire place	\$390.00	207.236	94.1548
4	22892949	Sydney	Entire place	\$119.00	207.236	94.1548
5	19298810	Sydney	Entire place	\$210.00	207.236	94.1548
6	11191291	Sydney	Entire place	\$175.00	207.236	94.1548
7	21898520	Sydney	Entire place	\$175.00	207.236	94.1548
8	8849197	Sydney	Entire place	\$345.00	207.236	94.1548
9	3648797	Sydney	Entire place	\$150.00	207.236	94.1548
10	4144830	Sydney	Entire place	\$90.00	207.236	94.1548

14a,b,c)

```

data project.low_price
      project.high_price;
set project.airbnb_prepended (keep=listing_id city room_type price);
/* 14b (KEEP= option) */

/* 14c: At least two informative PUTLOG statements */
if price <= 0 then putlog "WARNING: Non-positive price for listing " listing_id=; /* 14c */
if city = "" then putlog "NOTE: Missing city for listing " listing_id=; /* 14c */

/* 14a: SELECT group, OTHERWISE, explicit output to multiple tables */
select;
  when (price < 75) output project.low_price;      /* 14a(2) explicit output */
  when (price >= 75) output project.high_price;    /* 14a(2) explicit output */
  otherwise putlog "NOTE: Price not categorized: " listing_id= price=; /* 14a(1) OTHERWISE */
end;
run;                                              /* 14a, 14b, 14c */

```

NOTE: There were 181928 observations read from the data set PROJECT.AIRBNB_PREPPED.

NOTE: The data set PROJECT.LOW_PRICE has 48414 observations and 4 variables.

NOTE: The data set PROJECT.HIGH_PRICE has 133514 observations and 4 variables.

NOTE: DATA statement used (Total process time):

real time	0.06 seconds
user cpu time	0.01 seconds
system cpu time	0.01 seconds
memory	5088.93k
OS Memory	34740.00k
Timestamp	11/29/2025 07:27:06 PM
Step Count	224 Switch Count 7
Page Faults	0
Page Reclaims	877
Page Swaps	0
Voluntary Context Switches	257
Involuntary Context Switches	1
Block Input Operations	0
Block Output Operations	14360

15a,b,c,d)

```

/* 15d: PROC SORT */
proc sort data=project.airbnb_prepended
            out=project.by_city;
    by city;
run;                                         /* 15d */

/* 15a: RETAIN
15b: Sum statement
15c: Use FIRST. and LAST. */
data project.city_summary;
    set project.by_city;
    by city;

    retain total_price 0 n_listings 0;           /* 15a */

    if first.city then do;                      /* 15c */
        total_price = 0;
        n_listings = 0;
    end;

    total_price + price;                      /* 15b (sum statement) */
    n_listings + 1;                           /* 15b */

    if last.city then do;                     /* 15c */
        avg_price = total_price / n_listings;
        output;
    end;
run;

proc print data=project.city_summary (obs=10);
    var avg_price total_price n_listings;
    title "Numeric Summary Columns from CITY_SUMMARY";
run;
title;

```

Numeric Summary Columns from CITY_SUMMARY

Obs	avg_price	total_price	n_listings
1	2330.19	19382510	8318
2	2121.10	23571772	11113
3	1067.10	2176887	2040
4	490.10	6645284	13559
5	1174.85	13523708	11511
6	149.34	3989574	26715
7	101.54	5265721	51859
8	695.37	12028507	17298
9	97.46	1419754	14568
10	207.24	5169907	24947

17a, b, c, d, e)

```

data project.func_demo;
  set project.airbnb_preped (obs=100);

length clean_name $200 short_neighbourhood $40;           /* (support for 18) */

/* ---- 17: Date-related functions ---- */

today_date = today();                                     /* 17a */
months_back = mod(listing_id, 6);                         /* helper */

booking_date = intnx('month', today_date, -months_back, 'same'); /* 17e (INTNX)
format today_date booking_date date9.;

first_of_booking_month =
  mdy(month(booking_date), 1, year(booking_date)); /* 17b (MDY) */

booking_year      = year(booking_date);                  /* 17c (YEAR) */
booking_month     = month(booking_date);                 /* 17c (MONTH) */
booking_weekday   = weekday(booking_date);              /* 17c (WEEKDAY) */
days_since_booking = intck('day', booking_date, today_date); /* 17d (INTCK) */

/* ---- 18: Character functions ---- */

```

18 a,b,c,d,e,f)

```

/* ---- 18: Character functions ---- */

name_length = length(name);                                /* 18b (LENGTH) */
name_prefix = substr(name, 1, 15);                         /* 18a (SUBSTR) */

short_neighbourhood =
  propcase(scan(neighbourhood, 1, ','));             /* 18c (SCAN), 18d (PROPCASE)

city_upper = upcase(city);                               /* 18e (UPCASE) */

city_room = catx(' - ', city, room_type);            /* 18f (CATX) */

```

19 a,b,c,d,e)

```

/* ---- 19: FIND, SUBSTR on left, TRANWRD, COMPRESS, STRIP ---- */

/* 19a: FIND - look for the word "room" in the listing name (case-insensitive) */
pos_room = find(name, "room", 'i');                                /* 19a (FIND) */

/* 19b: SUBSTR on the left side - emphasize the word "room" if present */
name_modified = name;                                         /* create a writable copy */
if pos_room > 0 then substr(name_modified, pos_room, 4) = "ROOM";
                                                               /* 19b (SUBSTR on left) */

/* 19c: TRANWRD - replace 'St.' with 'Street' in neighbourhood text */
neighbourhood_clean = tranwrd(neighbourhood, "St.", "Street"); /* 19c (TRANWRD) */

/* 19d: COMPRESS - remove spaces from neighbourhood */
neighbourhood_nospace = compress(neighbourhood);           /* 19d (COMPRESS) */

/* 19e: STRIP - remove leading/trailing blanks from city */
city_stripped = strip(city);                                 /* 19e (STRIP) */

```

20 a,b,c)

```

/* ---- 20: Numeric/stat functions ---- */

score_mean      = mean(of review_scores_rating);          /* 20b (MEAN) */

rounded_price   = round(price, 1);                         /* 20a (ROUND) */
ceil_price       = ceil(price);                           /* 20a (CEIL) */
floor_price      = floor(price);                          /* 20a (FLOOR) */
int_price        = int(price);                            /* 20a (INT) */

high_metric     = largest(1, price, avg_price_per_night); /* 20b (LARGEST) */
min_metric       = min(price, avg_price_per_night);       /* 20b (MIN) */
max_metric       = max(price, avg_price_per_night);       /* 20b (MAX) */
sum_metric       = sum(price, avg_price_per_night);       /* 20b (SUM) */

n_nonmissing    = n(price, bedrooms, review_scores_rating); /* 20c (N) */
n_missing        = nmiss(price, bedrooms, review_scores_rating); /* 20c (NMISS) */
c_missing        = cmiss(city, neighbourhood, property_type); /* 20c (CMISS) */

```

16 a, b)

```

/* ---- 16: CALL routines ---- */

if room_type = "" then
  call missing(city_room, rounded_price);           /* 16a (CALL MISSING) */

x1 = price;
x2 = avg_price_per_night;
x3 = review_scores_rating;
call sortn(of x1-x3);                            /* 16b (CALL SORTN) */

run;

/*
PG2.03 - Manipulating Data with Functions
Topics 16-21
16: CALL routines
17: Date functions
18: Character functions (SUBSTR, LENGTH, SCAN, PROPCASE, UPCASE, CATX)
19: FIND, SUBSTR on left, TRANWRD, COMPRESS, STRIP/trim/compbl
20: At least 8 numeric/stat functions across categories
21: Converting column type
*/

```

```

proc print data=project.func_demo (obs=10);
var
  /* Original identifiers */
  listing_id city neighbourhood price

  /* 17: Date functions */
  today_date booking_date first_of_booking_month days_since_booking

  /* 18: Character functions */
  name_length name_prefix short_neighbourhood city_upper city_room

  /* 19: FIND + SUBSTR-left + TRANWRD + COMPRESS + STRIP */
  pos_room name_modified neighbourhood_clean neighbourhood_nospace city_stripped

  /* 20: Numeric/stat functions */
  rounded_price ceil_price floor_price int_price
  high_metric min_metric max_metric sum_metric
  n_nonmissing n_missing c_missing

  /* 16: CALL routines */
  x1 x2 x3
;
  title "PG2.03 Demonstration (Topics 16-21)";
run;
title;

```

PG2.03 Demonstration (Topics 16–21)

Obs	listing_id	city	neighbourhood	price	today_date	booking_date	first_of_booking_month	days_since_booking	name_length	name_prefix	short_neighbourhood	city_upper	city_room	pos_room	name_modified	neighbourhood_clean	
1	23537464	Rome	V Prenestino/Centoc	\$52.00	29NOV2025	29JUL2025		23923	123	28	Sant'Ele	V Prenestino/Centoc	ROME	Rome - Entire place	0	Sant'Ele Elena Guesthouse	V Prenestino/Centoc
2	31647336	Rome	X Ostia/Acilia	\$40.00	29NOV2025	29NOV2025		24046	0	42	Camera arredata	X Ostia/Acilia	ROME	Rome - Private room	0	Camera arredata zona Stella Polare in apt	X Ostia/Acilia
3	5148970	Rome	IX Eur	\$60.00	29NOV2025	29JUL2025		23923	123	43	Suite penthouse	IX Eur	ROME	Rome - Private room	0	Suite penthouse in Villa Spinaceto Roma Eur	IX Eur
4	9587470	Rome	I Centro Storico	\$80.00	29NOV2025	29JUL2025		23923	123	48	SPACIOUS APARTM	I Centro Storico	ROME	Rome - Entire place	0	SPACIOUS APARTMENT WITH BIG TERRACE AT COLOSSEUM	I Centro Storico
5	23669201	New Y	Williamsburg	\$95.00	29NOV2025	29JUN2025		23893	153	50	Great Price: WI	Williamsburg	NEW Y	New Y - Entire place	0	Great Price: Williamsburg Brooklyn Loft off L stop	Williamsburg
6	1101224	New Y	Bedford-Stuyvesant	\$117.00	29NOV2025	29SEP2025		23985	61	50	THE PUTNAM-3 Bd	Bedford-Stuyvesant	NEW Y	New Y - Entire place	0	THE PUTNAM-3 Bd w/Outdoor Zen space/washer&dryer	Bedford-Stuyvesant
7	43559737	Paris	Opera	\$46.00	29NOV2025	29OCT2025		24015	31	50	STUDIO-OPERA/GR	Opera	PARIS	Paris - Entire place	0	STUDIO-OPERA/GRADS BOULEVARDS-1 PEOPLE-LONG TERM	Opera
8	43564916	New Y	Upper East Side	\$95.00	29NOV2025	29SEP2025		23985	61	43	Sweet Spot - Ca	Upper East Side	NEW Y	New Y - Entire place	0	Sweet Spot - Calm, Clean, Comfy, Accessible	Upper East Side
9	10267242	New Y	Gravesend	\$149.00	29NOV2025	29NOV2025		24046	0	43	Corona free. C	Gravesend	NEW Y	New Y - Private room	40	Corona free. Clean and Quiet Queen BedROOM	Gravesend
10	2577	Paris	Enclos-St-Laurent	\$125.00	29NOV2025	29AUG2025		23954	92	32	Loft for 4 by C	Enclos-St-Laurent	PARIS	Paris - Entire place	0	Loft for 4 by Canal Saint Martin	Enclos-St-Laurent

neighbourhood_nospace	city_stripped	rounded_price	ceil_price	floor_price	int_price	high_metric	min_metric	max_metric	sum_metric	n_nonmissing	n_missing	c_missing	x1	x2	x3
V Prenestino/Centoc	Rome	52	52	52	52	52	52	52	104	3	0	0	52	52	90
X Ostia/Acilia	Rome	40	40	40	40	40	40	40	80	3	0	0	40	40	100
IX Eur	Rome	60	60	60	60	60	60	60	120	3	0	0	60	60	75
I Centro Storico	Rome	80	80	80	80	80	80	80	160	3	0	0	80	80	91
Williamsburg	New Y	95	95	95	95	95	95	95	190	2	1	0	.	95	95
Bedford-Stuyvesant	New Y	117	117	117	117	117	117	117	234	3	0	0	98	117	117
Opera	Paris	46	46	46	46	46	46	46	92	2	1	0	46	46	100
UpperEastSide	New Y	95	95	95	95	95	95	95	190	2	1	0	.	95	95
Gravesend	New Y	149	149	149	149	149	149	149	298	3	0	0	95	149	149
Enclos-St-Laurent	Paris	125	125	125	125	125	125	125	250	3	0	0	100	125	125

21 a, b, c, d)

```
/* 21a-21d: Converting column types + PROC CONTENTS */

data project.type_conversion;
set project.func_demo;

bedrooms_char = put(bedrooms, 8.);
bedrooms_num   = input(bedrooms_char, 8.);                                /* 21b helper + 21b? */
                                                               /* 21a (INPUT: char->num) */

price_char     = put(price, dollar8.2);                                 /* 21b (PUT: num->char) */

auto_conv = bedrooms_char + 1;                                         /* 21c (automatic conversion in log) */

run;                                                               /* 21d */

proc contents data=project.type_conversion;
  title "Type Conversion Demo – Showing Character/Numeric Types";
run;                                                               /* 21d */
title;
```

Type Conversion Demo – Showing Character/Numeric Types

The CONTENTS Procedure

Data Set Name	PROJECT.TYPE_CONVERSION	Observations	100
Member Type	DATA	Variables	54
Engine	V9	Indexes	0
Created	11/29/2025 13:45:34	Observation Length	1192
Last Modified	11/29/2025 13:45:34	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	1
First Data Page	1
Max Obs per Page	109
Obs in First Data Page	100
Number of Data Set Repairs	0
Filename	/home/u64313069/pg3/type_conversion.sas7bdat
Release Created	9.0401M8
Host Created	Linux
Inode Number	15233977327
Access Permission	rw-r--r--
Owner Name	u64313069
File Size	256KB
File Size (bytes)	262144

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
8	accommodates	Num	8	BEST12.	BEST32.
54	auto_conv	Num	8		
14	avg_price_per_night	Num	8		
9	bedrooms	Num	8	BEST12.	BEST32.
51	bedrooms_char	Char	8		
52	bedrooms_num	Num	8		
21	booking_date	Num	8	DATE9.	
24	booking_month	Num	8		
25	booking_weekday	Num	8		
23	booking_year	Num	8		
47	c_missing	Num	8		
38	ceil_price	Num	8		
5	city	Char	5	\$5.	\$5.
30	city_room	Char	200		
35	city_stripped	Char	5		
29	city_upper	Char	5		
17	clean_name	Char	200		
26	days_since_booking	Num	8		
4	district	Char	1	\$1.	\$1.
22	first_of_booking_month	Num	8		
39	floor_price	Num	8		
41	high_metric	Num	8		
40	int_price	Num	8		
1	listing_id	Num	8	BEST12.	BEST32.
13	long_stay_flag	Char	3		
43	max_metric	Num	8		
16	maximum_nights	Num	8		

42	min_metric	Num	8		
15	minimum_nights	Num	8		
20	months_back	Num	8		
46	n_missing	Num	8		
45	n_nonmissing	Num	8		
2	name	Char	53	\$53.	\$53.
27	name_length	Num	8		
32	name_modified	Char	53		
28	name_prefix	Char	53		
3	neighbourhood	Char	19	\$19.	\$19.
33	neighbourhood_clean	Char	200		
34	neighbourhood_nospace	Char	19		
31	pos_room	Num	8		
10	price	Num	8	DOLLAR8.2	BEST32.
53	price_char	Char	8		
6	property_type	Char	16	\$16.	\$16.
11	review_scores_rating	Num	8	BEST12.	BEST32.
7	room_type	Char	12	\$12.	\$12.
37	rounded_price	Num	8		
36	score_mean	Num	8		
18	short_neighbourhood	Char	40		
12	stay_category	Char	12		
44	sum_metric	Num	8		
19	today_date	Num	8	DATE9.	
48	x1	Num	8		
49	x2	Num	8		
50	x3	Num	8		

24)

```
/*
PG2.05 - Combining Tables
Topic 24: Concatenate, simple merge (24a, 24b)
Topic 25: Merge with non-matches + options (25a, 25b)
*/
/* Split into two tables for concatenation and merging */
data project.entire_home project.private_room;
  set project.airbnb_prep;
  if room_type = "Entire home/apt" then output project.entire_home;
  else if room_type = "Private room" then output project.private_room;
run; /* (prep for 24, 25) */
```

24a)

```
/* 24a: Concatenate two tables using RENAME= option */
data project.concat_example;
  set project.entire_home
    project.private_room (rename=(price=price_private));
run;
```

24b)

```
/* 24b: Simple ONE-to-ONE merge by listing_id */

/* Build a small table with host_is_superhost from the RAW data */
data project.host_flag;
  set project.airbnb_raw (keep=listing_id host_is_superhost);
run; /* 24b: create host_flag with listing_id + host_is_superhost */

/* Sort both datasets by listing_id for the merge */
proc sort data=project.airbnb_prep;
  by listing_id;
run; /* 24b prep */

proc sort data=project.host_flag;
  by listing_id;
run; /* 24b prep */

/* Simple one-to-one merge on listing_id */
data project.merge_one_to_one;
  merge project.airbnb_prep
    project.host_flag;
  by listing_id;
run; /* 24b */
```

25a, b)

```

/* 25a: Merge with non-matches using IN= and output matches/non-matches
25b: Use RENAME= and KEEP=/DROP= options */

data project.city_info;
  set project.city_summary (rename=(avg_price=city_avg_price)); /* 25b(1) RENAME= */
  keep city city_avg_price; /* 25b(2) KEEP= */
run;

proc sort data=project.airbnb_preped out=project.airbnb_by_city by city; run;
proc sort data=project.city_info; by city; run;

data project.matches
  project.nonmatches;
  merge project.airbnb_by_city (in=inListings)
    project.city_info (in=inCity);
  by city;

  if inListings and inCity then
    output project.matches; /* 25a(1),(2) matches table */
  else if inListings and not inCity then
    output project.nonmatches; /* 25a(1),(2) non-matches table */
run;

proc print data=project.city_info;
  title "PG2.05 (25b): City Info Table with City_Avg_Price";
run;
title;
proc print data=project.matches (obs=10);
  var city listing_id price city_avg_price;
  title "PG2.05 (25a): Sample of MATCHES (Listings with City Info)";
run;
title;

```

PG2.05 (25b): City Info Table with City_Avg_Price

Obs	city	city_avg_price
1	Bangk	2330.19
2	Cape	2121.10
3	Hong	1067.10
4	Istan	490.10
5	Mexic	1174.85
6	New Y	149.34
7	Paris	101.54
8	Rio d	695.37
9	Rome	97.46
10	Sydney	207.24

PG2.05 (25a): Sample of MATCHES (Listings with City Info)

Obs	city	listing_id	price	city_avg_price
1	Bangk	27942	\$901.00	2330.19
2	Bangk	28354	\$1049.00	2330.19
3	Bangk	28745	\$800.00	2330.19
4	Bangk	35656	\$4496.00	2330.19
5	Bangk	35780	\$800.00	2330.19
6	Bangk	35807	\$1649.00	2330.19
7	Bangk	35875	\$2998.00	2330.19
8	Bangk	45859	\$1500.00	2330.19
9	Bangk	48736	\$1351.00	2330.19
10	Bangk	49372	\$649.00	2330.19

26a, b, c, d)


```

/* Show DO loop index example (26a, 26d) */
proc print data=project.do_loop_index;
  title "PG2.06: DO Loop with Index (26a, 26d)";
run;

/* Show a few rows from DO UNTIL example */
proc print data=project.do_until_example (obs=10);
  title "PG2.06: DO UNTIL with BY and OUTPUT (26b, 26c, 26d)";
run;

/* Show a few rows from DO WHILE example */
proc print data=project.do_while_example (obs=10);
  title "PG2.06: DO WHILE with BY and OUTPUT (26b, 26c, 26d)";
run;
title;

```

PG2.06: DO Loop with Index (26a, 26d)

Obs	i	squared	cubed
1	1	1	1
2	2	4	8
3	3	9	27
4	4	16	64
5	5	25	125
6	6	36	216
7	7	49	343
8	8	64	512
9	9	81	729
10	10	100	1000

PG2.06: DO UNTIL with BY and OUTPUT (26b, 26c, 26d)

Obs	listing_id	name	neighbourhood	district	city	property_type	room_type	accommodes	bedrooms	price	review_scores_rating	stay_category	long_stay_flag	avg_price_per_night	minimum_nights	maximum_nights	count_high	threshold
1	43345284	Hong Kong cosy à¼—é¼¢æº™çšæ	Sai Kung		Hong	Tent	Entire place	4	2	\$1.00	.	Short stay	No		1	.	1	300
2	21743681	22 jump street	Waverley		Sydney	Shared room in a	Shared room	3	1	\$5.00	.	Short stay	No		5	.	2	300
3	3666224	Home-sitting chic & center area	Passy		Paris	Private room in	Private room	1	1	\$8.00	80	Short stay			8	.	3	300
4	4348122	Amazing flat in the heart of Paris	Temple		Paris	Entire apartment	Entire place	2	1	\$8.00	.	Short stay	No		8	.	4	300
5	7433116	Bright and cosy room w/big terrace	Buttes-Montmartre		Paris	Private room in	Private room	2	1	\$8.00	100	Short stay	Yes		8	.	5	300
6	12971022	100% Gratis pr sinistres Mac Murray	Popincourt		Paris	Private room in	Private room	2	1	\$8.00	.	Short stay	No		8	.	6	300
7	21366242	Il geco in una stanza	XIV Monte Mario		Rome	Private room in	Private room	4	1	\$8.00	.	Short stay	No		8	.	7	300
8	21883031	Chambre calme quartier Buttes de Chaumont	Buttes-Chaumont		Paris	Private room in	Private room	1	1	\$8.00	.	Short stay	No		8	.	8	300
9	7127245	Bonbon & Bio's home - Roof of Paris	Buttes-Montmartre		Paris	Entire apartment	Entire place	2	1	\$9.00	100	Short stay	Yes		9	.	9	300
10	17635687	Cozy apartment near the Gemelli Metro Station	XIV Monte Mario		Rome	Entire apartment	Entire place	6	2	\$9.00	97	Short stay	Yes		9	.	10	300

PG2.06: DO WHILE with BY and OUTPUT (26b, 26c, 26d)

Obs	listing_id	name	neighbourhood	district	city	property_type	room_type	accommodes	bedrooms	price	review_scores_rating	stay_category	long_stay_flag	avg_price_per_night	minimum_nights	maximum_nights	running_total	extra_flag
1	43345284	Hong Kong cosy à¼—é¼¢æº™çšæ	Sai Kung		Hong	Tent	Entire place	4	2	\$1.00	.	Short stay	No		1	.	1	1
2	21743681	22 jump street	Waverley		Sydney	Shared room in a	Shared room	3	1	\$5.00	.	Short stay	No		5	.	6	1
3	3666224	Home-sitting chic & center area	Passy		Paris	Private room in	Private room	1	1	\$8.00	80	Short stay			8	.	14	1
4	4348122	Amazing flat in the heart of Paris	Temple		Paris	Entire apartment	Entire place	2	1	\$8.00	.	Short stay	No		8	.	22	1
5	7433116	Bright and cosy room w/big terrace	Buttes-Montmartre		Paris	Private room in	Private room	2	1	\$8.00	100	Short stay	Yes		8	.	30	1
6	12971022	100% Gratis pr sinistres Mac Murray	Popincourt		Paris	Private room in	Private room	2	1	\$8.00	.	Short stay	No		8	.	38	1
7	21366242	Il geco in una stanza	XIV Monte Mario		Rome	Private room in	Private room	4	1	\$8.00	.	Short stay	No		8	.	46	1
8	21883031	Chambre calme quartier Buttes de Chaumont	Buttes-Chaumont		Paris	Private room in	Private room	1	1	\$8.00	.	Short stay	No		8	.	54	1
9	7127245	Bonbon & Bio's home - Roof of Paris	Buttes-Montmartre		Paris	Entire apartment	Entire place	2	1	\$9.00	100	Short stay	Yes		9	.	63	1
10	17635687	Cozy apartment near the Gemelli Metro Station	XIV Monte Mario		Rome	Entire apartment	Entire place	6	2	\$9.00	97	Short stay	Yes		9	.	72	1