

Data type constraints

CLEANING DATA IN R



Maggie Matsui

Content Developer @ DataCamp

Course outline



Diagnose dirty
data

Course outline



Diagnose dirty
data



Side effects of
dirty data

Course outline



Diagnose dirty
data



Side effects of
dirty data



Clean data

Course outline



Diagnose dirty
data



Side effects of
dirty data



Clean data

Chapter 1 - Common data problems

Why do we need clean data?



Why do we need clean data?



Why do we need clean data?



Data type constraints

Data type	Example
Text	First name, last name, address, ...
Integer	Subscriber count, # products sold, ...
Decimal	Temperature, exchange rate, ...
Binary	Is married, new customer, yes/no, ...
Category	Marriage status, color, ...
Date	Order dates, date of birth, ...

R data type
<code>character</code>
<code>integer</code>
<code>numeric</code>
<code>logical</code>
<code>factor</code>
<code>Date</code>

Glimpsing at data types

```
sales <- read.csv("sales.csv")  
head(sales)
```

	order_id	revenue	quantity
1	7432	5,454	494
2	7808	5,668	334
3	4893	4,062	259
4	6107	3,936	15
5	7661	1,067	307
6	5908	6,635	235

```
library(dplyr)  
glimpse(sales)
```

```
Observations: 100  
Variables: 3  
$ order_id <dbl> 7432, 7808, ...  
$ revenue <chr> "$5454", "$5668", ...  
$ quantity <dbl> 494, 334, ...
```

Checking data types

```
is.numeric(sales$revenue)
```

```
FALSE
```

```
library(assertive)  
assert_is_numeric(sales$revenue)
```

```
Error: is_numeric : sales$revenue is not of class 'numeric'; it has class 'character'.
```

```
assert_is_numeric(sales$quantity)
```

Checking data types

Logical checking - returns `TRUE` / `FALSE`

- `is.character()`
- `is.numeric()`
- `is.logical()`
- `is.factor()`
- `is.Date()`
- ...

assertive checking - errors when `FALSE`

- `assert_is_character()`
- `assert_is_numeric()`
- `assert_is_logical()`
- `assert_is_factor()`
- `assert_is_date()`
- ...

Why does data type matter?

```
class(sales$revenue)
```

```
"character"
```

```
mean(sales$revenue)
```

```
NA
```

```
Warning message:
```

```
In mean.default(sales$revenue) :
```

```
argument is not numeric or logical: returning NA
```

Comma problems

```
sales$revenue
```

```
"5,454" "5,668" "4,062" "3,936" "1,067" ...
```

Character to number

```
library(stringr)
revenue_trimmed = str_remove(sales$revenue, ",")
revenue_trimmed
```

```
"5454" "5668" "4062" "3936" "1067" ...
```

```
as.numeric(revenue_trimmed)
```

```
5454 5668 4062 3936 1067 ...
```

Putting it together

```
sales %>%  
  mutate(revenue_usd = as.numeric(str_remove(revenue, ",")))
```

```
# A tibble: 100 x 4  
  order_id revenue quantity revenue_usd  
    <dbl> <chr>    <dbl>    <dbl>  
1     7432 5,454      494      5454  
2     7808 5,668      334      5668  
3     4893 4,062      259      4062  
4     6107 3,936       15      3936  
5     7661 1,067      307      1067  
# ... with 95 more rows
```


Same function, different outcomes

```
mean(sales$revenue)
```

```
NA
```

```
Warning message:
```

```
In mean.default(sales$revenue) :
```

```
argument is not numeric or logical: returning NA
```

```
mean(sales$revenue_usd)
```

```
5361.4
```

Converting data types

- `as.character()`
- `as.numeric()`
- `as.logical()`
- `as.factor()`
- `as.Date()`
- ...

Watch out: factor to numeric

```
product_type
```

```
1000 1000 3000 2000 3000  
Levels: 1000 2000 3000
```

```
class(product_type)
```

```
"factor"
```

```
as.numeric(product_type)
```

```
1 1 3 2 3
```

```
as.numeric(as.character(product_type))
```

```
1000 1000 3000 2000 3000
```

Range constraints

CLEANING DATA IN R



Maggie Matsui

Content Developer @ DataCamp

What's an out of range value?

- SAT score: 400-1600
- Package weight: at least 0 lb/kg
- Adult heart rate: 60-100 beats per minute

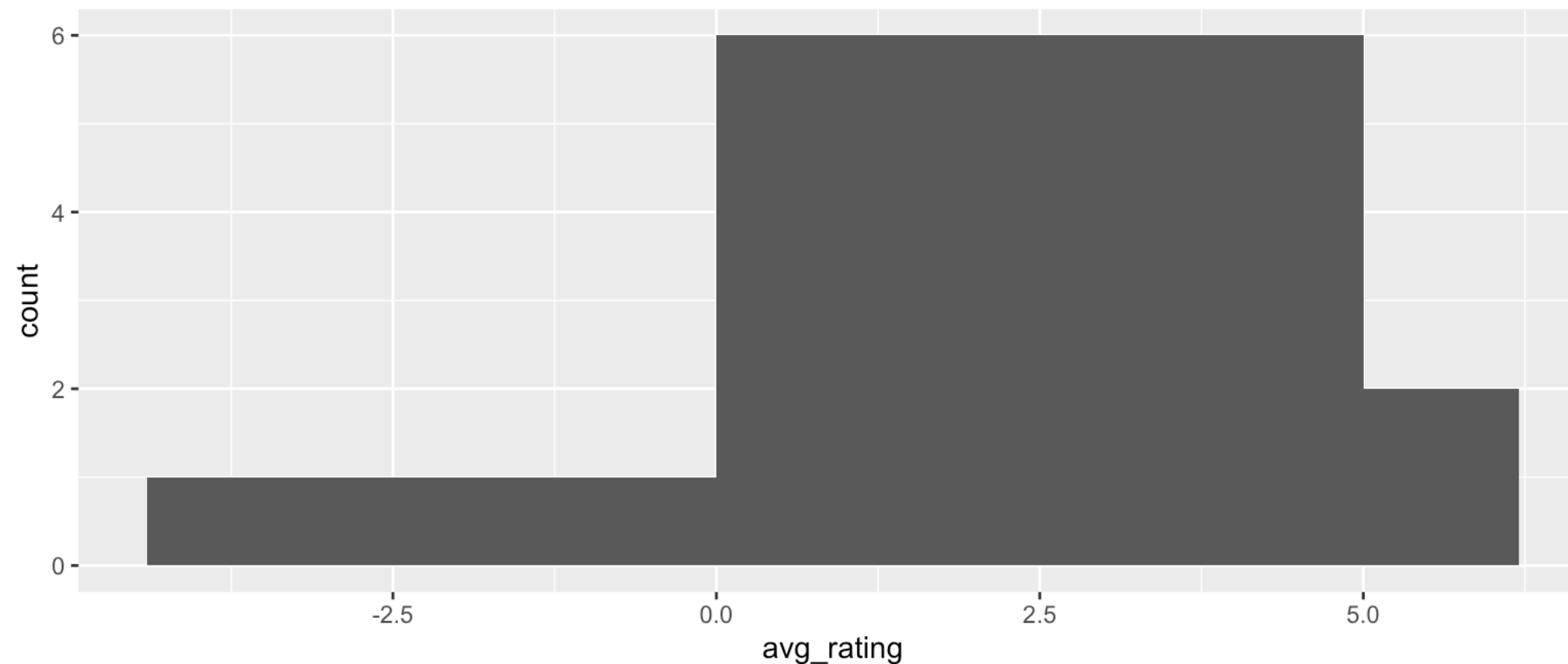
Finding out of range values

```
movies
```

```
  title          avg_rating
  <chr>          <dbl>
1 A Beautiful Mind      4.1
2 La Vita e Bella      4.3
3 Amelie               4.2
4 Meet the Parents     3.5
5 Unbreakable          5.8
6 Gone in Sixty Seconds 3.3
...
```

Finding out of range values

```
breaks <- c(min(movies$avg_rating), 0, 5, max(movies$avg_rating))  
ggplot(movies, aes(avg_rating)) +  
  geom_histogram(breaks = breaks)
```



Finding out of range values

```
library(assertive)
assert_all_are_in_closed_range(movies$avg_rating, lower = 0, upper = 5)
```

```
Error: is_in_closed_range : movies$avg_rating are not all in the range [0,5].
```

```
There were 3 failures:
```

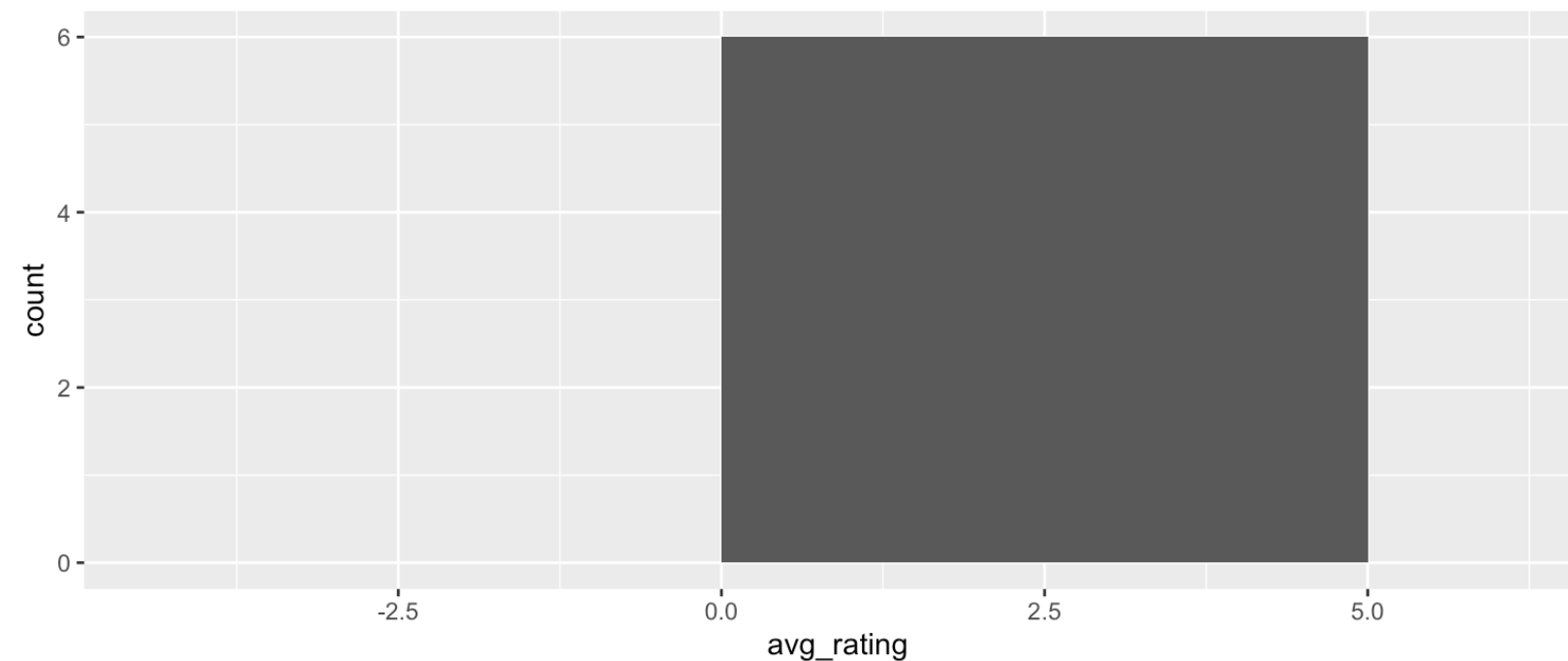
	Position	Value	Cause
1	5	5.8	too high
2	8	6.2	too high
3	9	-4.4	too low

Handling out of range values

- Remove rows
- Treat as missing (NA)
- Replace with range limit
- Replace with other value based on domain knowledge and/or knowledge of dataset

Removing rows

```
movies %>%  
  filter(avg_rating >= 0, avg_rating <= 5) %>%  
  
  ggplot(aes(avg_rating)) +  
  geom_histogram(breaks = c(min(movies$avg_rating), 0, 5, max(movies$avg_rating)))
```



Treat as missing

```
movies
```

```
  title          avg_rating
  <chr>          <dbl>
1 A Beautiful Mind      4.1
2 La Vita e Bella      4.3
3 Amelie               4.2
4 Meet the Parents     3.5
5 Unbreakable          5.8
6 Gone in Sixty Seconds 3.3
...
```

```
replace(col, condition, replacement)
```

```
movies %>%
  mutate(rating_miss =
    replace(avg_rating, avg_rating > 5, NA))
```

```
  title          rating_miss
  <chr>          <dbl>
1 A Beautiful Mind      4.1
2 La Vita e Bella      4.3
3 Amelie               4.2
4 Meet the Parents     3.5
5 Unbreakable          NA
6 Gone in Sixty Seconds 3.3
...
```

Replacing out of range values

```
heart_rate %>%  
  mutate(rating_const =  
    replace(avg_rating, avg_rating > 5, 5))
```

```
title          rating_const  
<chr>          <dbl>  
1 A Beautiful Mind      4.1  
2 La Vita e Bella      4.3  
3 Amelie              4.2  
4 Meet the Parents     3.5  
5 Unbreakable          5.0  
6 Gone in Sixty Seconds 3.3  
...
```

Date range constraints

```
assert_all_are_in_past(movies$date_recorded)
```

```
Error: is_in_past : movies$date_recorded are not all in the past.  
There was 1 failure:
```

	Position	Value	Cause
1	3	2064-09-22 20:00:00	in future

```
library(lubridate)  
heart_rate %>%  
  filter(date_recorded > today())
```

	title	avg_rating	date_recorded
1	Amelie	4.2	2064-09-23

Removing out-of-range dates

```
library(lubridate)
movies <- movies %>%
  filter(date_recorded <= today())
```

```
library(assertive)
assert_all_are_in_past(movies$date_recorded)
```

Remember, no output = passed!

Uniqueness constraints

CLEANING DATA IN R



Maggie Matsui

Content Developer @ DataCamp

What's a duplicate?

	First name	Last name	Address	Credit score
1	Miriam	Day	6042 Sollicitudin Avenue	313
2	Miriam	Day	6042 Sollicitudin Avenue	313

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit St	356
2	Tamekah	Forbes	P.O. Box 147, 511 Velit St	342

Why do duplicates occur?



**Data Entry &
Human Error**

Why do duplicates occur?



**Data Entry &
Human Error**

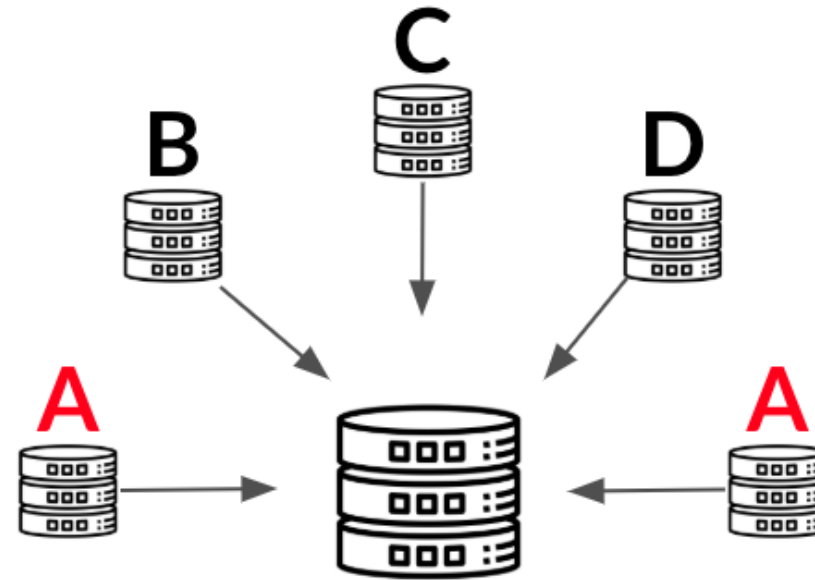


**Bugs and design
errors**

Why do duplicates occur?



Data Entry &
Human Error



Join or merge
Errors



Bugs and design
errors

Full duplicates

	First name	Last name	Address	Credit score
1	Harper	Taylor	P.O. Box 212, 6557 Nunc Road	655
2	Miriam	Day	6042 Sollicitudin Avenue	313
3	Eagan	Schmidt	507-6740 Cursus Avenue	728
4	Miriam	Day	6042 Sollicitudin Avenue	313
5	Katell	Roy	Ap #434-4081 Mi Av.	455
6	Katell	Roy	Ap #434-4081 Mi Av.	455
...

Finding full duplicates

```
 duplicated(credit_scores)
```

```
FALSE FALSE FALSE TRUE FALSE ...
```

```
sum(duplicated(credit_scores))
```

```
2
```

Finding full duplicates

```
filter(credit_scores, duplicated(credit_scores))
```

	first_name	last_name	address	credit_score
1	Miriam	Day	6042 Sollicitudin Avenue	313
2	Katell	Roy	Ap #434-4081 Mi Av.	455

Dropping full duplicates

```
credit_scores_unique <- distinct(credit_scores)
sum(duplicated(credit_scores_unique))
```

```
0
```

Partial duplicates

	First name	Last name	Address	Credit score
1	Harper	Taylor	P.O. Box 212, 6557 Nunc Road	655
2	Eagan	Schmidt	507-6740 Cursus Avenue	728
3	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
4	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342
5	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620
6	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	636
...

Finding partial duplicates

```
credit_scores %>%  
  count(first_name, last_name) %>%  
  filter(n > 1)
```

	first_name	last_name	n
	<fct>	<fct>	<int>
1	Katell	Roy	2
2	Miriam	Day	2
3	Tamekah	Forbes	2
4	Xandra	Barrett	2

Finding partial duplicates

```
dup_ids <- credit_scores %>%  
  count(first_name, last_name) %>%  
  filter(n > 1)  
credit_scores %>%  
  filter(credit_scores, first_name %in% dup_ids$first_name, last_name %in% dup_ids$last_name)
```

	first_name	last_name	address	credit_score
1	Xandra	Barrett	P.O. Box 309, 2462 Pharetra, Rd.	620
2	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
3	Miriam	Day	6042 Sollicitudin Avenue	313
4	Xandra	Barrett	P.O. Box 309, 2462 Pharetra, Rd.	636
5	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342
...				

Handling partial duplicates: dropping

Drop all duplicates except one

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
2	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620
4	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	636

Handling partial duplicates: dropping

Drop all duplicates except one

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
2				
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620
4				

Handling partial duplicates: dropping

Drop all duplicates except one

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620

Dropping partial duplicates

```
credit_scores %>%  
  distinct(first_name, last_name, .keep_all = TRUE)
```

	first_name	last_name	address	credit_score
1	Harlan	Hebert	P.O. Box 356, 3869 Non Av.	305
2	Drake	Soto	643-1409 Ac Avenue	642
3	Felix	Morales	741-1497 Velit Ave	780
4	Brynne	Charles	313-3757 Ultrices St.	513
5	Aquila	Dillon	P.O. Box 945, 5550 Aliquam Street	748
...				

Handling partial duplicates: summarizing

Summarize differing values using statistical summary functions (`mean()`, `max()`, etc.)

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356
2	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620
4	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	636

Handling partial duplicates: summarizing

Summarize differing values using statistical summary functions (`mean()`, `max()`, etc.)

	First name	Last name	Address	Credit score	Mean credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356	349
2	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342	
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620	628
4	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	636	

Handling partial duplicates: summarizing

Summarize differing values using statistical summary functions (`mean()`, `max()`, etc.)

	First name	Last name	Address	Credit score	Mean credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356	349
2					
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	620	628
4					

Handling partial duplicates: summarizing

Summarize differing values using statistical summary functions (`mean()`, `max()`, etc.)

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	349
2				
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	628
4				

Handling partial duplicates: summarizing

Summarize differing values using statistical summary functions (`mean()`, `max()`, etc.)

	First name	Last name	Address	Credit score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	349
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra Rd.	628

Summarizing partial duplicates

```
credit_scores %>%  
  group_by(first_name, last_name) %>%  
  mutate(mean_credit_score = mean(credit_score))
```

	first_name	last_name	address	credit_score	mean_score
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	356	349
2	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	342	349
3	Xandra	Barrett	P.O. Box 309, 2462 Pharetra, Rd.	636	628
4	Xandra	Barrett	P.O. Box 309, 2462 Pharetra, Rd.	620	628
5	Katell	Roy	Ap #434-4081 Mi Av.	455	455
...					

Summarizing partial duplicates

```
credit_scores %>%  
  group_by(first_name, last_name) %>%  
  mutate(mean_credit_score = mean(credit_score)) %>%  
  distinct(first_name, last_name, .keep_all = TRUE) %>%  
  select(-credit_score)
```

	first_name	last_name	address	mean_score
	<fct>	<fct>	<fct>	<dbl>
1	Tamekah	Forbes	P.O. Box 147, 511 Velit Street	349
2	Xandra	Barrett	P.O. Box 309, 2462 Pharetra, Rd.	628
3	Katell	Roy	Ap #434-4081 Mi Av.	455
4	Miriam	Day	6042 Sollicitudin Avenue	313
...				