Advanced Data Journalism: Doing More with R

Module 3: Loops

Andrew Ba Tran

Loops!

```r
x <- 5
```

```r
x <- 5

y <- 1
```

```r
x <- 5

y <- 1

x * y
```

```
[1] 5
```

```r
x <- 5

y <- 1

x * y

y <- y + 1
```

```
[1] 5
```

```
x <- 5

y <- 1

x * y

y <- y + 1

y
```

```
[1] 5

[1] 2
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y
```

```
[1] 5

[1] 2

[1] 10
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1
```

```
[1] 5

[1] 2

[1] 10
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1

y
```

```
[1] 5

[1] 2

[1] 10

[1] 3
```

```
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1

y

x * y
```

```
[1] 5

[1] 2

[1] 10

[1] 3

[1] 15
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1

y

x * y

y <- y + 1
```

```
[1] 5

[1] 2

[1] 10

[1] 3

[1] 15
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1

y

x * y

y <- y + 1

y
```

```
[1] 5

[1] 2

[1] 10

[1] 3

[1] 15

[1] 4
```

```r
x <- 5

y <- 1

x * y

y <- y + 1

y

x * y

y <- y + 1

y

x * y

y <- y + 1

y

x * y
```

```
[1] 5

[1] 2

[1] 10

[1] 3

[1] 15

[1] 4

[1] 20
```

```r
x <- 5
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}

for (y in 11:15) {
  print(str_c(x, " x ", y, " = ", x*y))
}
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75

[1] "5 x 11 = 55"
[1] "5 x 12 = 60"
[1] "5 x 13 = 65"
[1] "5 x 14 = 70"
[1] "5 x 15 = 75"
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}

for (y in 11:15) {
  print(str_c(x, " x ", y, " = ", x*y))
}

# can cycle through an array
text_array <- c("Bob", "Linda", "Tina", "Louise")
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75

[1] "5 x 11 = 55"
[1] "5 x 12 = 60"
[1] "5 x 13 = 65"
[1] "5 x 14 = 70"
[1] "5 x 15 = 75"
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}

for (y in 11:15) {
  print(str_c(x, " x ", y, " = ", x*y))
}

# can cycle through an array
text_array <- c("Bob", "Linda", "Tina", "Louise")

# get the end

length(text_array)
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75

[1] "5 x 11 = 55"
[1] "5 x 12 = 60"
[1] "5 x 13 = 65"
[1] "5 x 14 = 70"
[1] "5 x 15 = 75"

[1] 4
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}

for (y in 11:15) {
  print(str_c(x, " x ", y, " = ", x*y))
}

# can cycle through an array
text_array <- c("Bob", "Linda", "Tina", "Louise")

# get the end

length(text_array)

for (y in 1:length(text_array)) {
  print(y)
  print(text_array[y])
}
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75

[1] "5 x 11 = 55"
[1] "5 x 12 = 60"
[1] "5 x 13 = 65"
[1] "5 x 14 = 70"
[1] "5 x 15 = 75"

[1] 4

[1] 1
[1] "Bob"
[1] 2
[1] "Linda"
[1] 3
[1] "Tina"
[1] 4
[1] "Louise"
```

```r
x <- 5
for (y in 1:3) {
  print(y)
  print(x * y)
}

# start at any number
for (y in 11:15) {
  print(y)
  print(x * y)
}

for (y in 11:15) {
  print(str_c(x, " x ", y, " = ", x*y))
}

# can cycle through an array
text_array <- c("Bob", "Linda", "Tina", "Louise")

# get the end

length(text_array)

for (y in 1:length(text_array)) {
  print(y)
  print(text_array[y])
}

# doesn't have  to be y
```

```
[1] 1
[1] 5
[1] 2
[1] 10
[1] 3
[1] 15

[1] 11
[1] 55
[1] 12
[1] 60
[1] 13
[1] 65
[1] 14
[1] 70
[1] 15
[1] 75

[1] "5 x 11 = 55"
[1] "5 x 12 = 60"
[1] "5 x 13 = 65"
[1] "5 x 14 = 70"
[1] "5 x 15 = 75"

[1] 4

[1] 1
[1] "Bob"
[1] 2
[1] "Linda"
[1] 3
[1] "Tina"
[1] 4
[1] "Louise"
```

```r
df <- data.frame(x=c("a", "b", "c", "d"))
df <- data.frame(x=c("a", "b", "c", "d"))
df$y <- NA
```

```r
df <- data.frame(x=c("a", "b", "c", "d"))
df$y <- NA

df
```

```
  x  y
1 a NA
2 b NA
3 c NA
4 d NA
```

```r
df <- data.frame(x=c("a", "b", "c", "d"))
df$y <- NA

df

nrow(df)
```

```
  x  y
1 a NA
2 b NA
3 c NA
4 d NA

[1] 4
```

```r
df <- data.frame(x=c("a", "b", "c", "d"))
df$y <- NA

df

nrow(df)

for (i in 1:nrow(df)) {
  df$y[i] <- i
  print(df)
}
```

```
  x  y
1 a NA
2 b NA
3 c NA
4 d NA

[1] 4

  x  y
1 a  1
2 b NA
3 c NA
4 d NA
  x  y
1 a  1
2 b  2
3 c NA
4 d NA
  x  y
1 a  1
2 b  2
3 c  3
4 d NA
  x y
1 a 1
2 b 2
3 c 3
4 d 4
```

```r
df <- data.frame(x=c("a", "b", "c", "d"))
df$y <- NA

df

nrow(df)


for (i in 1:nrow(df)) {
  df$y[i] <- i
  print(df)
}
```

```
  x  y
1 a NA
2 b NA
3 c NA
4 d NA

[1] 4

  x  y
1 a  1
2 b NA
3 c NA
4 d NA
  x  y
1 a  1
2 b  2
3 c NA
4 d NA
  x  y
1 a  1
2 b  2
3 c  3
4 d NA
  x y
1 a 1
2 b 2
3 c 3
4 d 4
```

If Else

```
if (LOGIC TEST) { CODE GOES HERE }
```

```
if (LOGIC TEST) {
      CODE GOES HERE
      } ELSE IF (LOGIC TEST) {
      MORE CODE GOES HERE
      } ELSE {
      LAST BIT OF CODE
      }
```

```
if (LOGIC TEST) { CODE GOES HERE }

if (LOGIC TEST) {
      CODE GOES HERE
      } ELSE IF (LOGIC TEST) {
      MORE CODE GOES HERE
      } ELSE {
      LAST BIT OF CODE
      }
```

```r
for (i in 1:10) {

  if (i <= 5) {
    print(str_c(i, " is less than or equal to 5"))
  } else if (i==6) {
    print(str_c(i, " is equal to 6"))
  } else (
    print(str_c(i, " is greater than or equal to 7"))
  )

}
```

```r
for (i in 1:10) {

  if (i <= 5) {
    print(str_c(i, " is less than or equal to 5"))
  } else if (i==6) {
    print(str_c(i, " is equal to 6"))
  } else (
    print(str_c(i, " is greater than or equal to 7")
  )

}
```

```
[1] "1 is less than or equal to 5"
[1] "2 is less than or equal to 5"
[1] "3 is less than or equal to 5"
[1] "4 is less than or equal to 5"
[1] "5 is less than or equal to 5"
[1] "6 is equal to 6"
[1] "7 is greater than or equal to 7"
[1] "8 is greater than or equal to 7"
[1] "9 is greater than or equal to 7"
[1] "10 is greater than or equal to 7"
```

```r
for (i in 1:10) {

  if (i <= 5) {
    print(str_c(i, " is less than or equal to 5"))
  } else if (i==6) {
    print(str_c(i, " is equal to 6"))
  } else (
    print(str_c(i, " is greater than or equal to 7")
  )

}
```

```
[1] "1 is less than or equal to 5"
[1] "2 is less than or equal to 5"
[1] "3 is less than or equal to 5"
[1] "4 is less than or equal to 5"
[1] "5 is less than or equal to 5"
[1] "6 is equal to 6"
[1] "7 is greater than or equal to 7"
[1] "8 is greater than or equal to 7"
[1] "9 is greater than or equal to 7"
[1] "10 is greater than or equal to 7"
```

Appending data

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)
```

```
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  print(df)

}
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

   x y employee
1  1 a      Bob
2  5 b      Bob
3 10 c      Bob
4 12 d      Bob
   x y employee
1  1 a    Linda
2  5 b    Linda
3 10 c    Linda
4 12 d    Linda
   x y employee
1  1 a     Tina
2  5 b     Tina
3 10 c     Tina
4 12 d     Tina
   x y employee
1  1 a   Louise
2  5 b   Louise
3 10 c   Louise
4 12 d   Louise
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  print(df)

}

print(df)
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

   x y employee
1  1 a      Bob
2  5 b      Bob
3 10 c      Bob
4 12 d      Bob
   x y employee
1  1 a    Linda
2  5 b    Linda
3 10 c    Linda
4 12 d    Linda
   x y employee
1  1 a     Tina
2  5 b     Tina
3 10 c     Tina
4 12 d     Tina
   x y employee
1  1 a   Louise
2  5 b   Louise
3 10 c   Louise
4 12 d   Louise

   x y employee
1  1 a   Louise
2  5 b   Louise
3 10 c   Louise
4 12 d   Louise
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  print(df)

}

print(df)
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

   x y employee
1  1 a      Bob
2  5 b      Bob
3 10 c      Bob
4 12 d      Bob
   x y employee
1  1 a    Linda
2  5 b    Linda
3 10 c    Linda
4 12 d    Linda
   x y employee
1  1 a     Tina
2  5 b     Tina
3 10 c     Tina
4 12 d     Tina
   x y employee
1  1 a   Louise
2  5 b   Louise
3 10 c   Louise
4 12 d   Louise

   x y employee
1  1 a   Louise
2  5 b   Louise
3 10 c   Louise
4 12 d   Louise
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)
```

```
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
  print(i)
}
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

[1] 1
[1] 2
[1] 3
[1] 4
```

```r
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
  print(i)
}

print(big_df)
```

```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

[1] 1
[1] 2
[1] 3
[1] 4

    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise
```

```
df <- data.frame(
  x = c(1, 5, 10, 12),
  y = c("a", "b", "c", "d")
)

df

text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
  print(i)
}

print(big_df)
```

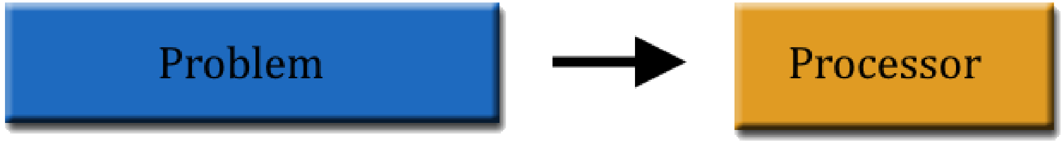
```
   x y
1  1 a
2  5 b
3 10 c
4 12 d

[1] 1
[1] 2
[1] 3
[1] 4

    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise
```

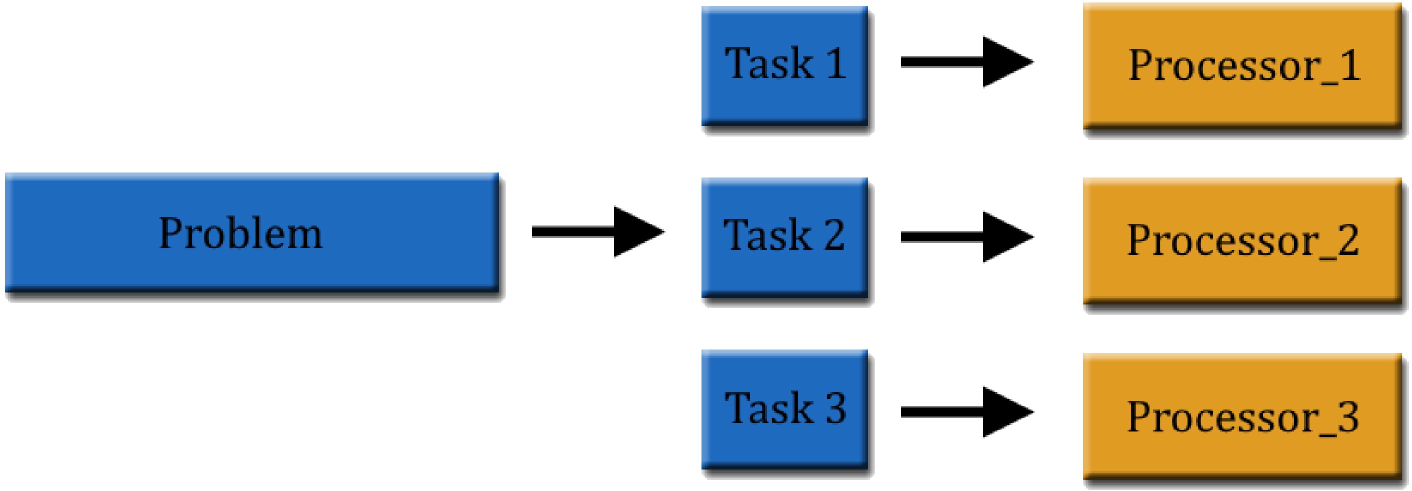
Parallel processing

Serial Computing



Problem → Processor

Parallel Computing



Problem → Task 1, Task 2, Task 3

Task 1 → Processor_1

Task 2 → Processor_2

Task 3 → Processor_3

doParallel package

```r
# if you haven't installed doParallel yet, run this first.
# Otherwise, you can comment it out by adding a # in front of the line
install.packages("doParallel")
```

```r
library(doParallel)

detectCores()
```

```
[1] 10
```

```
#### How many cores do you want to use?

registerDoParallel(cores=4)
```

```r
text_array <- c("Bob", "Linda", "Tina", "Louise")
text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
}
```

```r
text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
}

print(big_df)
```

```
    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise
```

```
text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
}

print(big_df)

# the parallel version
big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
    mutate(employee=text_array[i])
}
```

```
    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise
```

```r
text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
}

print(big_df)

# the parallel version
big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
    mutate(employee=text_array[i])
}

print(big_df)
```

```
    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise

    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
```

```r
text_array <- c("Bob", "Linda", "Tina", "Louise")

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

  df <- df %>%
    mutate(employee=text_array[i])

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }
}

print(big_df)

# the parallel version
big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
    mutate(employee=text_array[i])
}

print(big_df)
```

```
    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
15 10 c   Louise
16 12 d   Louise

    x y employee
1   1 a      Bob
2   5 b      Bob
3  10 c      Bob
4  12 d      Bob
5   1 a    Linda
6   5 b    Linda
7  10 c    Linda
8  12 d    Linda
9   1 a     Tina
10  5 b     Tina
11 10 c     Tina
12 12 d     Tina
13  1 a   Louise
14  5 b   Louise
```

```r
system.time(
  for (i in 1:length(text_array)) {

    df <- df %>%
      mutate(employee=text_array[i])

    if (i==1) {
      big_df <- df
    } else {
      big_df <- rbind(big_df, df)
    }

  Sys.sleep(5)
  }
)
```

```
 user  system elapsed 
0.005   0.000  20.023 
```

```r
system.time(
  for (i in 1:length(text_array)) {

    df <- df %>%
      mutate(employee=text_array[i])

    if (i==1) {
      big_df <- df
    } else {
      big_df <- rbind(big_df, df)
    }

  Sys.sleep(5)
  }
)

system.time(
  big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
    mutate(employee=text_array[i])
    Sys.sleep(5)
  }
)
```

```
 user  system elapsed
0.006   0.000  20.021

 user  system elapsed
0.008   0.029   5.016
```

```
system.time(
  for (i in 1:length(text_array)) {

    df <- df %>%
      mutate(employee=text_array[i])

    if (i==1) {
      big_df <- df
    } else {
      big_df <- rbind(big_df, df)
    }

  Sys.sleep(5)
  }
)


system.time(
  big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
    mutate(employee=text_array[i])
    Sys.sleep(5)
  }
)
```

```
  user  system elapsed 
 0.007   0.000  20.017 
```

```
  user  system elapsed 
 0.008   0.024   5.017 
```

```
system.time(
  for (i in 1:length(text_array)) {

    df <- df %>%
      mutate(employee=text_array[i])

    if (i==1) {
      big_df <- df
    } else {
      big_df <- rbind(big_df, df)
    }

  Sys.sleep(5)
  }
)
```

```
  user  system elapsed 
 0.010   0.000  20.026 
```

```
system.time(
  big_df <- foreach(i=1:length(text_array), .combine=rbind) %dopar% {
    df %>%
install.packages("usethis")

usethis::use_course("https://github.com/r-journalism/adj-mod-3/archive/master.zip")
```

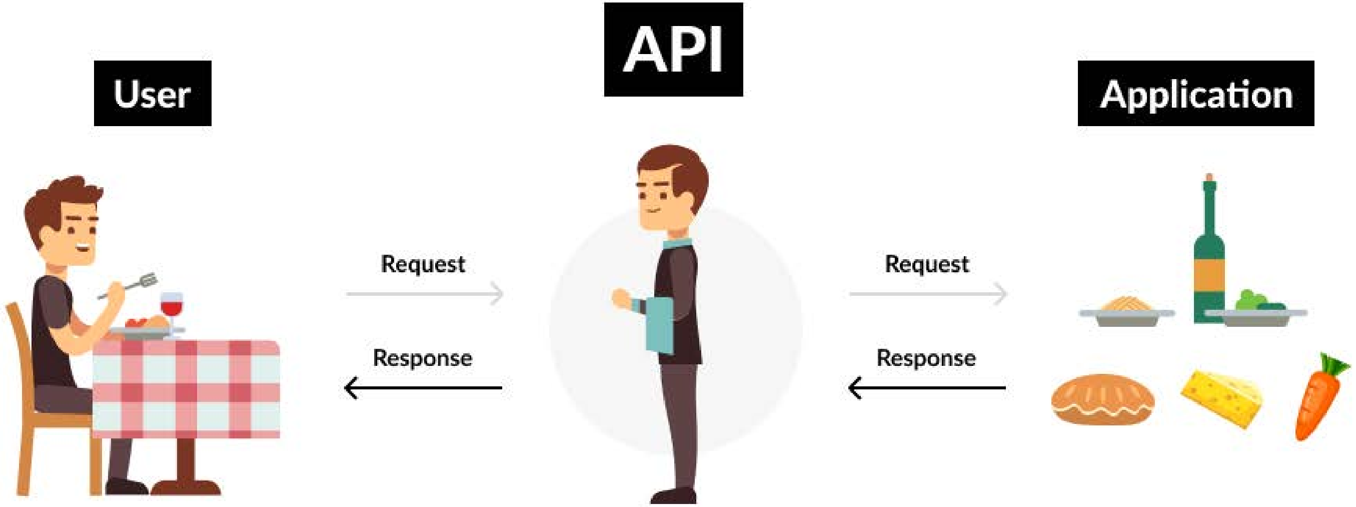
Advanced Data Journalism: Doing More with R

Module 3: APIs

Andrew Ba Tran

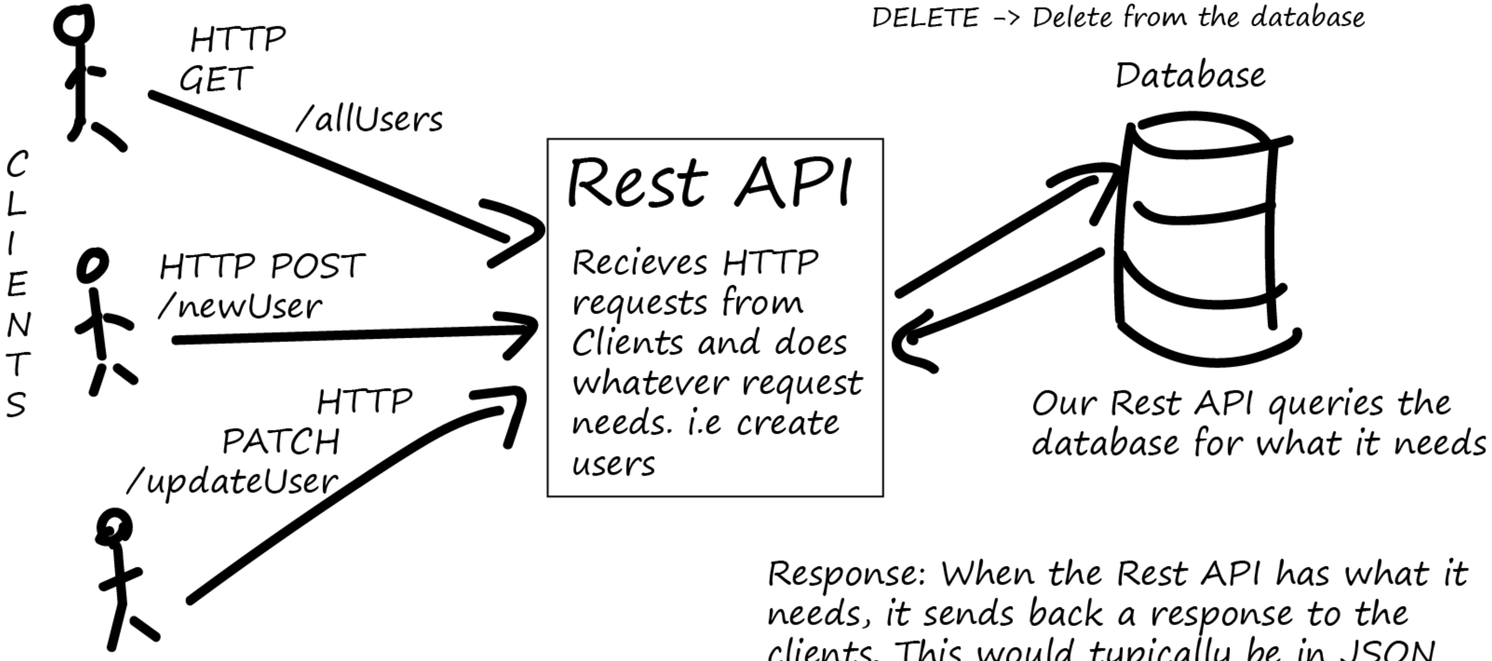
APIS

Application Programming Interface


API
User
Application
Request
Response
Request
Response

Rest API Basics

Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database



CLIENTS

HTTP GET /allUsers

HTTP POST /newUser

HTTP PATCH /updateUser

Rest API

Recieves HTTP requests from Clients and does whatever request needs. i.e create users

Database

Our Rest API queries the database for what it needs

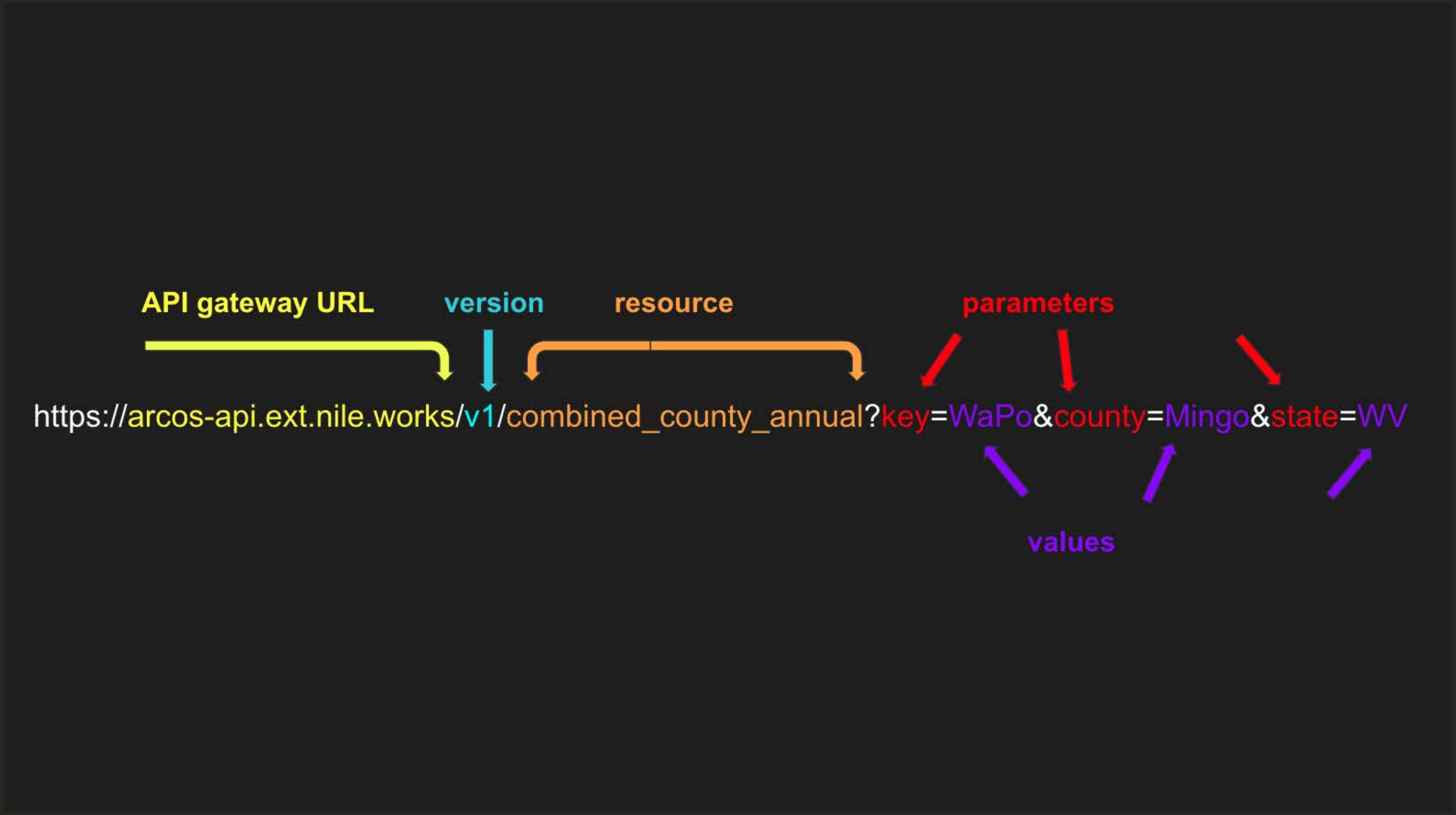
Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

Look for documentation

https://arcos-api.ext.nile.works/__docs__/

```
https://arcos-api.ext.nile.works/v1/combined_county_annual?
key=WaPo&county=Mingo&state=WV
```


API gateway URL
version
resource
parameters
https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&county=Mingo&state=WV
values

Importing JSON files

fromJSON() from jsonlite

```r
library(jsonlite)
library(jsonlite)

url <- "https://arcos-api.ext.nile.works/v1/combined
```

```r
library(jsonlite)

url <- "https://arcos-api.ext.nile.works/v1/combined

url
```

```
[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&county=Mingo&state=
```

```r
library(jsonlite)

url <- "https://arcos-api.ext.nile.works/v1/combined

url

fromJSON(url)
```

```
[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&county=Mingo&state=

  BUYER_COUNTY BUYER_STATE year count DOSAGE_UNIT countyfips
1        MINGO          WV 2006  4192     4691980      54059
2        MINGO          WV 2007  5389     7511830      54059
3        MINGO          WV 2008  7496     9795100      54059
4        MINGO          WV 2009  7227     7464630      54059
5        MINGO          WV 2010  4466     3162950      54059
6        MINGO          WV 2011  4243     2863950      54059
7        MINGO          WV 2012  4141     2779190      54059
8        MINGO          WV 2013  4189     2562840      54059
9        MINGO          WV 2014  3784     2414610      54059
```

```r
library(jsonlite)

url <- "https://arcos-api.ext.nile.works/v1/combined

url

fromJSON(url)
```

```
[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&county=Mingo&state=

  BUYER_COUNTY BUYER_STATE year count DOSAGE_UNIT countyfips
1        MINGO          WV 2006  4192     4691980      54059
2        MINGO          WV 2007  5389     7511830      54059
3        MINGO          WV 2008  7496     9795100      54059
4        MINGO          WV 2009  7227     7464630      54059
5        MINGO          WV 2010  4466     3162950      54059
6        MINGO          WV 2011  4243     2863950      54059
7        MINGO          WV 2012  4141     2779190      54059
8        MINGO          WV 2013  4189     2562840      54059
9        MINGO          WV 2014  3784     2414610      54059
```

loops + API json

```
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
```

```
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
```

```
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url
```

```
[1] "county=Mingo"
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
```

```
[1] "county=Mingo"
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
```

```
[1] "county=Mingo"
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
state_url
```

```
[1] "county=Mingo"

[1] "state=WV"
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
state_url

new_url <-str_c(base_url, key, "&", county_url, "&", state_url)
```

```
[1] "county=Mingo"

[1] "state=WV"
```

```
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
state_url

new_url <-str_c(base_url, key, "&", county_url, "&", state_url)
new_url
```

```
[1] "county=Mingo"

[1] "state=WV"

[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&count
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
state_url

new_url <-str_c(base_url, key, "&", county_url, "&", state_url)
new_url

fromJSON(new_url)
```

```
[1] "county=Mingo"

[1] "state=WV"

[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&count

  BUYER_COUNTY BUYER_STATE year count DOSAGE_UNIT countyfips
1        MINGO          WV 2006  4192     4691980      54059
2        MINGO          WV 2007  5389     7511830      54059
3        MINGO          WV 2008  7496     9795100      54059
4        MINGO          WV 2009  7227     7464630      54059
5        MINGO          WV 2010  4466     3162950      54059
6        MINGO          WV 2011  4243     2863950      54059
7        MINGO          WV 2012  4141     2779190      54059
8        MINGO          WV 2013  4189     2562840      54059
9        MINGO          WV 2014  3784     2414610      54059
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
county <- "Mingo"
county_url <- str_c("county=", county)
county_url

state <- "WV"
state_url <- str_c("state=", state)
state_url

new_url <-str_c(base_url, key, "&", county_url, "&", state_url)
new_url

fromJSON(new_url)
```

```
[1] "county=Mingo"

[1] "state=WV"

[1] "https://arcos-api.ext.nile.works/v1/combined_county_annual?key=WaPo&count

  BUYER_COUNTY BUYER_STATE year count DOSAGE_UNIT countyfips
1        MINGO          WV 2006  4192     4691980      54059
2        MINGO          WV 2007  5389     7511830      54059
3        MINGO          WV 2008  7496     9795100      54059
4        MINGO          WV 2009  7227     7464630      54059
5        MINGO          WV 2010  4466     3162950      54059
6        MINGO          WV 2011  4243     2863950      54059
7        MINGO          WV 2012  4141     2779190      54059
8        MINGO          WV 2013  4189     2562840      54059
9        MINGO          WV 2014  3784     2414610      54059
```

```
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"

counties <- c("Harris", "Tarrant", "Dallas")
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"

counties <- c("Harris", "Tarrant", "Dallas")

for (i in 1:length(counties)) {
  county <- counties[i]
  county_url <- str_c("county=", county)

  state <- "TX"
  state_url <- str_c("state=", state)
  new_url <-str_c(base_url, key, "&", county_url, "&", state_url)

  df <- fromJSON(new_url)

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }

}
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"

counties <- c("Harris", "Tarrant", "Dallas")

for (i in 1:length(counties)) {
  county <- counties[i]
  county_url <- str_c("county=", county)

  state <- "TX"
  state_url <- str_c("state=", state)
  new_url <-str_c(base_url, key, "&", county_url, "&", state_url)

  df <- fromJSON(new_url)

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }

}

big_df
```

```
   BUYER_COUNTY BUYER_STATE year  count DOSAGE_UNIT countyfips
1        HARRIS          TX 2006 206680   145690196      48201
2        HARRIS          TX 2007 243222   183003763      48201
3        HARRIS          TX 2008 250315   136983464      48201
4        HARRIS          TX 2009 264200   156467571      48201
5        HARRIS          TX 2010 276285   160142484      48201
6        HARRIS          TX 2011 285296   151801441      48201
7        HARRIS          TX 2012 266336   126963737      48201
8        HARRIS          TX 2013 263131   116538024      48201
9        HARRIS          TX 2014 187644   105184824      48201
10      TARRANT          TX 2006 125137    50777151      48439
11      TARRANT          TX 2007 134830    55849300      48439
12      TARRANT          TX 2008 142360    58077064      48439
13      TARRANT          TX 2009 143047    59735736      48439
14      TARRANT          TX 2010 146837    63097890      48439
15      TARRANT          TX 2011 150230    71161004      48439
16      TARRANT          TX 2012 143256    72627160      48439
17      TARRANT          TX 2013 139980    72562334      48439
18      TARRANT          TX 2014 112501    67604150      48439
19       DALLAS          TX 2006 140376    51676450      48113
20       DALLAS          TX 2007 154585    57535470      48113
21       DALLAS          TX 2008 162708    61494845      48113
22       DALLAS          TX 2009 162333    65300300      48113
23       DALLAS          TX 2010 165299    70972703      48113
24       DALLAS          TX 2011 168427    79077645      48113
25       DALLAS          TX 2012 154628    75089714      48113
26       DALLAS          TX 2013 150723    75276296      48113
27       DALLAS          TX 2014 114649    70160960      48113
```

```r
base_url <- "https://arcos-api.ext.nile.works/v1/combined_county_a
key <- "key=WaPo"

counties <- c("Harris", "Tarrant", "Dallas")

for (i in 1:length(counties)) {
  county <- counties[i]
  county_url <- str_c("county=", county)

  state <- "TX"
  state_url <- str_c("state=", state)
  new_url <-str_c(base_url, key, "&", county_url, "&", state_url)

  df <- fromJSON(new_url)

  if (i==1) {
    big_df <- df
  } else {
    big_df <- rbind(big_df, df)
  }

}

big_df
```

```
   BUYER_COUNTY BUYER_STATE year  count DOSAGE_UNIT countyfips
1        HARRIS          TX 2006 206680   145690196      48201
2        HARRIS          TX 2007 243222   183003763      48201
3        HARRIS          TX 2008 250315   136983464      48201
4        HARRIS          TX 2009 264200   156467571      48201
5        HARRIS          TX 2010 276285   160142484      48201
6        HARRIS          TX 2011 285296   151801441      48201
7        HARRIS          TX 2012 266336   126963737      48201
8        HARRIS          TX 2013 263131   116538024      48201
9        HARRIS          TX 2014 187644   105184824      48201
10      TARRANT          TX 2006 125137    50777151      48439
11      TARRANT          TX 2007 134830    55849300      48439
12      TARRANT          TX 2008 142360    58077064      48439
13      TARRANT          TX 2009 143047    59735736      48439
14      TARRANT          TX 2010 146837    63097890      48439
15      TARRANT          TX 2011 150230    71161004      48439
16      TARRANT          TX 2012 143256    72627160      48439
17      TARRANT          TX 2013 139980    72562334      48439
18      TARRANT          TX 2014 112501    67604150      48439
19       DALLAS          TX 2006 140376    51676450      48113
20       DALLAS          TX 2007 154585    57535470      48113
21       DALLAS          TX 2008 162708    61494845      48113
22       DALLAS          TX 2009 162333    65300300      48113
23       DALLAS          TX 2010 165299    70972703      48113
24       DALLAS          TX 2011 168427    79077645      48113
25       DALLAS          TX 2012 154628    75089714      48113
26       DALLAS          TX 2013 150723    75276296      48113
27       DALLAS          TX 2014 114649    70160960      48113
```