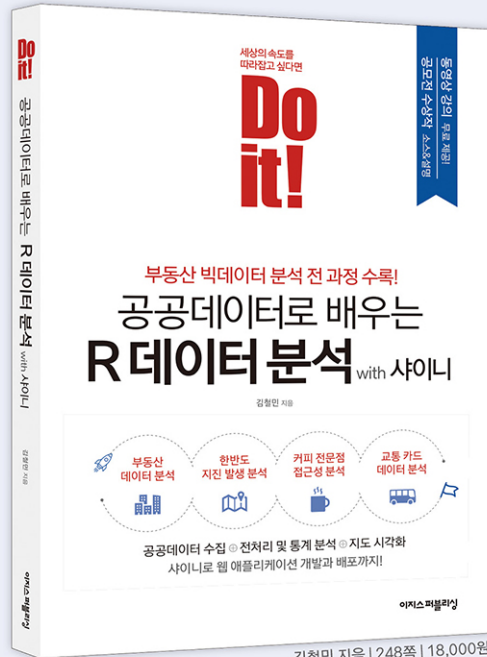


R로 공공데이터를 분석하는 전 과정 실습!
공모전 수상작으로 배우는 R 데이터 분석



공모전
수상작
소스 & 설명

동영상
강의
무료 제공

김철민 지음 | 248쪽 | 18,000원

웹 기반 애플리케이션은 데이터의 특성을 직관적으로 살펴볼 수 있는 좋은 도구입니다. 한 화면에 데이터가 가진 연결성을 직관적으로 파악할 수 있도록 도와줌으로써 중요한 의사결정 정보를 한눈에 파악할 수 있습니다.

12

샤이니 애플리케이션 활용 사례

12-1 아파트 가격 상관관계 분석하기

12-2 여러 지역 상관관계 비교하기

12-1 아파트 가격 상관관계 분석하기

1단계 데이터 준비하기

이번 절에서는 지금까지 사용한 2021 아파트 실거래 데이터를 활용하여 아파트 크기, 층수, 건축 연도, 가격이라는 4가지 요소가 지역별로 어떠한 관계를 보이는지 살펴보겠습니다.

Do it! 데이터 준비하기

12-1_아파트가격_상관관계분석.R

```
08: setwd(dirname(rstudioapi::getSourceEditorContext())$path)) # 작업 폴더 설정
09: load("./06_geodataframe/06_apr_price.rdata") # 실거래 데이터 불러오기
10: library(sf)
11: apt_price <- st_drop_geometry(apt_price) # 공간 정보 제거
12: apt_price$py_area <- round(apt_price$area / 3.3, 0) # 크기 변환 (㎡ -> 평)
13: head(apt_price$py_area)
```

실행 결과

```
[1] 39 44 53 53 37 37
```

12-1 아파트 가격 상관관계 분석하기

2단계 사용자 화면 구현하기

Do it! 사용자 화면 시작과 제목 패널 구현 12-1_아파트가격_상관관계분석.R

```
18: library(shiny)
19: ui <- fluidPage(
20:   #---# 제목 설정
21:   titlePanel("아파트 가격 상관관계 분석"),
```

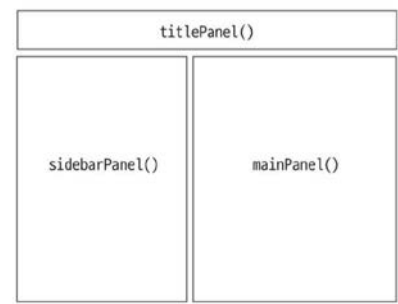


그림 12-2 사용자 화면 레이아웃

Do it! 사이드바 패널 시작과 지역 선택 메뉴 구현 12-1_아파트가격_상관관계분석.R

```
22: #---# 사이드 패널
23: sidebarPanel(
24:   #---# 선택 메뉴 1: 지역
25:   selectInput(
26:     inputId = "sel_gu",          # 입력 아이디
27:     label = "지역을 선택하세요",  # 라벨
28:     choices = unique(apt_price$addr_1), # 지역 리스트
29:     selected = unique(apt_price$addr_1)[1]), # 기본 지역 선택
```



그림 12-3 사용자 화면 입출력 위젯 구성

12-1 아파트 가격 상관관계 분석하기

2단계 사용자 화면 구현하기



그림 12-3 사용자 화면 입출력 위젯 구성

```
30: #---# 선택 메뉴 2: 크기(평)
31: sliderInput(
32:   inputId = "range_py", # 입력 아이디
33:   label = "평수",       # 라벨
34:   min = 0,              # 선택 범위 최솟값
35:   max = max(apt_price$py_area), # 선택 범위 최댓값
36:   value = c(0, 30)),    # 기본 선택 범위
37: #---# 선택 메뉴 3: X축 변수
38: selectInput(
39:   inputId = "var_x",     # 입력 아이디
40:   label = "X축 변수를 선택하십시오", # 라벨
41:   choices = list(        # 선택 리스트
42:     "매매가(평당)"="py",
43:     "크기(평)"="py_area",
44:     "건축 연도"="con_year",
45:     "층수"="floor"),
46:   selected = "py_area"), # 기본 선택
47: #---# 선택 메뉴 4: Y축 변수
48: selectInput(
49:   inputId = "var_y",     # 입력 아이디
50:   label = "Y축 변수를 선택하십시오", # 라벨
51:   choices = list(        # 선택 리스트
52:     "매매가(평당)"="py",
53:     "크기(평)"="py_area",
54:     "건축 연도"="con_year",
55:     "층수"="floor"),
56:   selected = "py"),      # 기본 선택
```

12-1 아파트 가격 상관관계 분석하기

2단계 사용자 화면 구현하기

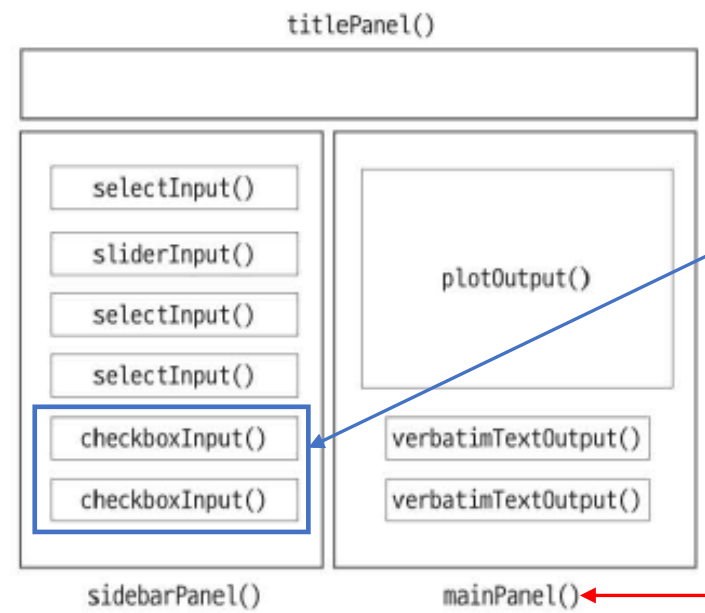


그림 12-3 사용자 화면 입출력 위젯 구성

```
57: #---# 체크박스 1: 상관 계수
58: checkboxInput(
59:   inputId = "corr_checked",      # 입력 아이디
60:   label = strong("상관 계수"),   # 라벨
61:   value = TRUE),                # 기본 선택
62: #---# 체크박스 2: 회귀 계수
63: checkboxInput(
64:   inputId = "reg_checked",       # 입력 아이디
65:   label = strong("회귀 계수"),   # 라벨
66:   value = TRUE)                  # 기본 선택
67: ),
```

```
68: #---# 메인 패널
69: mainPanel(
70:   #---#
71:   h4("플로팅"),                  # 라벨
72:   plotOutput("scatterPlot"),     # 플롯 출력
73:   #---#
74:   h4("상관 계수"),               # 라벨
75:   verbatimTextOutput("corr_coef"), # 텍스트 출력
76:   #---#
77:   h4("회귀 계수"),               # 라벨
78:   verbatimTextOutput("reg_fit")   # 텍스트 출력
79: )
80: )
```


12-1 아파트 가격 상관관계 분석하기

3단계 서버 구현하기

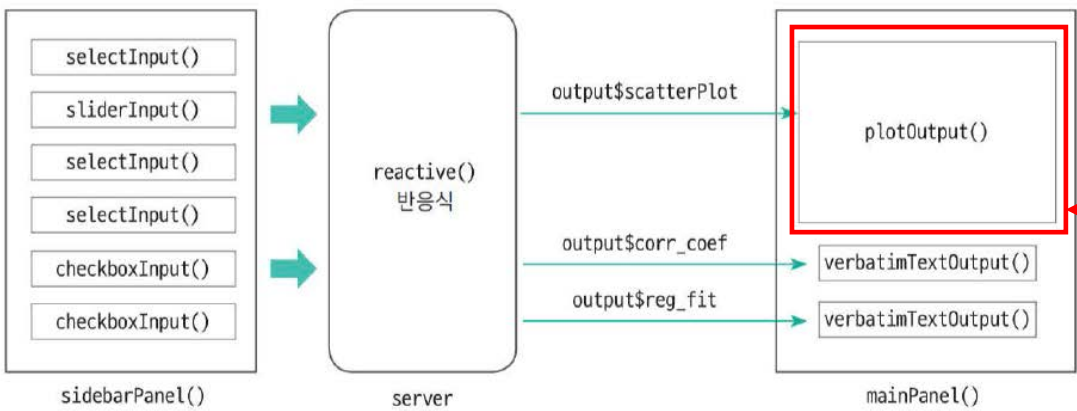


그림 12-4 데이터 전달 과정

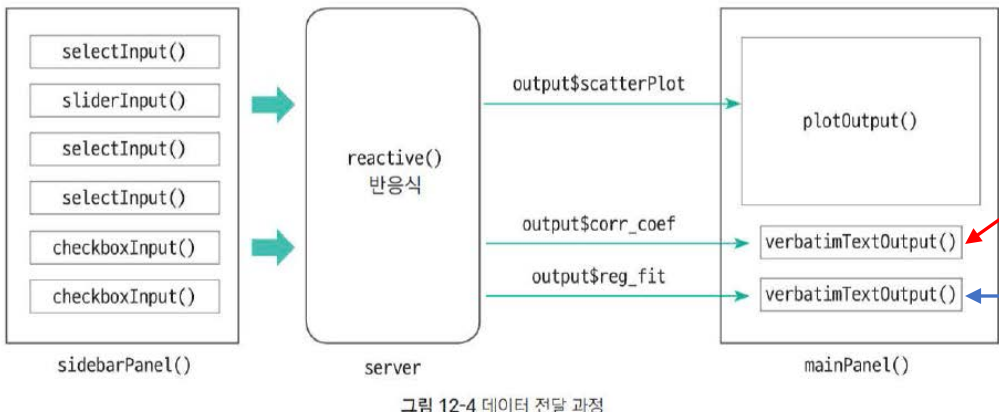
Do it! 서버 구현

12-1_아파트가격_상관관계분석.R

```
84: server <- function(input, output, session) {
85:   #---# 반응식
86:   apt_sel = reactive({
87:     apt_sel = subset(apt_price,
88:       addr_1 == input$sel_gu &          # 지역 선택
89:       py_area >= input$range_py[1] &    # 최소 크기(평) 선택
90:       py_area <= input$range_py[2])    # 최대 크기(평) 선택
91:   return(apt_sel)
92: })
93: #---# 플롯
94: output$scatterPlot <- renderPlot({
95:   var_name_x <- as.character(input$var_x) # X축 이름
96:   var_name_y <- as.character(input$var_y) # Y축 이름
97:   #---# 회귀선 그리기
98:   plot(
99:     apt_sel()[, input$var_x], # X축 설정
100:    apt_sel()[, input$var_y], # Y축 설정
101:    xlab = var_name_x, # X축 라벨
102:    ylab = var_name_y, # Y축 라벨
103:    main = "플로팅") # 플롯 제목
104:    fit <- lm(apt_sel()[, input$var_y] ~ apt_sel()[, input$var_x]) # 회귀식
105:    abline(fit, col="red") # 회귀선 그리기
106:  })
```

12-1 아파트 가격 상관관계 분석하기

3단계 서버 구현하기



```
107: #---# 상관 계수
108: output$corr_coef <- renderText({
109:   if(input$corr_checked){           # 체크박스 1 확인
110:     cor(apt_sel()[, input$var_x],   # 상관 분석 X축 설정
111:         apt_sel()[, input$var_y])   # 상관 분석 Y축 설정
112:   }
113: })
114: #---# 회귀 계수
115: output$reg_fit <- renderPrint({
116:   if(input$reg_checked){           # 체크박스 2 확인
117:     fit <- lm(apt_sel()[, input$var_y] ~ apt_sel()[, input$var_x]) # 회귀 분석
118:     names(fit$coefficients) <- c("Intercept", input$var_x) # 회귀 계수 추출
119:     summary(fit)$coefficients      # 회귀 계수 요약
120:   }
121: })
122: }
```

12-1 아파트 가격 상관관계 분석하기

4단계 애플리케이션 실행하기

Do it! 애플리케이션 실행

12-1_아파트가격_상관관계분석.R

```
126: shinyApp(ui = ui, server = server)
```

실행 결과

아파트 가격 상관관계 분석

지역을 선택하세요

서울_종로

평수

0

30

99

X축 변수를 선택하십시오

크기(평)

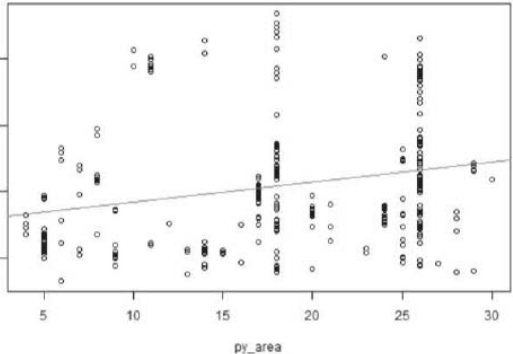
Y축 변수를 선택하십시오

매매가(평당)

☒ 상관계수

☒ 회귀계수

플로팅



상관 계수

0.2511747

회귀 계수

	Estimate	Std. Error	t value	Pr(> t)
Intercept	3097.45963	235.10466	13.17477	1.362505e-32
py_area	59.37749	12.09355	4.90985	1.387948e-06

12-2 여러 지역 상관관계 비교하기

1단계 데이터 준비하기

Do it! 데이터 준비하기

12-2_여러지역_상관관계비교.R

```
08: setwd(dirname(rstudioapi::getSourceEditorContext()$path)) # 작업 폴더 설정
09: load("./06_geodataframe/06_apr_price.rdata") # 실거래 데이터 불러오기
10: library(sf)
11: apt_price <- st_drop_geometry(apt_price) # 공간 정보 제거
12: apt_price$py_area <- round(apt_price$area / 3.3, 0) # 크기 변환 (㎡ -> 평)
13: head(apt_price$py_area)
```

실행 결과

```
[1] 39 44 53 53 37 37
```

12-2 여러 지역 상관관계 비교하기

2단계 사용자 화면 구현하기

앞 절에서는 패널을 이용하여 사용자 화면을 분할했지만, 여기서는 `fluidRow()`를 사용하여 칼럼 `column` 단위로 분할합니다. 이때 한 칼럼의 최대 길이는 12이며 이를 넘어서면 다음 행으로 넘기도록 구현합니다.

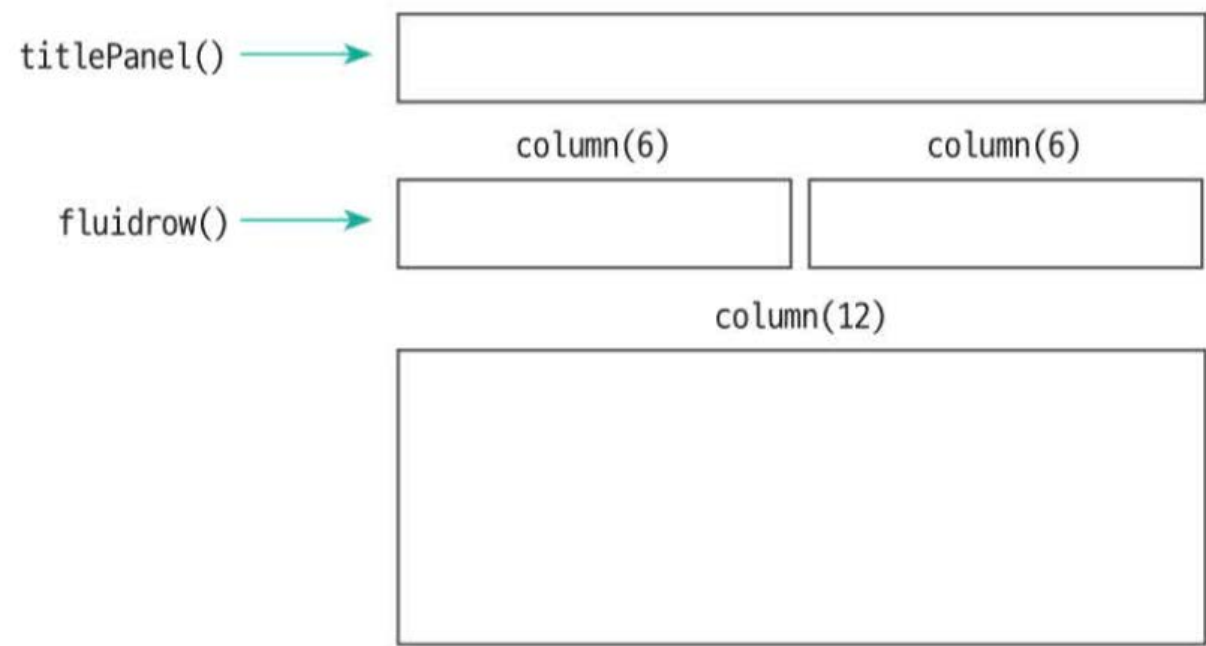


그림 12-6 사용자 화면 레이아웃

2단계 사용자 화면 구현하기

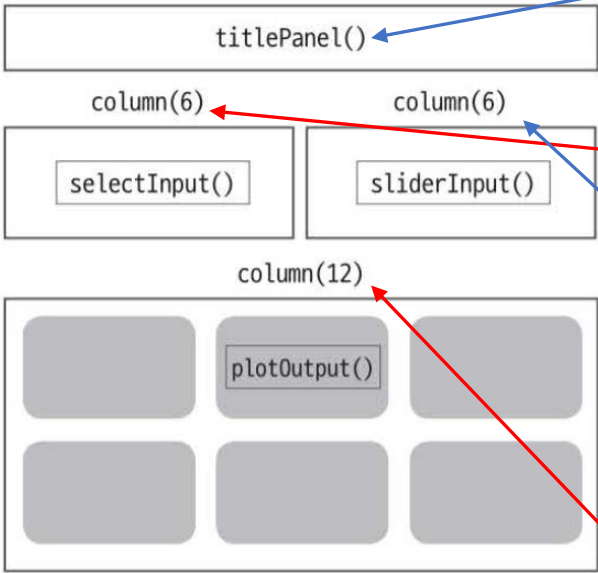


그림 12-7 사용자 화면 입출력 위젯 구성

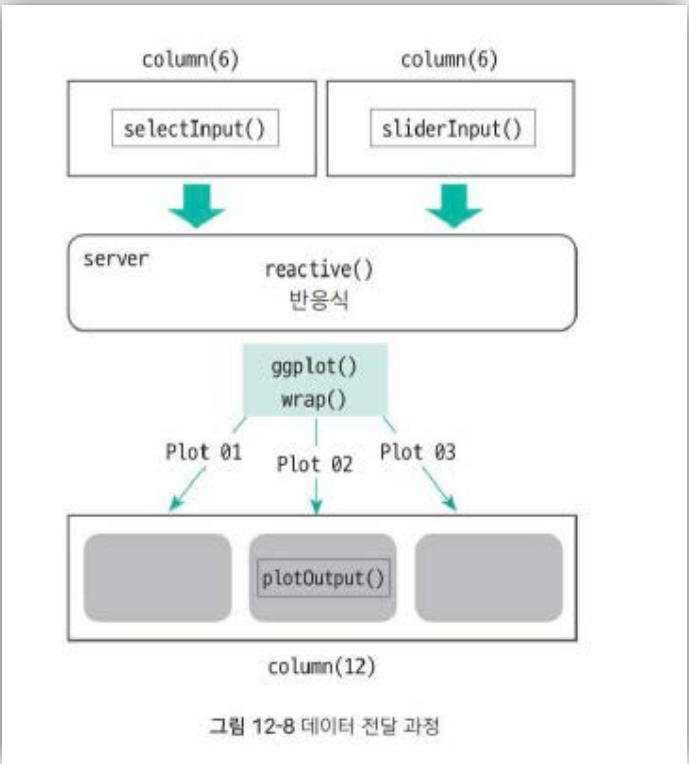
Do it! 사용자 화면 시작과 제목 패널 구현12-2_여러지역_상관관계비교.R

```
17: library(shiny) # install.packages("shiny")
18: library(ggpmisc) # install.packages("ggpmisc")
19:
20: ui <- fluidPage(
21:   #---# 타이틀 입력
22:   titlePanel("여러 지역 상관관계 비교"),
```

Do it! 입출력 화면12-2_여러지역_상관관계비교.R

```
24: fluidRow(
25:   #---# 선택 메뉴 1: 지역 선택
26:   column(6,
27:     selectInput(
28:       inputId = "region", # 입력 아이디
29:       label = "지역을 선택하세요", # 라벨
30:       unique(apt_price$addr_1), # 선택 범위
31:       multiple = TRUE)), # 복수 선택 옵션
32:   #---# 선택 메뉴 2: 크기 선택
33:   column(6,
34:     sliderInput(
35:       inputId = "range_py", # 입력 아이디
36:       label = "평수를 선택하세요", # 라벨
37:       min = 0, # 선택 범위 최솟값
38:       max = max(apt_price$py_area), # 선택 범위 최댓값
39:       value = c(0, 30))), # 기본 선택 범위
40:   #---# 출력
41:   column(12,
42:     plotOutput(outputId = "gu_Plot", height="600"))) # 차트 출력
43: )
```

3단계 서버 구현하기



이때 `facet_wrap()`과 `stat_poly_eq()` 함수에 주목해야 합니다. `facet_wrap()`은 특정 변수를 카테고리 변수로 만들어 줍니다. 따라서 `facet_wrap(~addr_1)`처럼 하면 똑같은 조건으로 여러 지역을 비교할 수 있습니다. 이때 Y축 값은 각 데이터의 특성에 따라 달라질 수 있으므로 상황에 따라 유연한 그래프를 그리고자 `scale='free_y'` 옵션을 사용합니다. 또한 `ncol=3`으로 한 행에 3개씩 그리도록 설정합니다.

```
47: server <- function(input, output, session){
48:   #---# 반응식
49:   apt_sel = reactive({
50:     apt_sel = subset(apt_price,
51:       addr_1 == unlist(strsplit(paste(input$region, collapse = ','),",")) &
52:       py_area >= input$range_py[1] &
53:       py_area <= input$range_py[2])
54:     return(apt_sel)
55:   })
56:   #---# 지역별 회귀선 그리기
57:   output$gu_Plot <- renderPlot({
58:     if (nrow(apt_sel()) == 0) # 선택 전 오류 메시지 없애기
59:       return(NULL)
60:     ggplot(apt_sel(), aes(x = py_area, y = py, col="red")) + # 축 설정
61:       geom_point() + # 플롯 유형: 포인트
62:       geom_smooth(method="lm", col="blue") + # 회귀선
63:       facet_wrap(~addr_1, scale='free_y', ncol=3) + # 테마 설정
64:       theme(legend.position="none") + # 카테고리별 그리기
65:       xlab('크기(평)') + # X축 설정
66:       ylab('평당 가격(만원)') + # Y축 설정
67:       stat_poly_eq(aes(label = paste(..eq.label..),
68:         label.x = "right", label.y = "top",
69:         formula = y ~ x, parse = TRUE, size = 5, col="black"))
70:   })
71: }
```

회귀식 설정

4단계 애플리케이션 실행하기

애플리케이션이 실행되면 서울시 25개 자치구 가운데 최소 두 곳을 선택합니다. 그리고 살펴 보려는 지역의 아파트 평수를 설정합니다. 그러면 지역별로 아파트 크기의 변화에 따라서 평 당 매매가가 어떻게 달라지는지 나타냅니다. 예를 들어 위 실행 결과에서는 서울시 광진구, 종로구의 30평 이하 아파트는 크기가 클수록 평당 가격이 상승하지만 강북구는 오히려 떨어 지는 모습을 보입니다.

Do it! 애플리케이션 실행

12-2_여러지역_상관관계비교.R

74: shinyApp(ui = ui, server = server)

실행 결과

여러 지역 상관관계 비교

지역을 선택하세요

서울_광진

서울_강북

서울_종로

서울_용산

서울_양천

서울_강서

평수를 선택하세요

0

30

95

서울_강북

$y = 3.52 \times 10^{-3} - 21.2 x$

서울_강서

$y = 3.87 \times 10^{-3} - 0.518 x$

서울_광진

$y = 3.63 \times 10^{-3} + 65.2 x$

서울_양천

$y = 4.99 \times 10^{-3} - 24.7 x$

서울_용산

$y = 6.12 \times 10^{-3} + 2.99 x$

서울_종로

$y = 2.68 \times 10^{-3} + 70.5 x$