Polynomial Regression

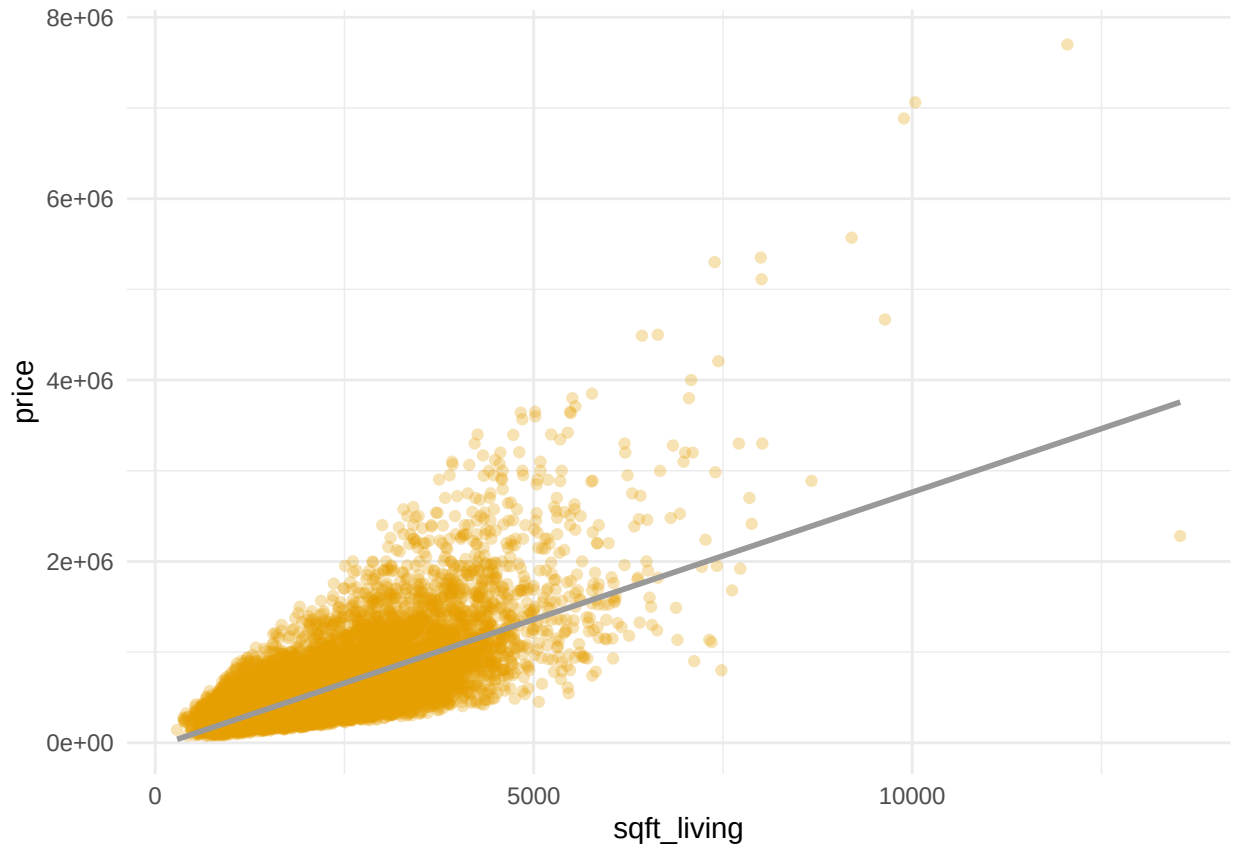
Guan-Yuan Wang

2020/9/7

```
data <- read.csv("kc_house_data.csv")
```

```
data %>%
  ggplot(aes(sqft_living, price)) +
  geom_point(color = "#E69F00", alpha = 0.3) +
  geom_smooth(method = lm, se = FALSE, color = "#999999")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
set1 <- read.csv("wk3_kc_house_set_1_data.csv")
set2 <- read.csv("wk3_kc_house_set_2_data.csv")
set3 <- read.csv("wk3_kc_house_set_3_data.csv")
```

```r
set4 <- read.csv("wk3_kc_house_set_4_data.csv")

lm1 <- lm(price ~ bs(sqft_living, 15), set1)
lm1$coefficients
```

```
##           (Intercept)  bs(sqft_living, 15)1  bs(sqft_living, 15)2
##              165459.9               89686.9              146194.8
##  bs(sqft_living, 15)3  bs(sqft_living, 15)4  bs(sqft_living, 15)5
##              196139.5              210910.1              221111.4
##  bs(sqft_living, 15)6  bs(sqft_living, 15)7  bs(sqft_living, 15)8
##              262596.8              245897.1              313719.1
##  bs(sqft_living, 15)9 bs(sqft_living, 15)10 bs(sqft_living, 15)11
##              341781.4              407321.5              428092.8
## bs(sqft_living, 15)12 bs(sqft_living, 15)13 bs(sqft_living, 15)14
##              643123.3             1560352.3             5928282.7
## bs(sqft_living, 15)15
##             3847879.0
```

```r
lm2 <- lm(price ~ bs(sqft_living, 15), set2)
lm2$coefficients
```

```
##           (Intercept)  bs(sqft_living, 15)1  bs(sqft_living, 15)2
##             250950.10              26313.52              26775.08
##  bs(sqft_living, 15)3  bs(sqft_living, 15)4  bs(sqft_living, 15)5
##             106946.27             126153.12             182164.38
##  bs(sqft_living, 15)6  bs(sqft_living, 15)7  bs(sqft_living, 15)8
##             166021.44             199568.20             206857.97
##  bs(sqft_living, 15)9 bs(sqft_living, 15)10 bs(sqft_living, 15)11
##             286687.23             271987.15             415083.08
## bs(sqft_living, 15)12 bs(sqft_living, 15)13 bs(sqft_living, 15)14
##             421685.33            1716006.54            1123272.99
## bs(sqft_living, 15)15
##            4685073.03
```

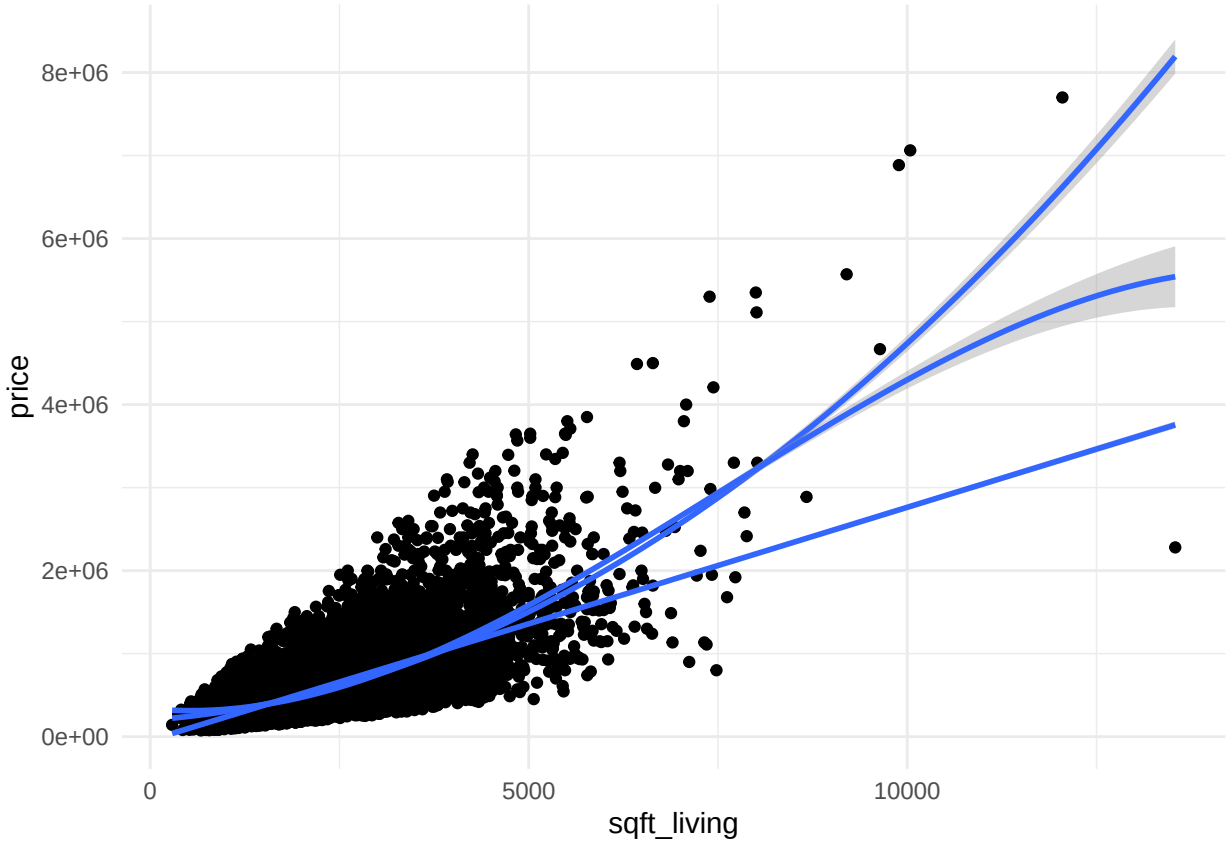
```r
lm3 <- lm(price ~ bs(sqft_living, 15), set3)
lm3$coefficients
```

```
##           (Intercept)  bs(sqft_living, 15)1  bs(sqft_living, 15)2
##              166527.0              116597.7              124066.1
##  bs(sqft_living, 15)3  bs(sqft_living, 15)4  bs(sqft_living, 15)5
##              199160.4              191807.5              245113.4
##  bs(sqft_living, 15)6  bs(sqft_living, 15)7  bs(sqft_living, 15)8
##              279009.6              256316.5              317751.4
##  bs(sqft_living, 15)9 bs(sqft_living, 15)10 bs(sqft_living, 15)11
##              351041.1              407774.4              475986.6
## bs(sqft_living, 15)12 bs(sqft_living, 15)13 bs(sqft_living, 15)14
##              500158.4             1866250.2             1710887.5
## bs(sqft_living, 15)15
##             6465224.1
```

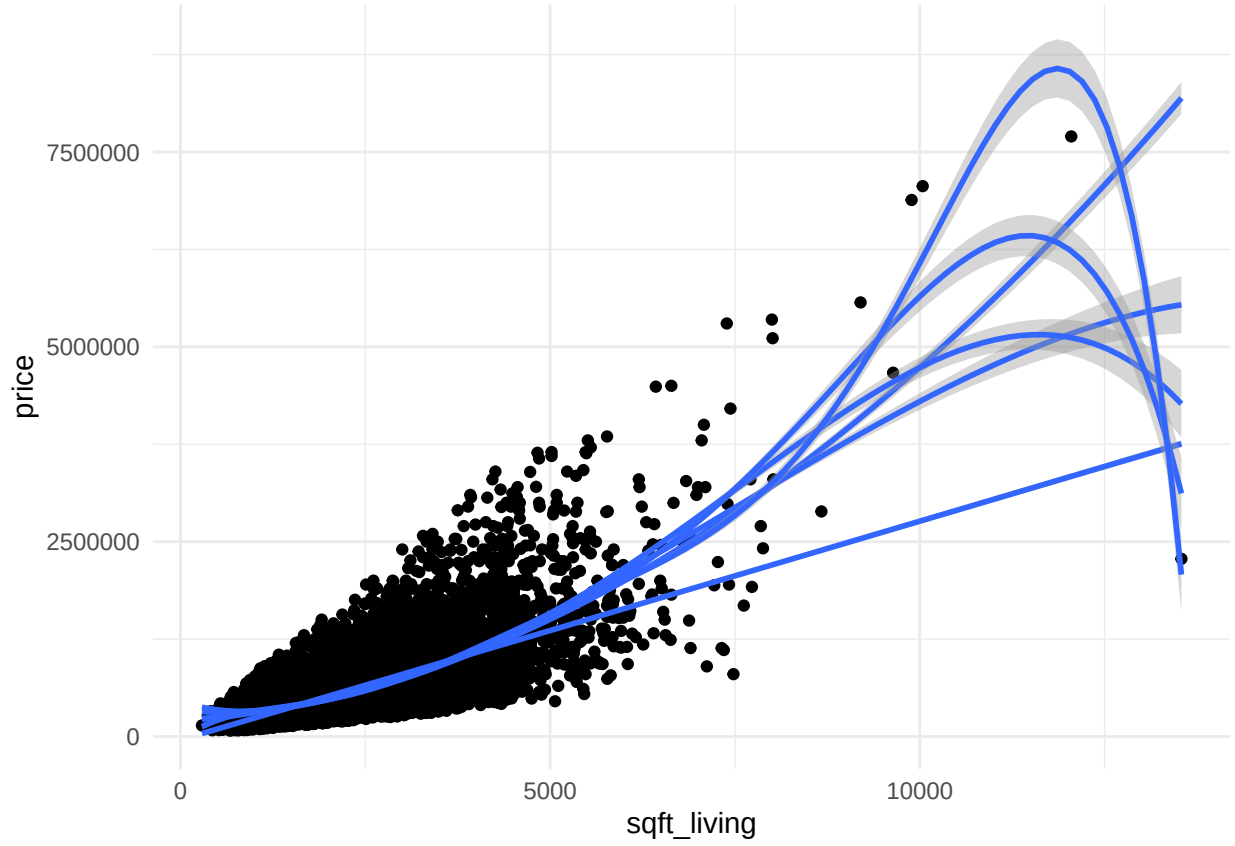
```r
lm4 <- lm(price ~ bs(sqft_living, 15), set4)
lm4$coefficients
```

```
##           (Intercept)  bs(sqft_living, 15)1  bs(sqft_living, 15)2
##             228029.267             -2129.005             80501.134
##  bs(sqft_living, 15)3  bs(sqft_living, 15)4  bs(sqft_living, 15)5
##             128146.566            129159.588            199244.992
##  bs(sqft_living, 15)6  bs(sqft_living, 15)7  bs(sqft_living, 15)8
##             192198.745            215931.358            225834.210
##  bs(sqft_living, 15)9 bs(sqft_living, 15)10 bs(sqft_living, 15)11
##             309849.935            301210.586            447109.390
## bs(sqft_living, 15)12 bs(sqft_living, 15)13 bs(sqft_living, 15)14
##             463843.817           1250348.491           1970280.538
## bs(sqft_living, 15)15
##            3092632.549
```

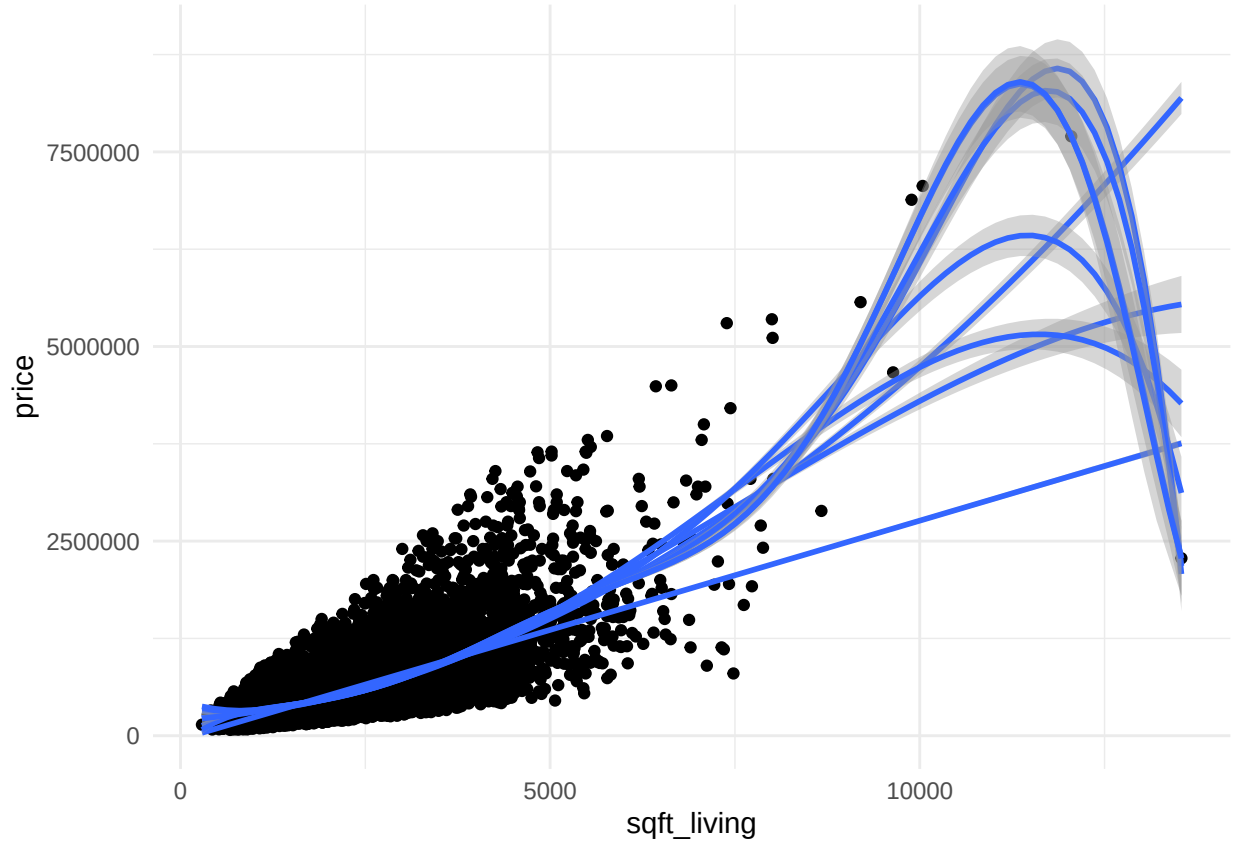
```r
p <- ggplot(data, aes(sqft_living, price)) + geom_point()
p1 <- p + stat_smooth(method = "lm", formula = y ~ x)
p2 <- p1 + stat_smooth(method = "lm", formula = y ~ poly(x, 2))
p3 <- p2 + stat_smooth(method = "lm", formula = y ~ poly(x, 3))
p4 <- p3 + stat_smooth(method = "lm", formula = y ~ poly(x, 4))
p5 <- p4 + stat_smooth(method = "lm", formula = y ~ poly(x, 5))
p6 <- p5 + stat_smooth(method = "lm", formula = y ~ poly(x, 6))
p7 <- p6 + stat_smooth(method = "lm", formula = y ~ poly(x, 7))
p8 <- p7 + stat_smooth(method = "lm", formula = y ~ poly(x, 8))
p9 <- p8 + stat_smooth(method = "lm", formula = y ~ poly(x, 9))
p10 <- p9 + stat_smooth(method = "lm", formula = y ~ poly(x, 10))
p11 <- p10 + stat_smooth(method = "lm", formula = y ~ poly(x, 11))
p12 <- p11 + stat_smooth(method = "lm", formula = y ~ poly(x, 12))
p13 <- p12 + stat_smooth(method = "lm", formula = y ~ poly(x, 13))
p14 <- p13 + stat_smooth(method = "lm", formula = y ~ poly(x, 14))
p15 <- p14 + stat_smooth(method = "lm", formula = y ~ poly(x, 15))

p3
```



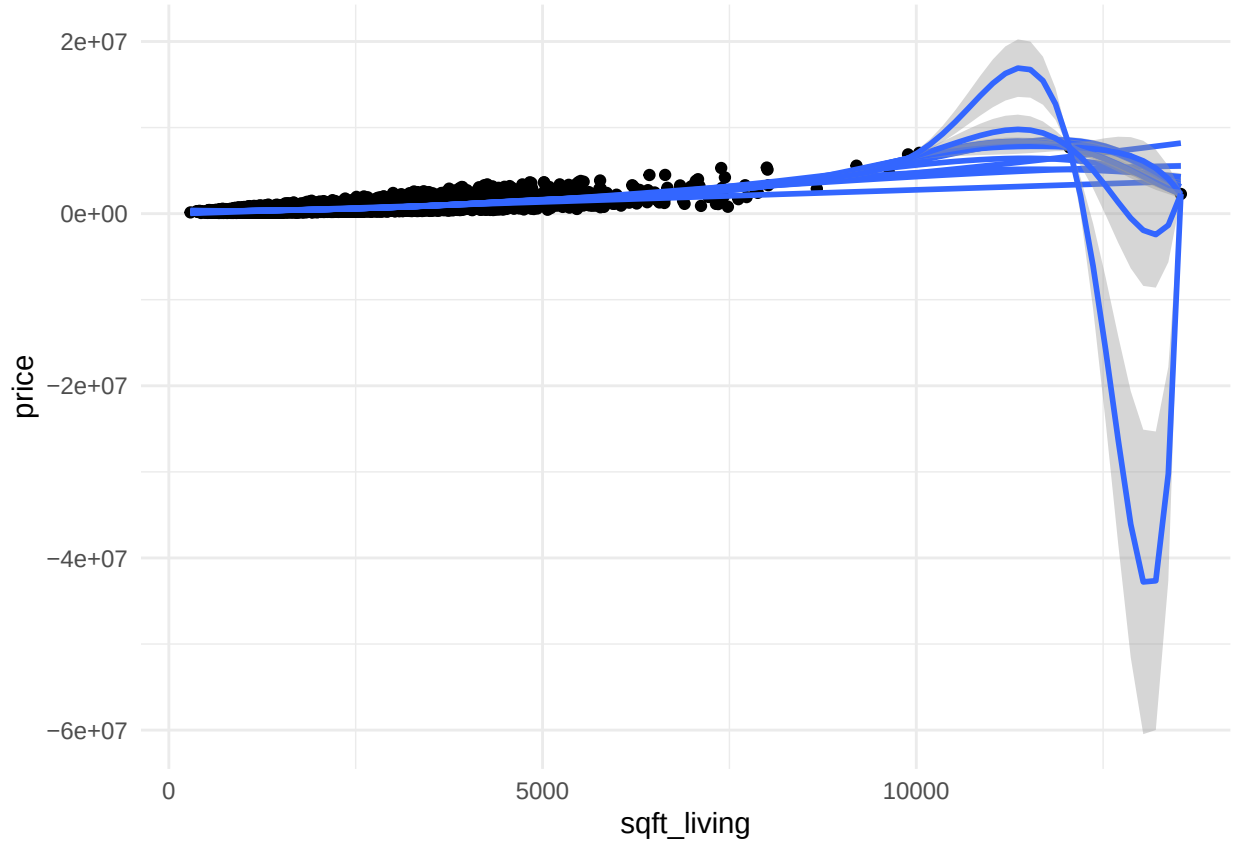

```
p6
```



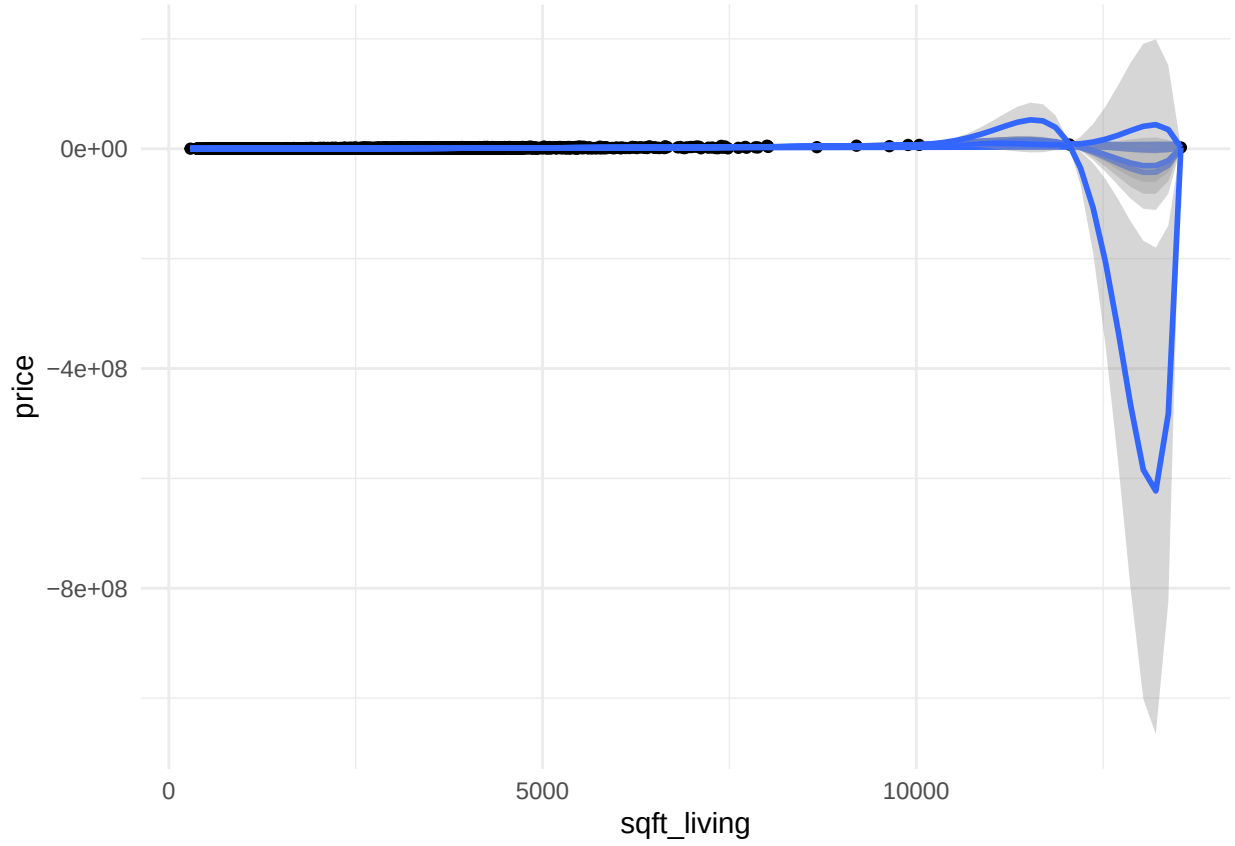

```
p9
```




```
p12
```




```
p15
```




```
sum((set1$price - lm1$fitted.values) ** 2)
```

```
## [1] 3.484187e+14
```

```
sum((set2$price - lm2$fitted.values) ** 2)
```

```
## [1] 3.003444e+14
```

```
sum((set3$price - lm3$fitted.values) ** 2)
```

```
## [1] 3.429065e+14
```

```
sum((set4$price - lm4$fitted.values) ** 2)
```

```
## [1] 3.263015e+14
```

```
train <- read.csv("wk3_kc_house_train_data.csv")
valid <- read.csv("wk3_kc_house_valid_data.csv")
test <- read.csv("wk3_kc_house_test_data.csv")

model.RSS <- function(train, valid, degree) {
  if (degree <= 15) {
```

```
    model <- lm(price ~ poly(sqft_living, degree), train)
    predict <- predict(model, data.frame(sqft_living = valid$sqft_living))
    RSS <- sum((valid$price - predict) ** 2)
    return(RSS)
  } else {
    model <- lm(price ~ bs(sqft_living, degree), train)
    predict <- predict(model, data.frame(sqft_living = valid$sqft_living))
    RSS <- sum((valid$price - predict) ** 2)
    return(RSS)
  }
}

RSS.history <- data.frame(RSS = NA, times = NA)
for (i in 1:15) {
  temp <- data.frame(RSS = model.RSS(train, valid , i), times = i)
  RSS.history <- rbind(RSS.history, temp)
}
RSS.history <- na.omit(RSS.history)
RSS.history
```

```
##             RSS times
## 2  6.290979e+14     1
## 3  6.239551e+14     2
## 4  6.258203e+14     3
## 5  6.299873e+14     4
## 6  6.200456e+14     5
## 7  6.201190e+14     6
## 8  9.869390e+14     7
## 9  6.959903e+14     8
## 10 3.415053e+16     9
## 11 7.269767e+14    10
## 12 1.097322e+19    11
## 13 9.205476e+20    12
## 14 9.899250e+22    13
## 15 1.241804e+24    14
## 16 1.425413e+24    15
```

```
RSS.history %>%
  filter(RSS == min(RSS.history$RSS))
```

```
##            RSS times
## 1 6.200456e+14     5
```

```
model.RSS(train, test, 1)
```

```
## [1] 1.423479e+14
```