STAT5003

Week 2: Regression and Smoothing

Dr. Justin Wishart

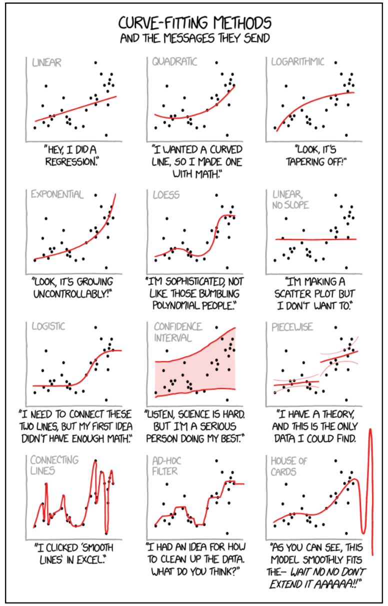
Readings 🏛 and R functions covered

- Introduction to Statistical Learning James, Witten, Hastie, and Tibshirani (2013)
  - Chapter 3 (Linear regression)
  - Chapter 7.4 to 7.6 (Smoothing)
- R functions
  - `outcome ~ feature1 + feature2` (formulae)
  - `lm` (Linear model)
  - `confint` (confidence intervals)
  - `subset` (argument and function)
  - `predict` (Make predictions from a model)

Regression

Regression

- Numerically fitting the model is easy

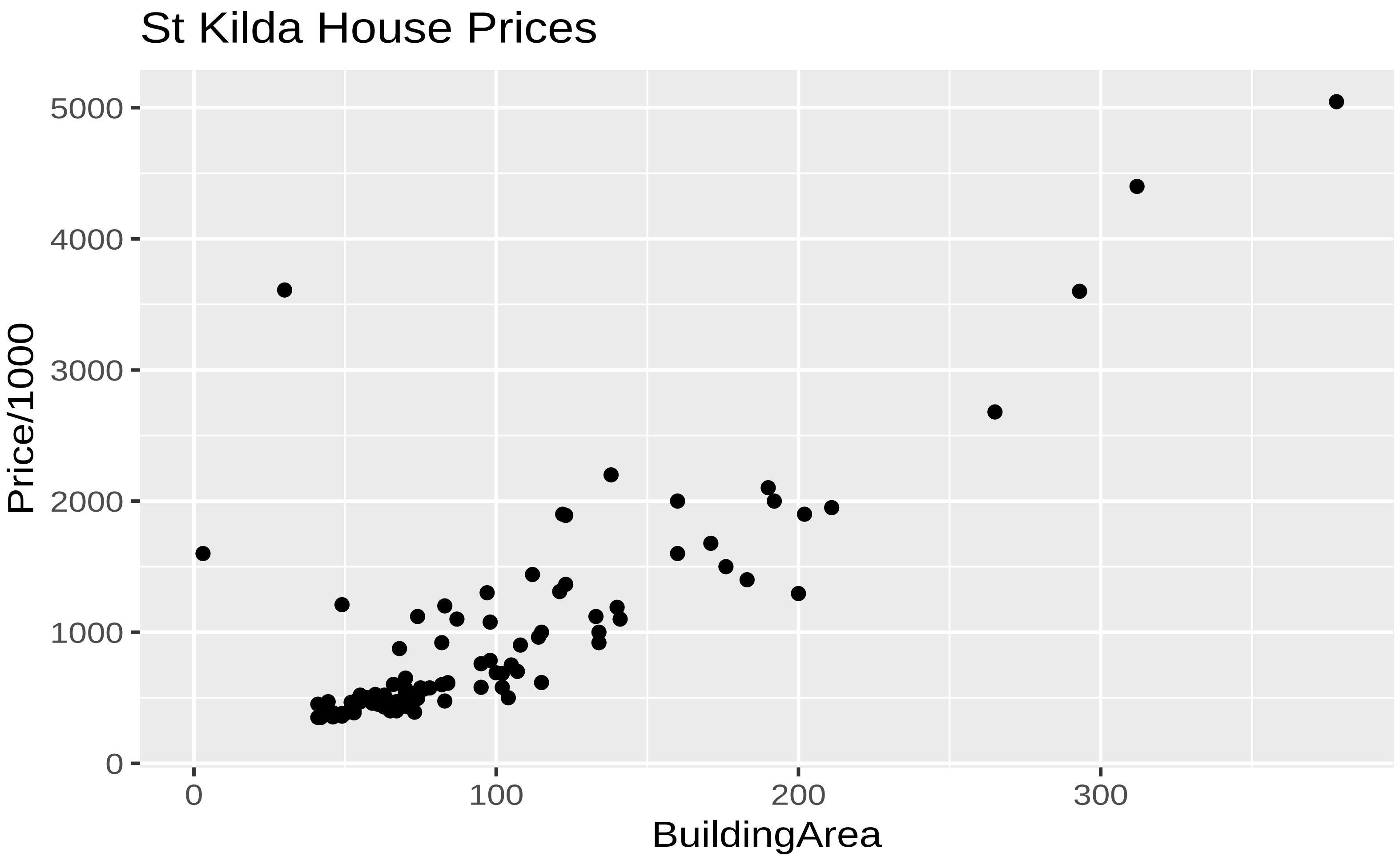


- Knowing how to appropriately fit the model is where you add value.

Line of Best Fit

The prediction problem

What is the price of a 100 sqm house in St Kilda?

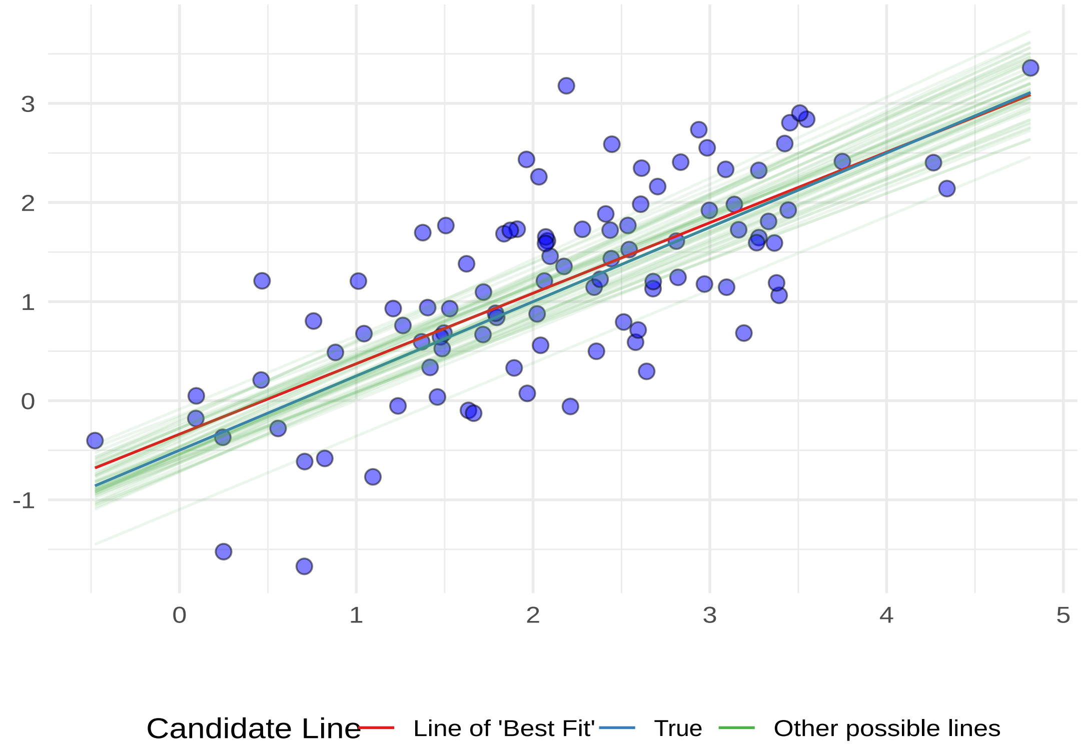


The linear regression model

$$
\begin{aligned}
Y \; &= \beta_0 + \beta_1 X + \varepsilon \\
&\downarrow \\
y_i &= b_0 + b_1 x_i + e_i
\end{aligned}
$$

- X is the predictor (feature or independent variable)
- Y is the response (target or dependent variable)
- β_0 is the intercept of the regression line
  - Expected value of Y when $X = 0$
- β_1 is the slope of the regression line
  - mean increase in Y for a *unit* increase in X
- ε is the unexplained variation or random error.
  - Classically assumed to be normally distributed with mean zero and finite variance.

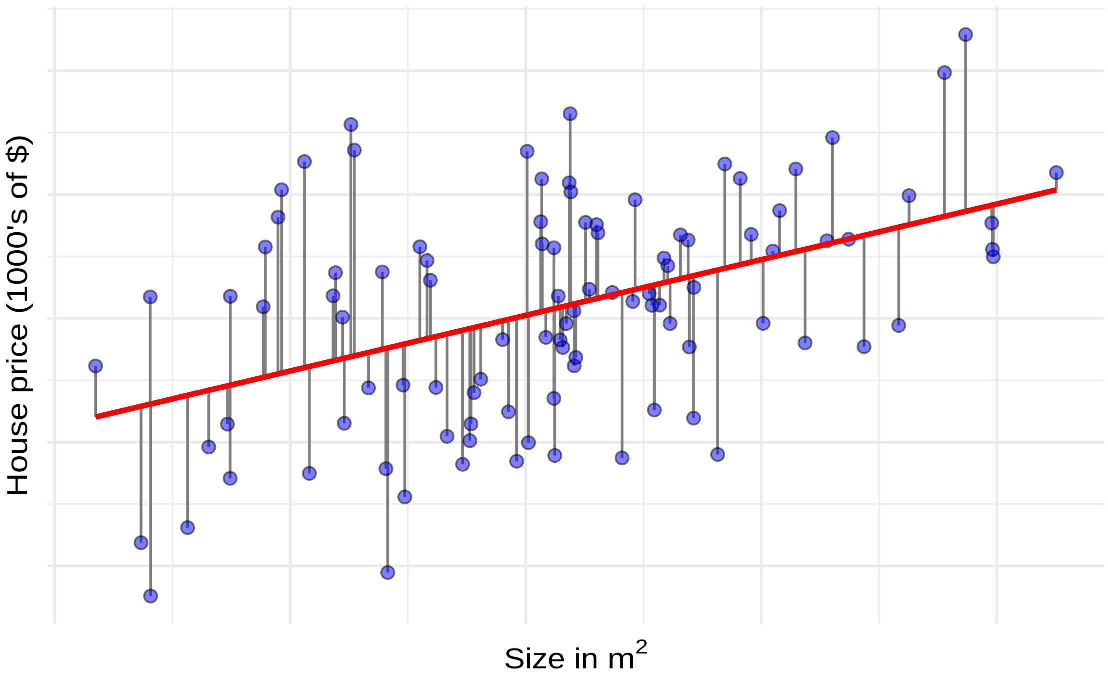
Performance of regression estimates



- Data was simulated from model $Y = -0.5 + 0.75X + \varepsilon$
- True line shown in blue
- Standard linear regression fit shown in red
- Why not one of the green lines?

How to determine the best estimates of $\beta_0 + \beta_1 X$?

- The notion of best needs a criterion to measure against.



- Easiest mathematical solution is the least squares criterion
  - Minimise the residual sum of squares $\mathrm{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$

Least squares equations

- Can show by simple calculus the following:
    - Regression (slope) coefficient: $b_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = \frac{\mathrm{cov}(x, y)}{\mathrm{var}(x)}$
    - Intercept: $b_0 = \overline{y} - b_1\overline{x}$
- This leads to the estimated regression line:

$$\hat{y} = b_0 + b_1 x$$

- Least squares regression line since it minimises the residual sum of squares.

Basic uses of Simple Linear Regression

Prediction using `lm`

```
lm.fit <- lm(Price ~ BuildingArea, data = st.kilda.data)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Price ~ BuildingArea, data = st.kilda.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -817415 -201614  -85181   19895 3403199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -129484.0    91775.9  -1.411    0.161
## BuildingArea   11209.5      799.8  14.015   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 490300 on 99 degrees of freedom
## Multiple R-squared:  0.6649,    Adjusted R-squared:  0.6615
## F-statistic: 196.4 on 1 and 99 DF,  p-value: < 2.2e-16
```

Standard error of population mean

- Consider single population estimation problem .
  - Wish to estimate some mean, μ, of some random variable Y .
  - If Y_i is sampled then $\hat{\mu} = \overline{Y}$ estimates μ with

$$Var(\hat{\mu}) = (SE(\hat{\mu}))^2 = \frac{\sigma^2}{n}$$

  - σ^2 is the variance of Y_i
  - n is the sample size.

Standard error of regression coefficient estimates

- Same concept applies to the regression estimates

$$SE(\hat{\beta}_0) = \sigma\sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}$$

$$SE(\hat{\beta}_1) = \frac{\sigma}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}}$$

where $\sigma^2 = Var(\varepsilon)$

- As $n \to \infty$, $SE(\hat{\beta}_0) \to 0$ and $SE(\hat{\beta}_1) \to 0$
- Interestingly, if the x_i are more spread out, the standard errors will be smaller
  - more leverage to estimate the parameters.

Using standard errors to compute confidence intervals

```r
summary(lm.fit) # Truncated output with coefficient table
```

```
...
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -129484.0    91775.9  -1.411    0.161
## BuildingArea   11209.5      799.8  14.015   <2e-16 ***
## ---
##
## Residual standard error: 490300 on 99 degrees of freedom
...
```

- We can use the standard error to estimate the 95% confidence interval as:
  - $(\hat{\beta}_1 - t_{n-2,0.975}\mathrm{SE}(\hat{\beta}_1), \hat{\beta}_1 + t_{n-2,0.975}\mathrm{SE}(\hat{\beta}_1)) = b_1 \pm t_{n-2,0.975}\mathrm{SE}(b_1) = b_1 \pm t_{99,0.975}\mathrm{SE}(b_1)$
- In our housing example, the 95% confidence interval for the coefficient of `BuildingArea` is [9622.6968, 12796.3032]

$$b_1 \pm t_{n-2,0.975}\mathrm{SE}(b_1) = 1.12095 \times 10^4 \pm 1.984 \times 799.8 = (9622.6968, 12796.3032)$$

Confidence intervals of regression coefficients

- More directly in R code.
  - Use the `confint` function.

```
confint(lm.fit)
```

```
##                    2.5 %   97.5 %
## (Intercept)  -311587.233 52619.18
## BuildingArea    9622.491 12796.50
```

- This is exact and no precision lost to rounding error.
- Easy to change confidence level (99% below)

```
confint(lm.fit, level = 0.99)
```

```
##                   0.5 %    99.5 %
## (Intercept)  -370524.63 111556.57
## BuildingArea    9108.86  13310.13
```

Is `BuildingArea` a good predictor of price?

```
summary(lm.fit) # truncated for coefficient table
```

```
...
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -129484.0    91775.9  -1.411    0.161
## BuildingArea   11209.5      799.8  14.015   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
...
```

- Linear regression assumes $Y = \beta_0 + \beta_1 X + \varepsilon$
- If `BuildingArea` is not linearly related to `Price`, then $\beta_1 = 0$.
- Can conduct a test of significance $H_0 : \beta_1 = 0$ against $H_1 : \beta_1 \neq 0$
- Can conduct a hypothesis test by computing the t-statistic:

$$t = \frac{\hat{\beta}_1 - \beta_1}{SE(\hat{\beta}_1)} \overset{H_0}{=} \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$$

Is `BuildingArea` a good predictor of price?

```
summary(lm.fit) # truncated for coefficient table
```

```
...
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -129484.0    91775.9  -1.411    0.161
## BuildingArea   11209.5      799.8  14.015   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
...
```

- The p-values for each significance test in the last column.
- Recall, p-value gives the probability of observing your test statistic (and other scenarios support H_1) assuming H_0 is true.
- Small p-value here gives very little evidence to support the claim that there is no relationship between `Price` and `BuildingArea`

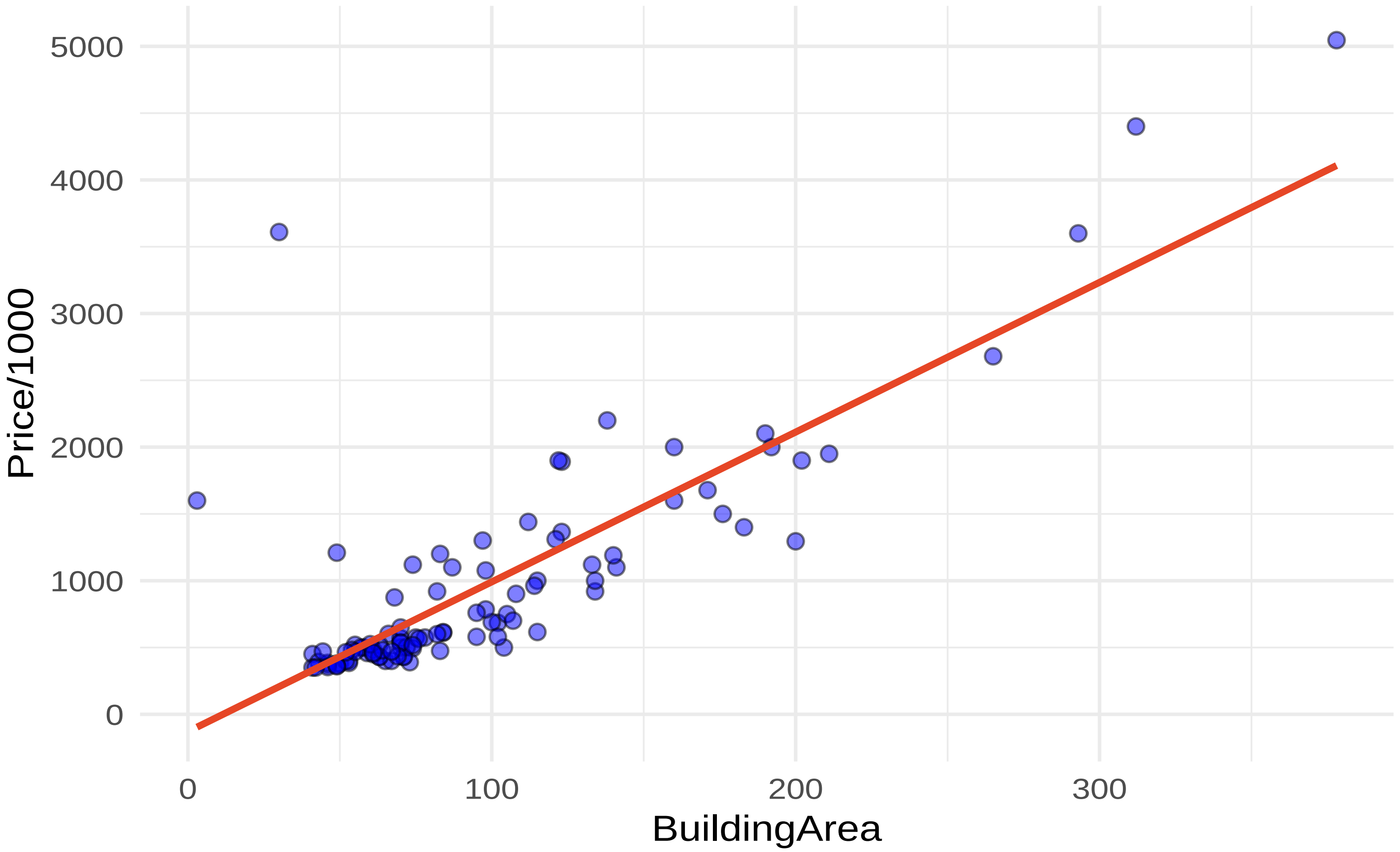
Goodness of fit statistic

- Goodness of fit is measured by the coefficient of determination or R^2

$$
\begin{aligned}
R^2 &= \frac{\text{Total Sum of Squares} - \text{Residual Sum of Squares}}{\text{Total Sum of Squares}} \\
&= \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}
\end{aligned}
$$

- R^2 is a measure between 0 and 1
- It measures the proportion of variation in the response Y, explained by the linear regression on X
  - A value of 0 indicates **none** of the variance in Y can be explained linearly by X
  - A value of 1 indicates **all** of the variance in Y can be explained linearly by X

Linear regression fit



Estimating the price of a 100 m^2 house in St Kilda

```r
new.100 <- data.frame(BuildingArea = 100)
predict(lm.fit, new.100, interval = "confidence")
```

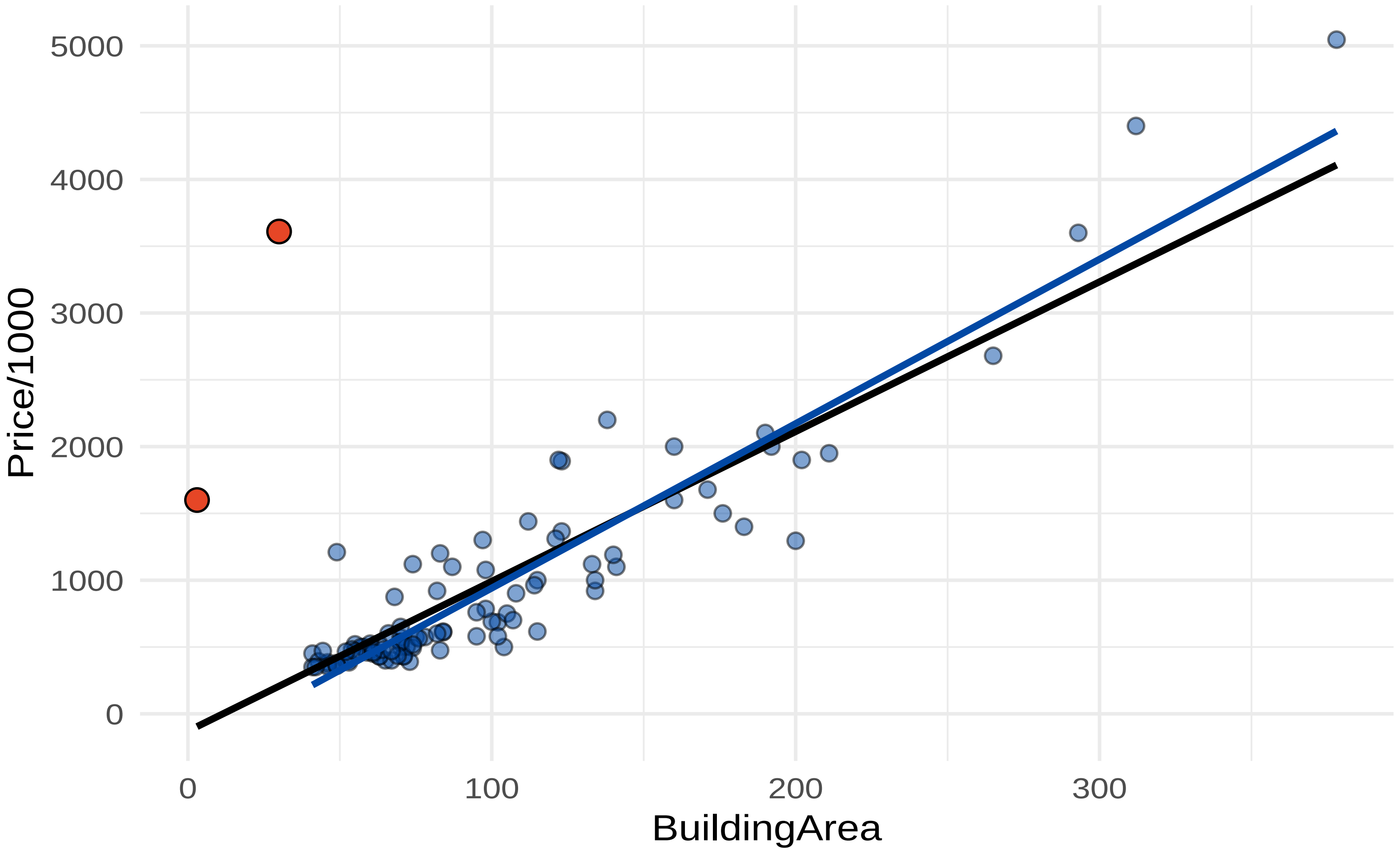
```
##        fit      lwr     upr
## 1 991465.5 894562.7 1088368
```

```r
predict(lm.fit, new.100, interval = "prediction")
```

```
##        fit      lwr     upr
## 1 991465.5 13820.26 1969111
```

Fit improvements

- Remove outliers: black line gives overall fit, blue line fit only to blue data (without red points)



Linear fit after removing the outliers

```
lm.without.outliers <- lm(Price/1000 ~ BuildingArea, data = st.kilda.data, subset = BuildingArea >= 40)
summary(lm.without.outliers)
```

```
##
## Call:
## lm(formula = Price/1000 ~ BuildingArea, data = st.kilda.data,
##     subset = BuildingArea >= 40)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -876.75 -137.30  -18.27  109.28  896.31
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -289.254     57.471  -5.033 2.22e-06 ***
## BuildingArea   12.305      0.496  24.807  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 298.5 on 97 degrees of freedom
## Multiple R-squared:  0.8638,    Adjusted R-squared:  0.8624
## F-statistic: 615.4 on 1 and 97 DF,  p-value: < 2.2e-16
```

Extending Simple Linear Regression

- What if I have more than one feature (predictor)?
- House prices depend on more than just BuildingArea! What about
  - land area.
  - Dwelling type (apartment vs unit vs house vs ...)
  - Suburb (location, location, location!)

formulae

- Example formula `Response ~ Predictor1 + Predictor2 + Predictor3`
- Left hand side of ~ is the response variable (target to predict)
- Right hand side of ~ are the the predictor variables (features)
- Relationship is assumed to be additive
  - I.e. each additional predictor is added to explain the response $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots$
- Interaction or multiplicative terms are denoted with `:` and `*` which are beyond the scope fo this course.
  - Would be used to define other relationships
  - E.g. $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1 X_2 + \beta_3 X_2 + \ldots$

Multiple linear regression

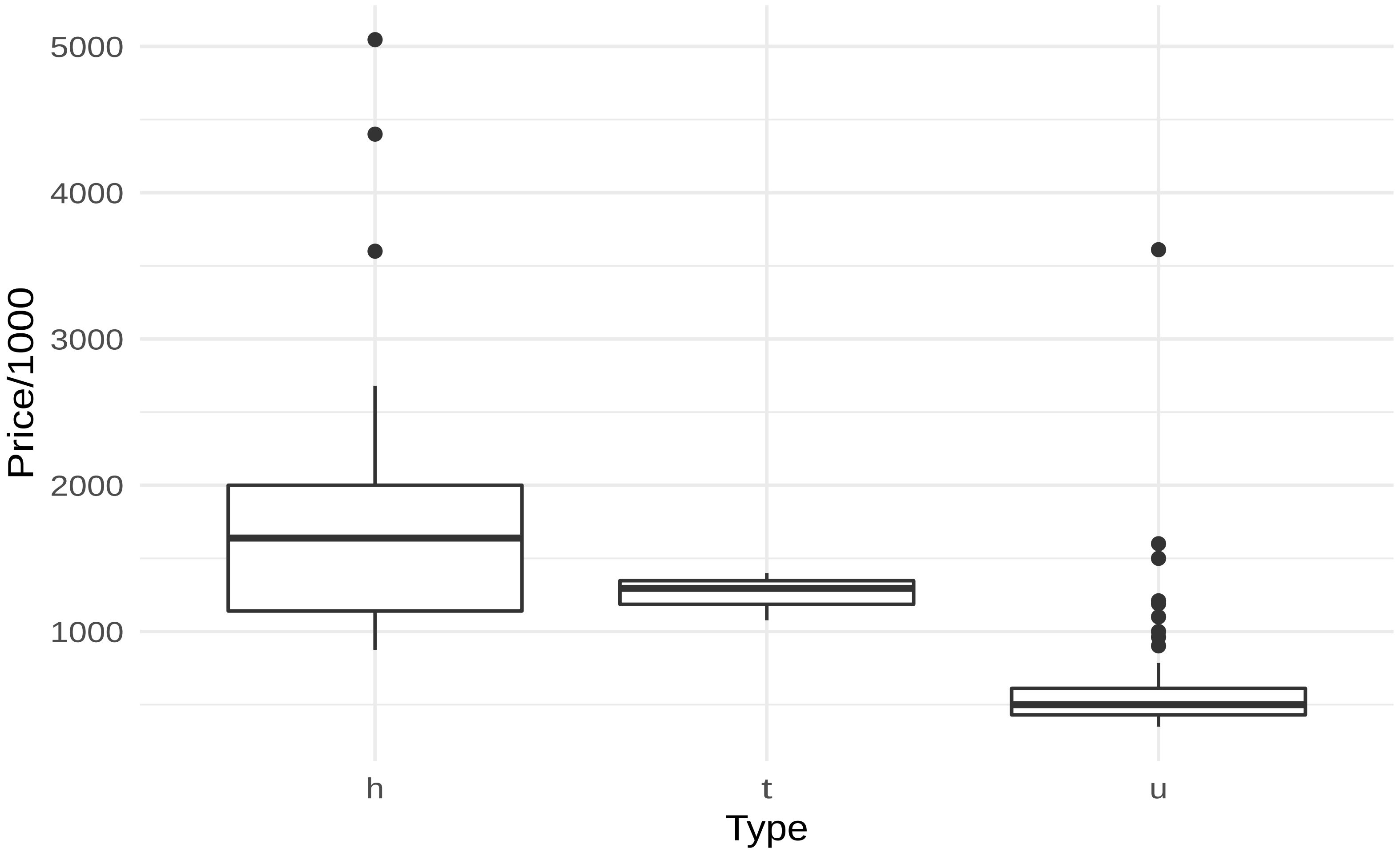
- Real life problems usually have more than one predictor.
  - Simple linear (single variable) regression can be extended to multiple predictors

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_p X_p + \varepsilon$$

- The interpretation is the β_p coefficient denotes the average increase/decrease in Y for each single unit increase in X_p, holding all the other predictors fixed.

Extending the house prediction model to multiple features

- Perhaps 100 m^2 houses cost more than 100 m^2 units?



Multiple regression with `lm`

```
multi.lm <- lm(Price/1000 ~ Type + BuildingArea, data = st.kilda.data)
summary(multi.lm)
```

```
## 
## Call:
## lm(formula = Price/1000 ~ Type + BuildingArea, data = st.kilda.data)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -700.3 -173.1  -65.9   18.6 3389.6 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   342.865    186.764   1.836  0.06945 .  
## Typet        -613.953    286.272  -2.145  0.03448 *  
## Typeu        -408.417    139.915  -2.919  0.00436 ** 
## BuildingArea    9.533      1.014   9.398 2.68e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 469.5 on 97 degrees of freedom
## Multiple R-squared:  0.6989,	Adjusted R-squared:  0.6896 
## F-statistic: 75.06 on 3 and 97 DF,  p-value: < 2.2e-16
```

Model interpretation

```r
summary(multi.lm)
```

```
...
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   342.865    186.764   1.836  0.06945 .
## Typet        -613.953    286.272  -2.145  0.03448 *
## Typeu        -408.417    139.915  -2.919  0.00436 **
## BuildingArea    9.533      1.014   9.398 2.68e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
...
```

```r
multi.pred.data <- data.frame(BuildingArea = rep(100, 3), Type = c("u", "t", "h"))
predict(multi.lm, newdata = multi.pred.data)
```

```
##         1         2         3
##  887.7252  682.1894 1296.1427
```

Nonparametric regression or Smoothing

Parametrics vs non-parametric methods

- Parametric methods involve selecting a statistical model (e.g. linear regression model) and fitting the parameters of the model (e.g. slope, intercept) using the training data
- Nonparametric methods don't require selecting a strict model. The data is allowed to *speak for itself*. However, don't have easily interpretable parameters. They are generally intended for description rather than formal inference (e.g. k-nearest neighbor smoothing)

Data smoothing

With predictor-response data, the random response variable Y is assumed to be a non-linear function of the predictor variable X.

$$Y_i = f(X_i) + \varepsilon_i$$

- f is some fixed, non-linear smooth function.
- ε_i is a zero-mean random variable.
- Smoothing is a non-parametric method to estimate f.

Local averaging

- Most smoothers (smoothing functions) rely on the concept of *local averaging*
  - In contrast, simple linear regression attempts to fit the best global line.
- E.g. Suppose you want to determine the response Y conditional on x.
  - The Y_i whose corresponding x_i are near x should be averaged with higher weight to atttempt to estimate $f(x)$.
- A generic local-averaging smoother can be written as

$$\hat{f}(x) = \text{average}\,(Y_i | x_i \in N(x))$$

  - where average is some generalised averaging operation.
  - $N(x)$ is some neighbourhood of x.

Constant-Span Running Mean, k-nearest neighbours

- A simple smoother takes the sample mean of k nearby points
- We define $N(x_i)$ as x_i itself, the $(k-1)/2$ points whose predictor values are nearest below x_i, and the $(k-1)/2$ points whose predictor values are nearest above x_i
- This neighbourhood is termed the *symmetric nearest neighbourhood*, and the smoother is called a moving average or a *k*-nearest neighbours (kNN) smoother.
- The constant-span running-mean smoother can be written as:

$$\hat{f}(x_i) = \text{mean}\left[Y_j \text{ such that} \max\left(i - \frac{k-1}{2}, 1\right) \le j \le \min\left(i + \frac{k-1}{2}, n\right)\right]$$

Regression splines

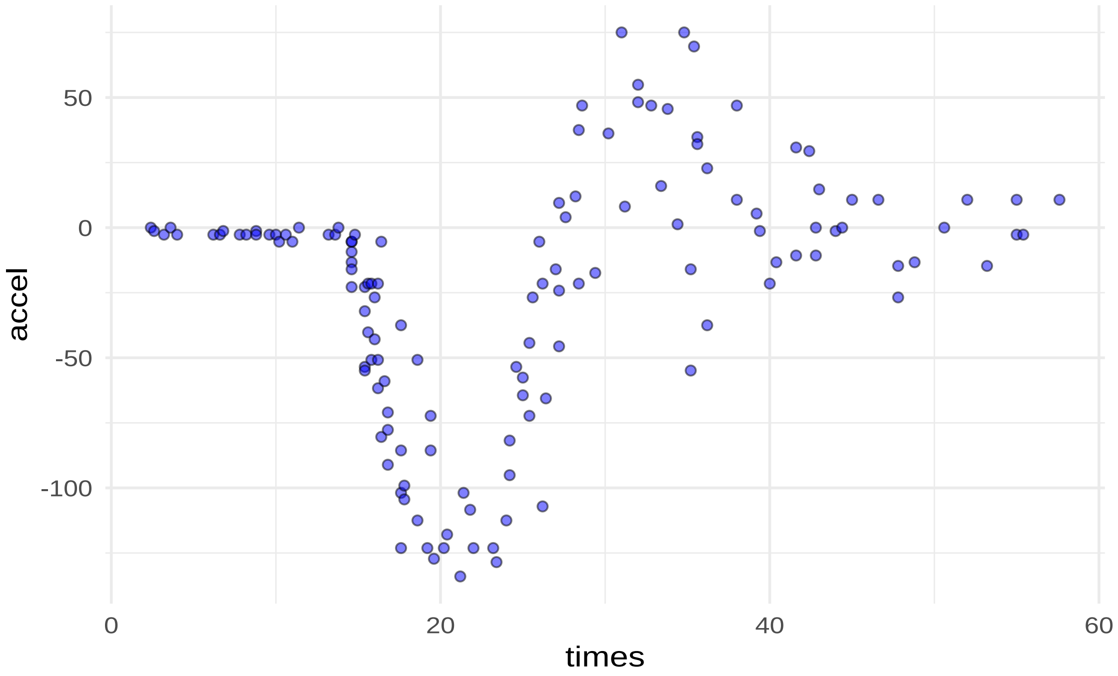
- Fit piece-wise functions, where each function can be a d-dimensional polynomial function
- Constrain the function to be smooth and continuous
- Cubic spline fits cubic polynomial functions, with the constraints:
  - continuous at each knot, continuity of the 1st derivative and continuity of the 2nd derivative
- Advantage of the cubic spline is that the curve looks smooth to the eye, and can be used to fit almost any function

Loess

- Loess is a Locally weighted scatterplot smoothing method
- The loess (**LO**cal regr**ESS**ion) smoother is a widely used method with good robustness properties.
- It is essentially a weighted running-line smoother, except that each local line is fitted using a robust method rather than least squares.
- As a result, the smoother is nonlinear.
- Loess is computationally intensive and require densely sampled data.

Local regression

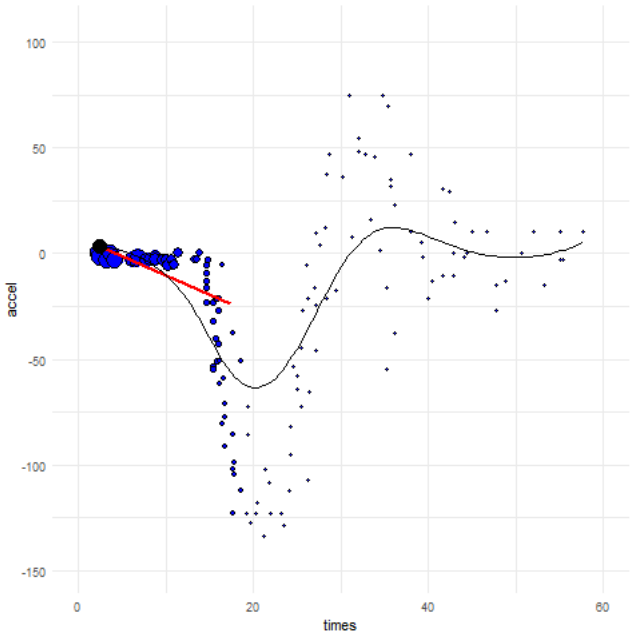
- Fitting local linear fits that are weighted.
- Formula for local constant fit is $\hat{f}(x) = \frac{\sum_{i=1}^{n} Y_i K((X_i - x)/h)}{\sum_{i=1}^{n} K((X_i - x)/h)}$



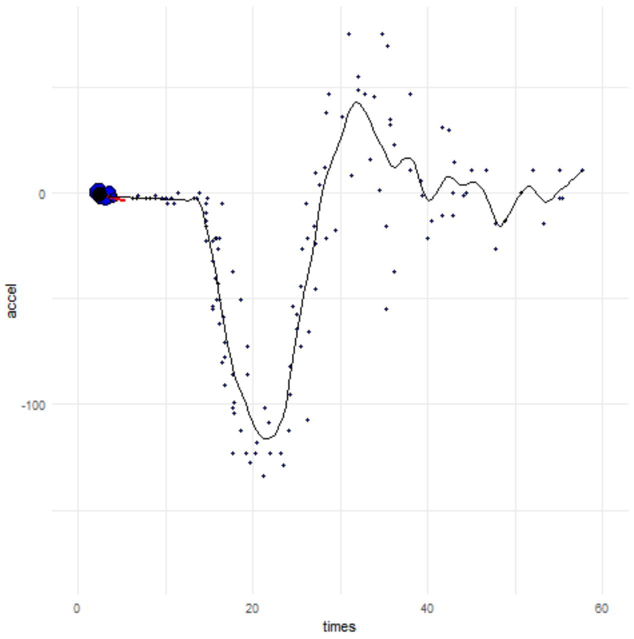
- Sharp changes in behaviour from the acceleration on a crash test dummy.

Local regression animation

- Using a large averaging window



- Using a smaller averaging window



Nonparametric smoothing vs linear regression

- Advantages of non-parametric smoothing
  - Can model non-linear functions (e.g. splines, loess)
  - Does not make any assumption about the functional form of the data
- Advantages of linear regression
  - Computationally efficient, even for multivariate linear regression
  - Model is interpretable, i.e. one can know the statistical meaning of the estimated slope parameters.

Reference list

James, G., D. Witten, T. Hastie, et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.