Chapter 3: Linear Regression

Linear regression is a simple approach for supervised learning when the response is quantitative. Linear regression has a long history and we could actually spend most of this semester talking about it.

Although linear regression is not the newest, shiniest thing out there, it is still a highly used technique out in the real world. It is also useful for talking about more modern techniques that are **generalizations** of it. Ridge regression, Lasso, logistic regression, GAMs.

We will review some key ideas underlying linear regression and discuss the least squares approach that is most commonly used to fit this model.

Linear regression can help us to answer the following questions about our `Advertising` data:

1. Is there a relationship btw/ advertising and sales?
   i.e. should people spend money on ads?
2. How strong is that relationship?
   i.e. how well can we predict sales based on ad budgets?
3. Which media contribute to sales?
4. How accurately can we predict the effect of each medium on sales?
5. How accurately can we predict future sales?
6. Is the relationship linear?
7. Is there synergy among the ad media?
   i.e. is \$50k for TV and \$50k for radio "better" for sales than \$100k on radio or TV alone?

1 Simple Linear Regression

Simple Linear Regression is an approach for predictiong a quantitative response Y on the basis of a single predictor variable X.

It assumes:

- approximately linear relationship between X and Y
- random error term is Normally distributed
- random error term has constant variance

Which leads to the following model:

$$Y = \overbrace{\beta_0 + \beta_1 X}^{\text{linear relationship}} + \varepsilon$$

$$\varepsilon \sim \underbrace{N(0, \sigma^2)}_{\text{assumptions about error}}$$

For example, we may be interested in regressing `sales` onto `TV` by fitting the model

$$\text{sales} = \beta_0 + \beta_1 \times \text{TV} + \varepsilon$$

unknown constants (intercept & slope).
("parameters" or "model coefficients").

Once we have used training data to produce estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we can predict future sales on the basis of a particular TV advertising budget.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

prediction of sales

particular TV budget

1.1 Estimating the Coefficients

In practice, β_0 and β_1 are **unknown**, so before we can predict $\hat{y}$, we must use our training data to estimate them.

"fit the model"

"train the model"

Let $(x_1, y_1), \ldots, (x_n, y_n)$ represent training data n observation pairs, each of which consists of a measurement of X and Y.

In advertising data
X = TV ad budget
Y = sales
$n = 200$ observations

Goal: Obtain coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ such that the linear model fits the available data well.

i.e. $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i \quad i = 1, \ldots, n$

We want to find intercept $\hat{\beta}_0$ and slope $\hat{\beta}_1$ s.t. resulting line is "close" to $n=200$ points.

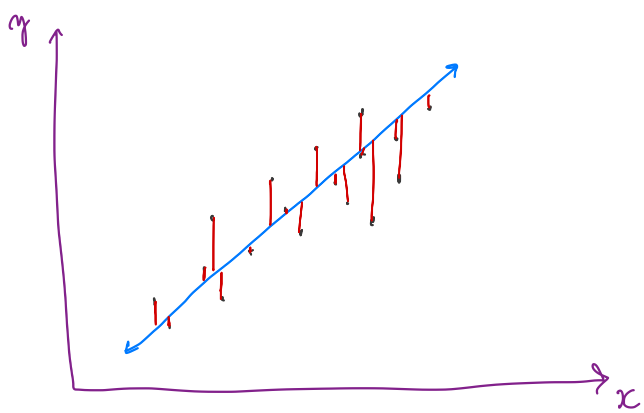
The most common approach involves minimizing the *least squares* criterion.

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ prediction for y based on ith value of X

$e_i = y_i - \hat{y}_i$ ith "residual"

$RSS = e_1^2 + \ldots + e_n^2$ residual sum of squares

choose $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize RSS.



The least squares approach results in the following estimates:

Use calculus → take derivatives set to 0, solve for $\hat{\beta}_0$ and $\hat{\beta}_1$.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

"least squares coefficients"

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

We can get these estimates using the following commands in `R` and `tidymodels`:

```
library(tidymodels) ## load library

## load the data in
ads <- read_csv("../data/Advertising.csv", col_select = -1)

## fit the model
lm_spec <- linear_reg() |>
  set_mode("regression") |>
  set_engine("lm")
```

general model specification

least squares approach

```
slr_fit <- lm_spec |>
  fit(sales ~ TV, data = ads)
```

data frame.

formula "regress y on x"

y ~ x

```
slr_fit |>
  pluck("fit") |>
  summary()
```

look into slr_fit |> tidy().

```
##
## Call:
## stats::lm(formula = sales ~ TV, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

1.2 Assessing Accuracy

Recall we assume the *true* relationship between X and Y takes the form

$$Y = f(X) + \varepsilon$$

f unknown, ε mean-zero random error

If f is to be approximated by a linear function, we can write this relationship as

population regression line.

average increase in Y associated w/ 1 unit increase in X

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

catch all term for what we miss w/ this model – true relationship f may not be linear, may be other variables that cause variation in Y, measurement error.

expected value of Y when $X=0$

and when we fit the model to the training data, we get the following estimate of the *population model*

least squares line.

$$Y = \hat\beta_0 + \hat\beta_1 X$$

But how close this this to the truth? measure w/ standard error.

$$\sqrt{Var(\hat\beta_0)} = SE(\hat\beta_0) = \sqrt{\sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}\right]}$$

$$\sqrt{Var(\hat\beta_1)} = SE(\hat\beta_1) = \sqrt{\frac{\sigma^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}$$

$Var(\varepsilon)$

In general, σ^2 is not known, so we estimate it with the *residual standard error*, $RSE = \sqrt{RSS/(n-2)}$.

residual sum of squares

We can use these standard errors to compute confidence intervals and perform hypothesis tests.

95% CI for β_1: $\hat\beta_1 \pm 2\widehat{SE}(\hat\beta_1)$

95% CI for β_0: $\hat\beta_0 \pm 2\widehat{SE}(\hat\beta_0)$

Hypothesis Test:

H_0: There is no relationship btw/ X and Y $\iff$ $H_0: \beta_1 = 0$

H_a: There is a (linear) relationship btw/ X and Y $\quad$ $H_a: \beta_1 \neq 0$

?: is $\hat\beta_1$ far enough away from 0 to be confident it is not zero? How far is enough? depends on $SE(\hat\beta_1)$.

$$t = \frac{\hat\beta_1 - 0}{\widehat{SE}(\hat\beta_1)} \sim t_{n-2} \implies$$

compute P(observing any number equal to $|t|$ or larger) = p-value.

Small p-value means highly unlikely to see this t given $H_0 \implies$ reject H_0!

Once we have decided that there is a significant linear relationship between X and Y that is captured by our model, it is natural to ask

> To what extent does the model fit the data?

The quality of the fit is usually measured by the *residual standard error* and the R^2 statistic.

RSE: Roughly speaking, the RSE is the average amount that the response will deviate from the true regression line. This is considered a measure of the *lack of fit* of the model to the data.

R^2: The RSE provides an absolute measure of lack of fit, but is measured in the units of Y. So, we don't know what a "good" RSE value is! R^2 gives the proportion of variation in Y explained by the model.

i.e. will be between 0 and 1 always!

Advertising data example

```
slr_fit |>
  pluck("fit") |>
  summary()
```

$\hat{\beta}_0, \hat{\beta}_1$ $\quad$ $\hat{SE}(\hat{\beta}_0), \hat{SE}(\hat{\beta}_1)$ $\quad$ $H_0: \beta_i = 0$ vs. $H_a: \beta_i \neq 0$ $\quad$ $i=0,1$

RSE

```
##
## Call:
## stats::lm(formula = sales ~ TV, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

R^2 = proportion of variability in Y explained by a linear relationship w/ X

2 Multiple Linear Regression

Simple linear regression is useful for predicting a response based on one predictor variable, but we often have **more than one** predictor.

How can we extend our approach to accommodate additional predictors?

- We could run separate SLR for each predictor

How to make single prediction for y based on levels of all predictors?

Also, each model would ignore the other predictors... but what if they are related?

↳ misleading results.

Solution: We can give each predictor a separate slope coefficient in a single model.

association w/ response

predictor

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$$

We interpret β_j as the "average effect on Y of a one unit increase in X_j, *holding all other predictors fixed*".

In our `Advertising` example,

$$\text{sales} = \beta_0 + \beta_1 \text{TV} + \beta_2 \text{radio} + \beta_3 \text{newspaper} + \varepsilon$$

2.1 Estimating the Coefficients

As with the case of simple linear regression, the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are unknown and must be estimated. Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, we can make predictions using the formula

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p.$$

now instead of a line, we are fitting a hyperplane.

The parameters are again estimated using the same least squares approach that we saw in the context of simple linear regression.

alternative way to specify the model.

```
# mlr_fit <- lm_spec |> fit(sales ~ TV + radio + newspaper, data = ads)
mlr_fit <- lm_spec |>
  fit(sales ~ ., data = ads)

mlr_fit |>
  pluck("fit") |>
  summary()
```

same linear model specification as before.

data frame.

regress sales on every other column in data frame.

```
##
## Call:
## stats::lm(formula = sales ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0$
$\hat{\beta}_1$
$\hat{\beta}_2$
$\hat{\beta}_3$

2.2 Some Important Questions

When we perform multiple linear regression we are usually interested in answering a few important questions:

1. Is at least one of the predictors $X_1, \ldots, X_p$ useful in predicting response?
2. Do all predictors help to explain Y? or only a subset of predictors useful?
3. How well does the model fit the data?
4. Given a set of predictor values what response value should we predict? and how accurate is that prediction.

2.2.1 Is there a relationship between response and predictors?

We need to ask whether all of the regression slope coefficients are zero, which leads to the following hypothesis test.

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

$$H_a: \text{at least one } \beta_j \text{ is non-zero}: j = 1, \ldots, p$$

This hypothesis test is performed by computing the F-statistic

variance explained by model

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)} \sim F_{p, n-p-1}$$

variance unexplained

If F is large (much larger than 1), evidence against null H_0.

i.e. evidence there is some relationship.

2.2.2 Deciding on Important Variables

After we have computed the F-statistic and concluded that there is a relationship between predictor and response, it is natural to wonder

> Which predictors are related to the response?

We could look at the p-values on the individual coefficients, but if we have many variables this can lead to false discoveries.

Instead we could consider *variable selection*. We will revisit this in Ch. 6.

2.2.3 Model Fit

Two of the most common measures of model fit are the RSE and R^2. These quantities are computed and interpreted in the same way as for simple linear regression.

Be careful with using these alone, because R^2 will **always increase** as more variables are added to the model, even if it's just a small increase.

How to overfitting?
Use test data! Ch. 5.

```r
# model with TV, radio, and newspaper
mlr_fit |> pluck("fit") |> summary()
```

$\hat{\beta}$ $\hat{SE}(\hat{\beta})$ individual t-tests

```
##
## Call:
## stats::lm(formula = sales ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

p-values.

R^2

F test

$H_0: \beta_1 = \cdots \beta_p = 0$

$H_a: \beta_j \neq 0 \quad j \in 1, \ldots, p.$

```r
# model without newspaper
lm_spec |> fit(sales ~ TV + radio, data = ads) |>
  pluck("fit") |> summary()
```

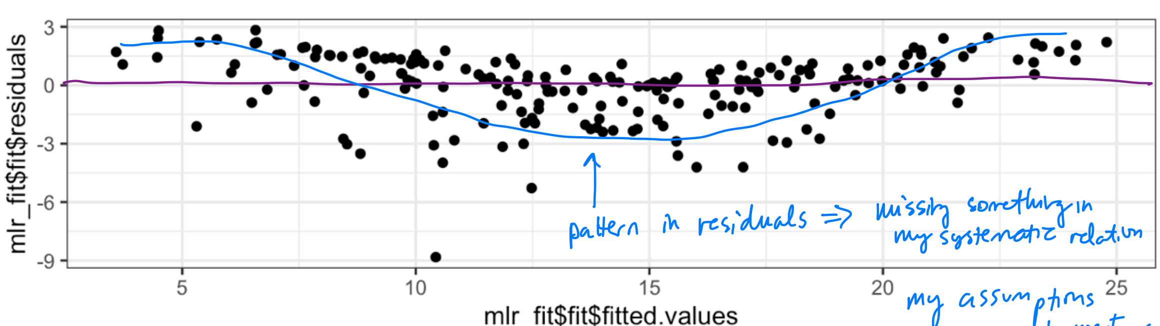
```
##
## Call:
## stats::lm(formula = sales ~ TV + radio, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7977 -0.8752  0.2422  1.1708  2.8328
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
## TV           0.04575    0.00139  32.909   <2e-16 ***
## radio        0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 197 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
## F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

R^2 barely decreased value when we took out newspaper ⇒ not contributing much.

It may also be useful to plot residuals to get a sense of the model fit.

$e_i = y_i - \hat{y}_i$

```r
ggplot() +
  geom_point(aes(mlr_fit$fit$fitted.values, mlr_fit$fit$residuals))
```



e

want: random noise around 0, no pattern.

my assumptions are not met about ε.

$\hat{y}$

can also check for Normality of ε using qq plot.

3 Other Considerations

3.1 Categorical Predictors

So far we have assumed all variables in our linear model are quantitiative.

what to do when x_j is categorical?

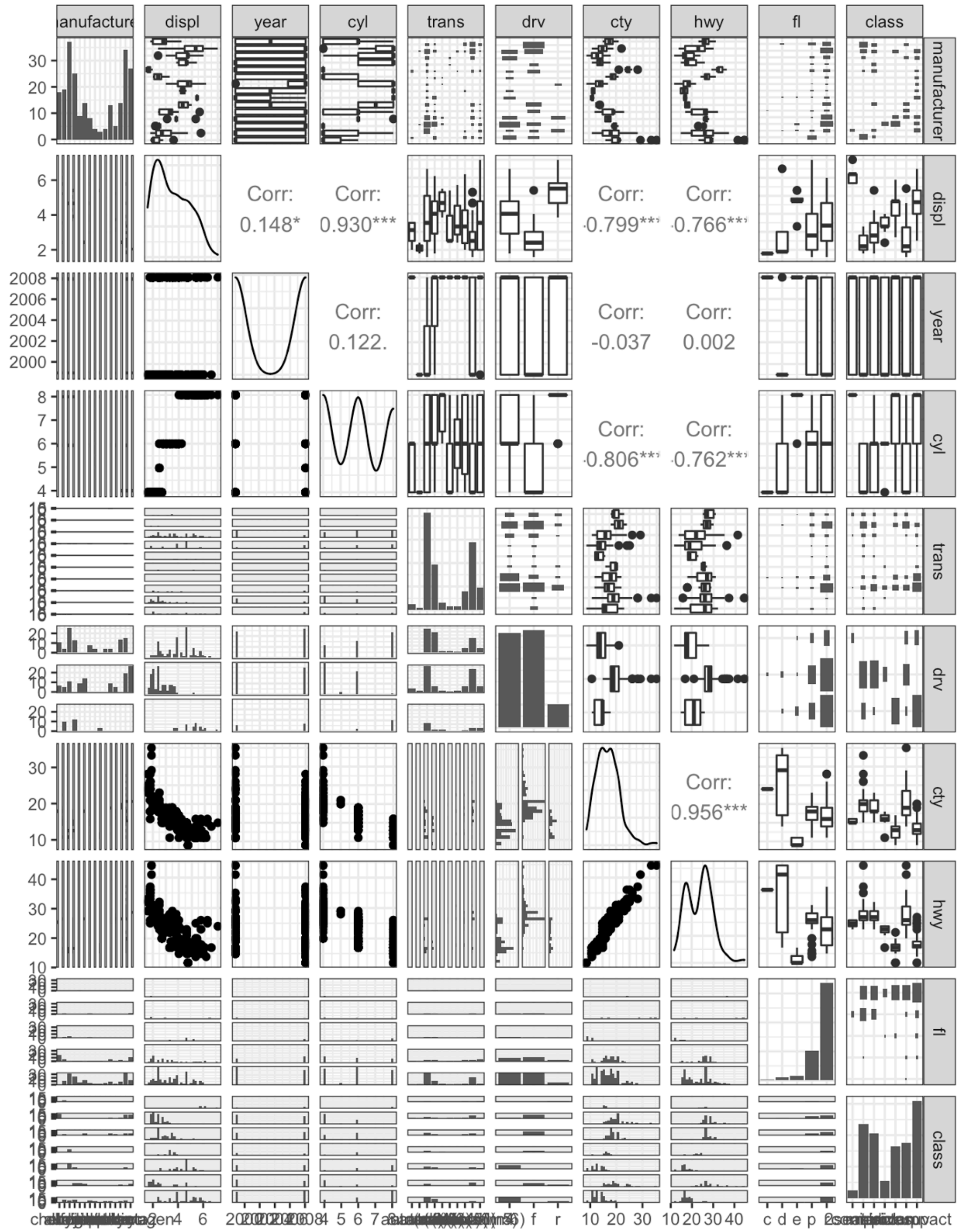
For example, consider building a model to predict highway gas mileage from the `mpg` data set.

```
head(mpg)
```

```
## # A tibble: 6 × 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy
fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int>
<chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29
p     compa…
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29
p     compa…
## 3 audi         a4      2    2008     4 manual(m6) f        20    31
p     compa…
## 4 audi         a4      2    2008     4 auto(av)   f        21    30
p     compa…
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26
p     compa…
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26
p     compa…
```

```
library(GGally)

mpg %>%
  select(-model) %>% # too many models
  ggpairs() # plot matrix
```

makes $\frac{p(p-1)}{2}$ plots to look at each pair of variables in a dataframe



looks at type of variable and chooses appropriate plot type

To incorporate these categorical variables into the model, we will need to introduce $k-1$ dummy variables, where $k =$ the number of levels in the variable, for each qualitative variable.

For example, for `drv`, we have 3 levels: `4`, `f`, and `r`. $k=3$

$$x_{i1} = \begin{cases} 1 & \text{if } i^{th} \text{ car is front WD} \\ 0 & \text{if } i^{th} \text{ car is not front WD} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if } i^{th} \text{ car is RWD} \\ 0 & \text{if } i^{th} \text{ car is not RWD.} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i^{th} \text{ car is FWD} \\ \beta_0 + \beta_2 + \varepsilon_i & \text{if } i^{th} \text{ car is RWD} \\ \beta_0 + \varepsilon_i & \text{if } i^{th} \text{ car is 4WD} \end{cases} \Longrightarrow$$

$\beta_0 =$ avg hwy mpg for 4WD cars.

$\beta_1 =$ difference in avg hwy mpg between FWD & 4WD cars.

$\beta_2 =$ difference in avg hwy mpg between RWD & 4WD cars.

```
lm_spec |>
  fit(hwy ~ displ + cty + drv, data = mpg) |>
  pluck("fit") |>
  summary()
```

quantitative — categorical.

```
##
## Call:
## stats::lm(formula = hwy ~ displ + cty + drv, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6499 -0.8764 -0.3001  0.9288  4.8632
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.42413    1.09313   3.132  0.00196 **
## displ       -0.20803    0.14439  -1.441  0.15100
## cty          1.15717    0.04213  27.466  < 2e-16 ***
## drvf         2.15785    0.27348   7.890 1.23e-13 ***
## drvr         2.35970    0.37013   6.375 9.95e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.49 on 229 degrees of freedom
## Multiple R-squared:  0.9384, Adjusted R-squared:  0.9374
## F-statistic: 872.7 on 4 and 229 DF,  p-value: < 2.2e-16
```

3.2 Extensions of the Model

The standard regression model provides interpretable results and works well in many problems. However it makes some very strong assumptions that may not always be reasonable.

* linear model
constant error variance
uncorrelated errors w/ predictors X
} captured enough predictors & relationship to response in model.

Additive Assumption

The additive assumption assumes that the effect of each predictor on the response is not affected by the value of the other predictors. What if we think the effect should depend on the value of another predictor?

interaction term

```
lm_spec |>
  fit(sales ~ TV + radio + TV*radio, data = ads) |>
  pluck("fit") |>
  summary()
```

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$

$$= \beta_0 + \underbrace{(\beta_1 + \beta_3 X_2)}_{\text{changes w/ respect to } X_2 \text{ values.}} X_1 + \beta_2 X_2 + \varepsilon$$

```
##
## Call:
## stats::lm(formula = sales ~ TV + radio + TV * radio, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3366 -0.4028  0.1831  0.5948  1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
## TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
## radio       2.886e-02  8.905e-03   3.241   0.0014 **
## TV:radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

$\hat{\beta}$

individual t-tests for significance.

$\hat{\beta}_3$ significantly different from 0.

$R^2 = .89$ without interaction, significant increase ⇒ better fit

* If we add interaction terms important to keep original variables, otherwise very confusing to interpret the results.

"an increase of \$1000 in radio advertising will be associated with an increase in sales of $(\hat{\beta}_2 + \hat{\beta}_3 TV) \times 1000$:
29 + 1.1 × TV units.

Alternatively:

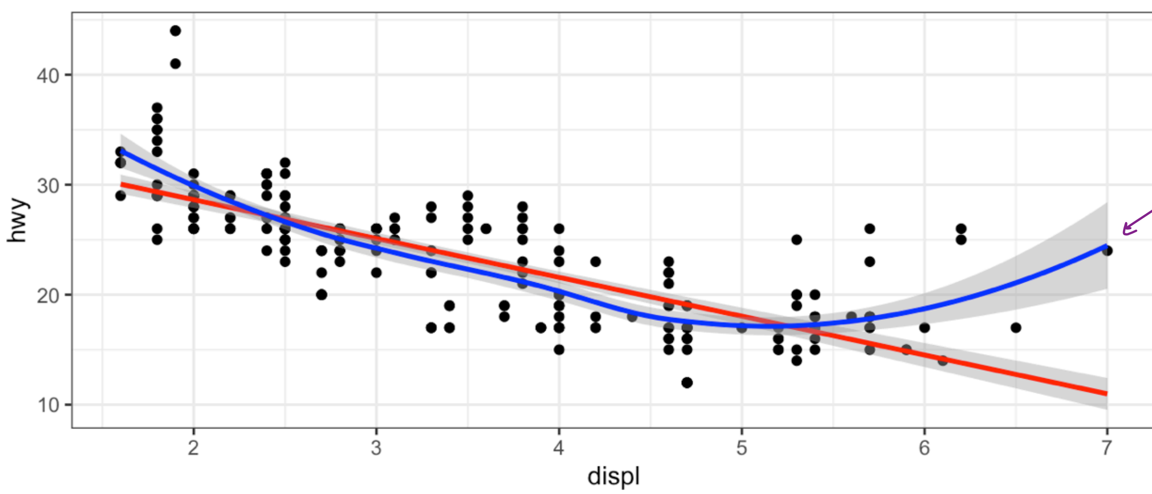
```
rec_spec_interact <- recipe(sales ~ TV + radio, data = ads) |>
  step_interact(~ TV:radio)

lm_wf_interact <- workflow() |>
  add_model(lm_spec) |>
  add_recipe(rec_spec_interact)

lm_wf_interact |> fit(ads)
```

```
## ══ Workflow [trained] ══════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: linear_reg()
##
## ── Preprocessor ────────────────────────────────────────────────────────────────
## 1 Recipe Step
##
## • step_interact()
##
## ── Model ───────────────────────────────────────────────────────────────────────
##
## Call:
## stats::lm(formula = ..y ~ ., data = data)
##
## Coefficients:
## (Intercept)           TV        radio    TV_x_radio
##    6.750220     0.019101     0.028860      0.001086
```

Linearity Assumption

The linear regression model assumes a linear relationship between response and predictors. In some cases, the true relationship may be non-linear.

```
ggplot(data = mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth(method = "lm", colour = "red") +
  geom_smooth(method = "loess", colour = "blue")
```



maybe not linear.

How to include nonlinear terms in the model?

(could alternatively mutate df)

"identity"

```
lm_spec |>
  fit(hwy ~ displ + I(displ^2), data = mpg) |>
  pluck("fit") |> summary()
```

```
##
## Call:
## stats::lm(formula = hwy ~ displ + I(displ^2), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6258 -2.1700 -0.7099  2.1768 13.1449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  49.2450     1.8576  26.510  < 2e-16 ***
## displ       -11.7602     1.0729 -10.961  < 2e-16 ***
## I(displ^2)    1.0954     0.1409   7.773 2.51e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.423 on 231 degrees of freedom
## Multiple R-squared:  0.6725, Adjusted R-squared:  0.6696
## F-statistic: 237.1 on 2 and 231 DF,  p-value: < 2.2e-16
```

significant

be careful throwing higher order (polynomial) terms → this will lead to overfitting & very bad predictions on the edges of your space.

3.3 Potential Problems

1. Non-linearity of response-predictor relationships

   diagnosis:
   plot residuals vs. fitted (vs. each predictor)
   see pattern.

   Solution:
   - add polynomial term. (for now)
   - transform predictor.

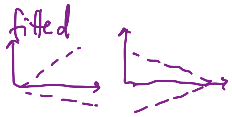
2. Correlation of error terms

   diagnosis
   understand how data is collected
   time series? spatial data?

   Solution:
   - use models formulated for correlated errors (not this class).
   - incorporate variables that capture the dependence in the systematic relationship

3. Non-constant variance of error terms

   diagnosis
   plot residuals vs. fitted
   see funnel pattern

   Solution:
   transform Y. Try $\log Y$ or $\sqrt{Y}$

4. Outliers

   diagnosis
   plot data

   Solution:
   Is your data wrong? i.e. error in collection? fix it.
   otherwise – maybe you are missing a predictor?

4 K-Nearest Neighbors

In Ch. 2 we discuss the differences between *parametric* and *nonparametric* methods. Linear regression is a parametric method because it assumes a linear functional form for $f(X)$.

easy to fit
easy to interpret
can perform hypothesis tests.

make strong assumptions, what if they are wrong?
- parametric method will perform poorly.

A simple and well-known non-parametric method for regression is called K-nearest neighbors regression (KNN regression).

\# of neighbors

Given a value for K and a prediction point x_0, KNN regression first identifies the K training observations that are closest to x_0 ($\mathcal{N}_0$). It then estimates $f(x_0)$ using the average of all the training responses in $\mathcal{N}_0$,

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_0} y_i$$

training data.

value to be predicted at.

```
set.seed(445) #reproducibility
```

↳ allows us to get same "random" numbers everytime.

fake data

```
## generate data
x <- rnorm(100, 4, 1) # pick some x values
y <- 0.5 + x + 2*x^2 + rnorm(100, 0, 2) # true relationship
df <- data.frame(x = x, y = y) # data frame of training data

knn_spec <- nearest_neighbor(mode = "regression")
for (k in seq(2, 10, by = 2)) {
  knn_spec |>
    fit(y ~ x, data = df, neighbors = k) |>
    augment(new_data = df) |>
    ggplot() +
    geom_point(aes(x, y)) +
    geom_line(aes(x, .pred), colour = "red") +
    ggtitle(paste("KNN, k = ", k)) +
    theme(text = element_text(size = 30)) -> p

  print(p)
}

lm_spec |>
  fit(y ~ x, df) |>
  augment(new_data = df) |>
  ggplot() +
    geom_point(aes(x, y)) +
```

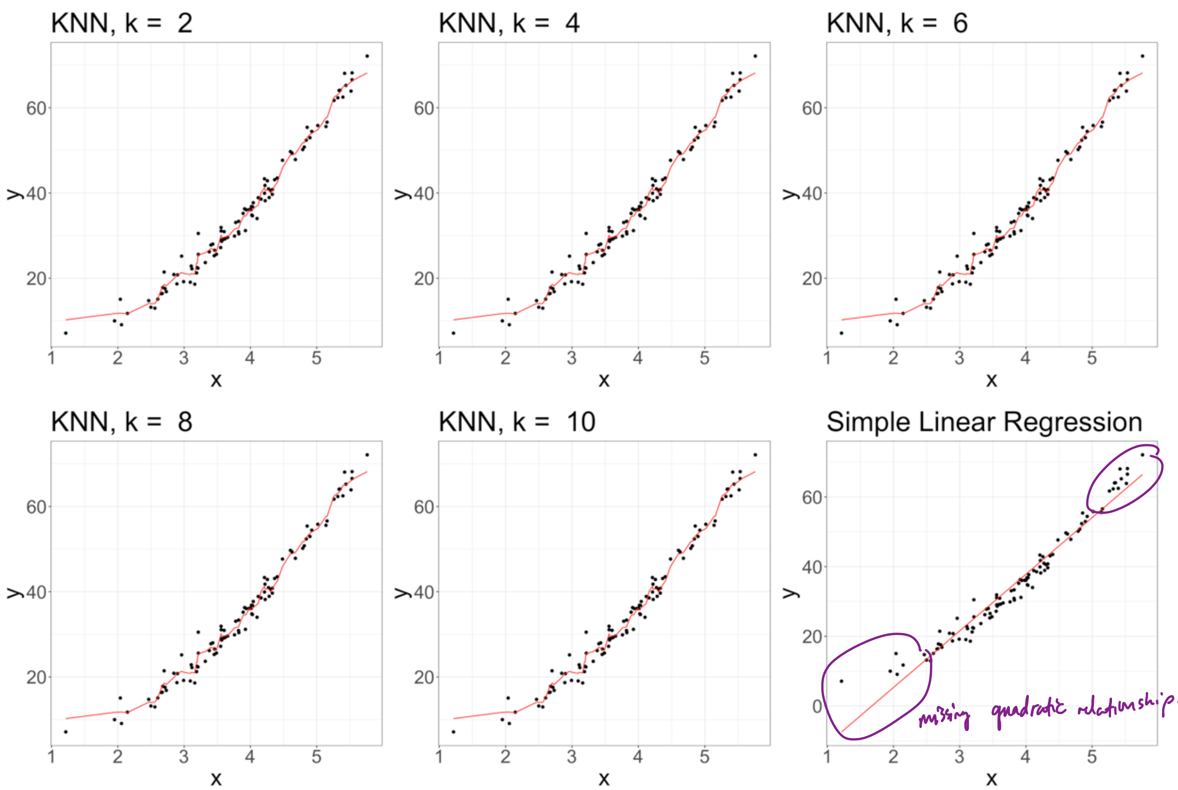
formula

k = 2, 4, 6, 8, 10

adds predictions (also residuals if applicable).

```
  geom_line(aes(x, .pred), colour = "red") +
  ggtitle("Simple Linear Regression") +
  theme(text = element_text(size = 30)) # slr plot
```

to fix?



as k ↑, KNN gets smoother