Multi Linear Regression Activity

Arjun Mannem

2023-11-28

# Load necessary packages  
library(tidyverse)

## Warning: package 'ggplot2' was built under R version 4.3.2

## Warning: package 'dplyr' was built under R version 4.3.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggthemes)  
library(flextable)

## Warning: package 'flextable' was built under R version 4.3.2

##   
## Attaching package: 'flextable'  
##   
## The following object is masked from 'package:purrr':  
##   
## compose

library(corrr)

## Warning: package 'corrr' was built under R version 4.3.2

library(ggfortify)  
library(broom)  
library(car)

## Warning: package 'car' was built under R version 4.3.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.2

##   
## Attaching package: 'car'  
##   
## The following object is masked from 'package:dplyr':  
##   
## recode  
##   
## The following object is masked from 'package:purrr':  
##   
## some

Set default theme settings for plots, and load a function to simplify table customization and creation using the code below.

# Set ggplot theme for visualizations  
theme\_set(ggthemes::theme\_few())  
  
# Set default ggplot2 colors  
my\_palette <- ggthemes::colorblind\_pal()(8)[c(4, 6:8, 2:3, 5, 1)]  
   
assign("scale\_colour\_discrete", function(..., values = my\_palette) scale\_colour\_manual(..., values = values), globalenv())  
  
assign("scale\_fill\_discrete", function(..., values = my\_palette) scale\_fill\_manual(..., values = values), globalenv())  
  
# Set options for flextables  
set\_flextable\_defaults(na\_str = "NA")  
  
# Load function for printing tables nicely  
source("https://raw.githubusercontent.com/dilernia/STA323/main/Functions/make\_flex.R")  
  
# Load function for visualizing VIF & GVIF values  
source("https://raw.githubusercontent.com/dilernia/STA323/main/Functions/vif\_plot.R")

## Multiple Linear Regression (MLR)

**What are the response and predictor variables for examples a. and b., and what are the types of each predictor variable (e.g., categorical or quantitative)?**

1. Predicting the selling price of houses using characteristics such as the size in square feet, number of bedrooms, number of bathrooms, and year it was built. 🏠

**Response Variable**: selling price of houses (y; quantitative)

**Predictor Variable**:size in square feet (x1; quantitative), number of bedrooms(x2; quantitative), number of bathrooms(x3; quantitative), and year it was built(x4; quantitative).

1. Forecasting the total sales of a product for a company using advertising expenses, the price of the product, the previous sales numbers for that product, and the time of year. 📈

**Response Variable**: total sales of a product (y; quantitative)

**Predictor Variable**: advertising expenses (x1; quantitative), the price of the product (x2; quantitative), the previous sales numbers for that product (x3; quantitative), and the time of year(x4; quantitative)

1. Exploring the relationship between employee salaries and factors such as education level, years of experience, job title, and gender. 💼 💸

**Response Variable**: employee salaries (y; quantitative)

**Predictor Variable**: education level (x1; categorical), years of experience (x2; quantitative), job title (x3; categorical), and gender (x4; categorical).

1. Constructing a model to determine what a typical blood-pressure should be for a patient using information such as the patient’s age, sex, and height. ⚕️🏥

**Response Variable**: blood-pressure (y; quantitative)

**Predictor Variable**: patient’s age (x1; quantitative), sex (x2; categorical), and height (x3; quantitative).

### Great Lakes Data Example

# Importing data from GitHub & clean column names  
greatLakes <- read\_csv("https://raw.githubusercontent.com/dilernia/STA323/main/Data/GreatLakesData.csv",  
 na = c("", "NA", ".")) %>%   
 janitor::clean\_names() %>%   
 dplyr::mutate(time\_edt = as.character(time\_edt))

## Rows: 129 Columns: 10  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): ID, Location, Owner, Direction, Lake, Date  
## dbl (3): Water (°F), Wind (kts/°), Waves (ft)  
## time (1): Time (EDT)  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

dataDictionary <- tibble(Variable = colnames(greatLakes),  
 Description = c("Buoy ID number",  
 "Location of buoy",  
 "Owner of buoy",  
 "Time of measurement",  
 "Water temperature in Fahrenheit",  
 "Wind speed in knots",  
 "Direction of wind",  
 "Peak wave height in feet",  
 "Great Lake",  
 "Date of measurement"),  
 Type = map\_chr(greatLakes, .f = function(x) {ifelse(is.numeric(x), "Quantitative", "Categorical")}))  
  
# Printing data dictionary  
dataDictionary %>%   
 make\_flex(caption = "Data dictionary for Great Lakes data.")

Table 1: Data dictionary for Great Lakes data.

| Variable | Description | Type |
| --- | --- | --- |
| id | Buoy ID number | Categorical |
| location | Location of buoy | Categorical |
| owner | Owner of buoy | Categorical |
| time\_edt | Time of measurement | Categorical |
| water\_f | Water temperature in Fahrenheit | Quantitative |
| wind\_kts | Wind speed in knots | Quantitative |
| direction | Direction of wind | Categorical |
| waves\_ft | Peak wave height in feet | Quantitative |
| lake | Great Lake | Categorical |
| date | Date of measurement | Categorical |

Let’s start with some exploratory data analysis to familiarize ourselves with the data. Below are 5 randomly selected rows from the data, a few scatter plots, and tables of summary statistics.

# Printing random 5 rows of data  
greatLakes %>%   
 drop\_na() %>%   
 slice\_sample(n = 5) %>%  
 make\_flex(caption = "Randomly selected rows from the Great Lakes data.")

Table 2: Randomly selected rows from the Great Lakes data.

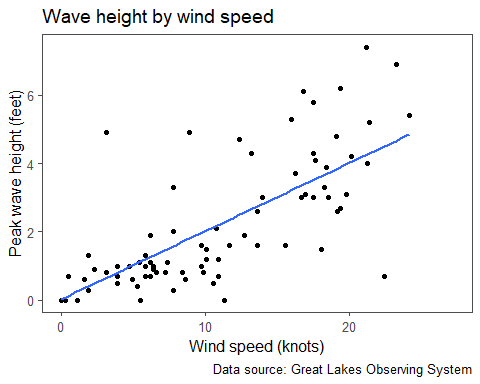
| id | location | owner | time\_edt | water\_f | wind\_kts | direction | waves\_ft | lake | date |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 45013 | Milwaukee Atwater Park | UW-Milwaukee | 09:30:00 | 63.60 | 21.20 | N | 7.40 | Lake Michigan | 9/22/2021 |
| 45135 | Prince Edward Pt | Env CA | 10:00:00 | 67.80 | 13.60 | SE | 2.60 | Lake Ontario | 9/22/2021 |
| 45132 | Northern Lake Erie | Env CA | 10:00:00 | 70.20 | 17.50 | NNE | 3.00 | Lake Erie | 9/22/2021 |
| 45132 | Northern Lake Erie | Env CA | 10:00:00 | 56.10 | 3.90 | ENE | 1.00 | Lake Erie | 11/9/2021 |
| 45159 | Northwest Ontario | Env CA | 10:00:00 | 68.50 | 7.80 | SSE | 2.00 | Lake Ontario | 9/22/2021 |

# Creating scatter plot of wave height by wind speed  
greatLakes %>%   
 ggplot(aes(x = wind\_kts,   
 y = waves\_ft)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(title = "Wave height by wind speed",  
 y = "Peak wave height (feet)",  
 x = "Wind speed (knots)",  
 caption = "Data source: Great Lakes Observing System")

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 43 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 43 rows containing missing values (`geom\_point()`).

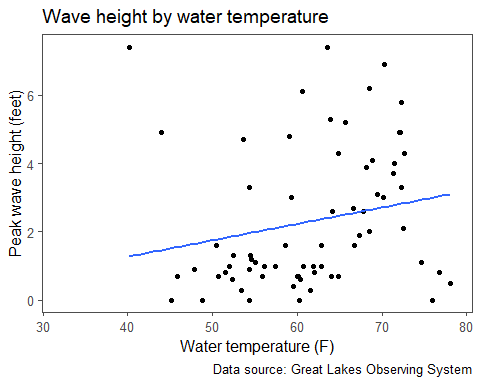


# Creating scatter plot of wave height by temperature  
greatLakes %>%   
 ggplot(aes(x = water\_f,   
 y = waves\_ft)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(title = "Wave height by water temperature",  
 y = "Peak wave height (feet)",  
 x = "Water temperature (F)",  
 caption = "Data source: Great Lakes Observing System")

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 57 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 57 rows containing missing values (`geom\_point()`).



# Calculating correlations  
corTable <- greatLakes %>% corrr::correlate(diagonal = 1)

## Non-numeric variables removed from input: `id`, `location`, `owner`, `time\_edt`, `direction`, `lake`, and `date`  
## Correlation computed with  
## • Method: 'pearson'  
## • Missing treated using: 'pairwise.complete.obs'

# Printing table of correlations  
corTable %>%   
 make\_flex(caption = "Table of pairwise correlations.")

Table 3: Table of pairwise correlations.

| term | water\_f | wind\_kts | waves\_ft |
| --- | --- | --- | --- |
| water\_f | 1.00 | 0.41 | 0.21 |
| wind\_kts | 0.41 | 1.00 | 0.70 |
| waves\_ft | 0.21 | 0.70 | 1.00 |

# Calculating descriptive statistics stratified by lake  
lakeStats <- greatLakes %>%   
 group\_by(lake) %>%   
 summarize(  
 N = n(),  
 Minimum = min(waves\_ft, na.rm = TRUE),  
 Q1 = quantile(waves\_ft, na.rm = TRUE, probs = 0.25),  
 Median = median(waves\_ft, na.rm = TRUE),  
 Q3 = quantile(waves\_ft, na.rm = TRUE, probs = 0.75),  
 Maximum = max(waves\_ft, na.rm = TRUE),  
 Mean = mean(waves\_ft, na.rm = TRUE),  
 Range = Maximum - Minimum,  
 SD = sd(waves\_ft, na.rm = TRUE),  
)  
  
# Printing table of statistics stratified by lake  
lakeStats %>%   
 make\_flex(caption = "Summary statistics for wave heights in feet stratified by lake.")

Table 4: Summary statistics for wave heights in feet stratified by lake.

| lake | N | Minimum | Q1 | Median | Q3 | Maximum | Mean | Range | SD |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lake Erie | 37 | 0.00 | 0.65 | 1.10 | 3.15 | 5.80 | 1.89 | 5.80 | 1.75 |
| Lake Huron | 13 | 0.70 | 1.40 | 3.05 | 3.77 | 6.20 | 2.99 | 5.50 | 1.97 |
| Lake Michigan | 32 | 0.00 | 0.85 | 2.60 | 4.55 | 7.40 | 2.87 | 7.40 | 2.28 |
| Lake Ontario | 19 | 0.30 | 0.55 | 1.45 | 1.90 | 2.60 | 1.35 | 2.30 | 0.92 |
| Lake Superior | 28 | 0.00 | 0.70 | 1.00 | 1.90 | 7.40 | 1.82 | 7.40 | 1.91 |

# Calculating descriptive statistics stratified by date  
dateStats <- greatLakes %>%   
 group\_by(date) %>%   
 summarize(  
 N = n(),  
 Minimum = min(waves\_ft, na.rm = TRUE),  
 Q1 = quantile(waves\_ft, na.rm = TRUE, probs = 0.25),  
 Median = median(waves\_ft, na.rm = TRUE),  
 Q3 = quantile(waves\_ft, na.rm = TRUE, probs = 0.75),  
 Maximum = max(waves\_ft, na.rm = TRUE),  
 Mean = mean(waves\_ft, na.rm = TRUE),  
 Range = Maximum - Minimum,  
 SD = sd(waves\_ft, na.rm = TRUE),  
)  
# Printing table of statistics stratified by date  
dateStats %>%   
 make\_flex(caption = "Summary statistics for wave heights in feet stratified by date.")

Table 5: Summary statistics for wave heights in feet stratified by date.

| date | N | Minimum | Q1 | Median | Q3 | Maximum | Mean | Range | SD |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11/9/2021 | 55 | 0.00 | 0.62 | 0.90 | 1.20 | 7.40 | 1.26 | 7.40 | 1.48 |
| 9/22/2021 | 74 | 0.00 | 1.00 | 3.00 | 4.30 | 7.40 | 2.87 | 7.40 | 2.01 |

**Based on the scatter plot and correlations, what can we say about the relationship between the wind speed in knots and the observed wave heights in feet?**

There is a positive and linear relationship between peak wave heights in feet and the wind speeds in knots (r = 0.41)

**Based on the scatter plot and correlations, what can we say about the relationship between the water temperature in Fahrenheit and the observed wave heights in feet?**

There is no clear relationship between the peak wave heights in feet and the temperature in Fahrenheit (r = 0.21).

**Provide a statement of our assumed / theoretical multiple linear regression model using the wave height (feet) as our response and the surface temperature of the water (f) and the wind speed (mph) as our predictors.**

The theoretical or assumed MLR model is

y= β0 + β1x1 + β2x2 +𝜀,

where:

y represents the peak wave height in feet

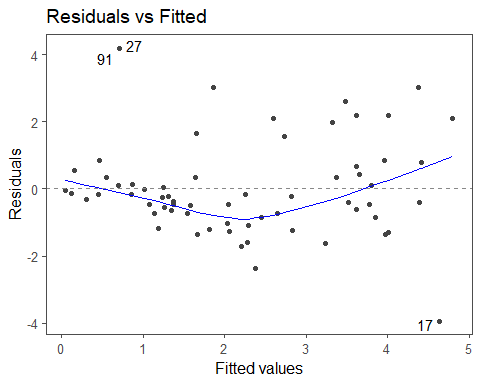
x1 represents the wind speed in knots

x2 represents the temperature in Fahrenheit

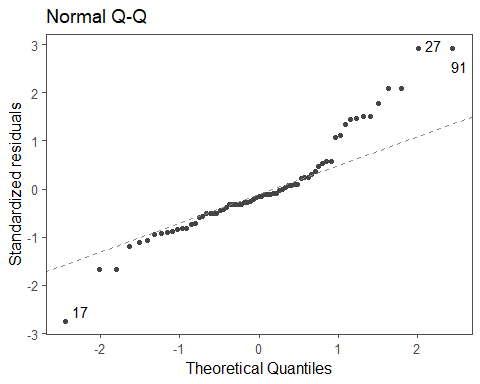
𝜀 is the independent and normally distributed error term

Next let’s take a look at some diagnostic plots to see how model assumptions look.

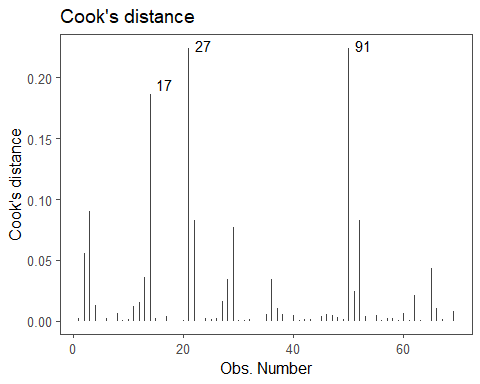
# Fitting two-predictor multiple linear regression model  
mlrModel <- lm(waves\_ft ~ wind\_kts + water\_f,  
 data = greatLakes)  
  
# Creating diagnostic plots  
  
# Residual by fitted value plot  
autoplot(mlrModel, which = 1, label.repel = TRUE)@plots[[1]]



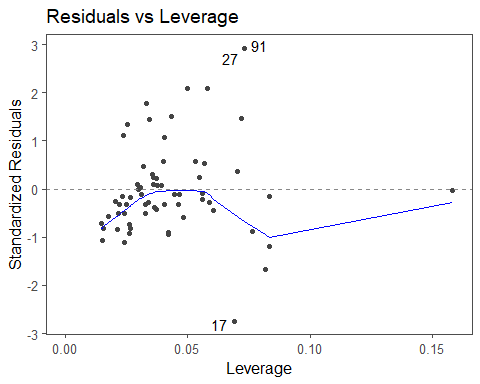
# Quantile-quantile (QQ) plot  
autoplot(mlrModel, which = 2, label.repel = TRUE)@plots[[1]]



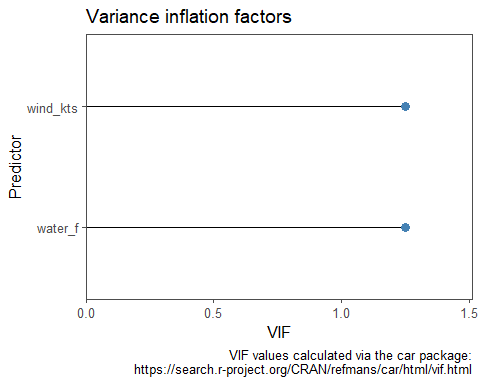
# Cook's Distance plot  
autoplot(mlrModel, which = 4, label.repel = TRUE)@plots[[1]]



# Standardized residuals by leverage  
autoplot(mlrModel, which = 5, label.repel = TRUE)@plots[[1]]



# Variance inflation values  
vif\_plot(mlrModel)



**Indicate if each assumption for fitting the multiple linear regression model is met, providing specific evidence to support your conclusions from the obtained output.**

**Random Sample**: We must assume that we have a random sample.

**Normality**: *the residuals must be normally distributed.* *Checked using a histogram or quantile-quantile (QQ) plot of the residuals. A fairly bell-shaped distribution in the histogram supports normality, and the points not straying far from the diagonal line on the QQ plot supports normality.*

Since there are several points that stray far from the diagonal line on the QQ plot, the normality assumption is violated.

**Homoskedasticity**: *the residuals must have constant variance / spread across all predicted values.* *Checked using the residual by predicted value plot. A fairly constant vertical spread / range of the residuals across all predicted values supports homoskedasticity.*

There is a slight outward fanning pattern in the residual by fitted value plot, suggesting that the homoskedasticity is questionable here.

**Independence**: *the residuals must be independent of the predicted values.* *Checked using the residual by predicted value plot. No apparent pattern in the relationship between the residuals and predicted values supports the independence assumption.*

There is no clear pattern in the residual by fitted value plot, indicating that the independent errors assumption is met.

**Linearity**: *there must be a linear relationship between each predictor and Y, our response variable.* *Checked using the residual by predicted value plot or a scatter plot between X and Y. Having no systematic or identifiable pattern in the residual by predicted value plot supports the linearity assumption. A linear or straight-line pattern in the scatter plot between each predictor and Y also supports the linearity assumption.*

There is a positive and linear relationship between peak wave heights in feet and the wind speeds in knots (r = 0.41)

However, from scatter plot, here is no clear relationship between the peak wave heights in feet and the temperature in Fahrenheit (r = 0.21).

Neither predictor exhibits non-linear relationships.

**No multicollinearity**: *the predictor variables should not be highly correlated with one another. This assumption is only applicable for multiple linear regression.*

Since all VIF values are less than 5, the assumption of no multicollinearity being present is met.

**Overall, not all assumptions are met, but we proceed for the sake of the problem.**

#### Interpreting model estimates

# Printing model output  
mlrModel %>%   
 broom::tidy() %>%   
 make\_flex(caption = "Two predictor MLR model estimates",  
 ndigits = 3)

Table 6: Two predictor MLR model estimates

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.136 | 1.325 | 0.103 | 0.919 |
| wind\_kts | 0.204 | 0.031 | 6.598 | 8.36e-09 |
| water\_f | -0.001 | 0.023 | -0.050 | 0.960 |

mlrModel %>%   
 broom::glance() %>%   
 make\_flex(caption = "Two predictor MLR model output")

Table 7: Two predictor MLR model output

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.45 | 0.43 | 1.49 | 26.98 | 2.7e-09 | 2.00 | -123.88 | 255.76 | 264.70 | 146.49 | 66 | 69 |

**Using the obtained output, provide a statement of the estimated regression equation.**

The estimated MLR model is

ŷ = 0.136 + 0.204 \* x1 - 0.001 \* x2,

where:

ŷ represents the estimated peak wave height in feet

x1 represents the wind speed in knots

x2 represents the temperature in Fahrenheit

**Interpret the estimated slopes in context.**

Since βHat1 = 0.204, for every 1 knot increase in wind speed, we expect the peak wave height in feet to increase by 0.204 feet, holding the temperature constant.

Since βHat2 = -0.001, for every 1 degree Fahrenheit increase in temperature, we expect the peak wave height in feet to decrease by 0.001 feet, holding the wind speed constant.

**Interpret the estimated intercept in context. Is this sensible in this context?**

Since βHat0 = 0.136, the expected the peak wave height in feet on a day that is 0 degrees Fahrenheit with no wind is 0.136 feet.

**Practical issue**: This is not sensible since 0 degrees Fahrenheit would almost certainly mean the Great Lake is frozen over.

**Extrapolation issue**: Since a temperature of 0F is the outside the scope of this data set, that would be extrapolation.

**For a 70 degree F day with no wind (0 knots), what is the estimated peak wave height?**

ŷ = 0.136 + 0.204 \* x1 - 0.001 \* x2

ŷ = 0.136 + 0.204 \* 0 - 0.001 \* 70

ŷ = 0.066 feet

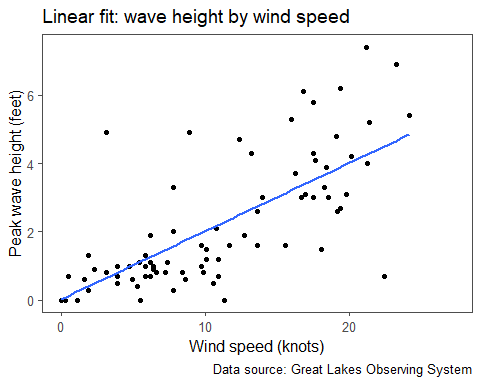
### Functions of Predictors

# Creating scatter plot of wave height by wind speed with a linear fit  
greatLakes %>%   
 ggplot(aes(x = wind\_kts,   
 y = waves\_ft)) +   
 geom\_point() +   
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(title = "Linear fit: wave height by wind speed",  
 y = "Peak wave height (feet)",  
 x = "Wind speed (knots)",  
 caption = "Data source: Great Lakes Observing System")

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: Removed 43 rows containing non-finite values (`stat\_smooth()`).

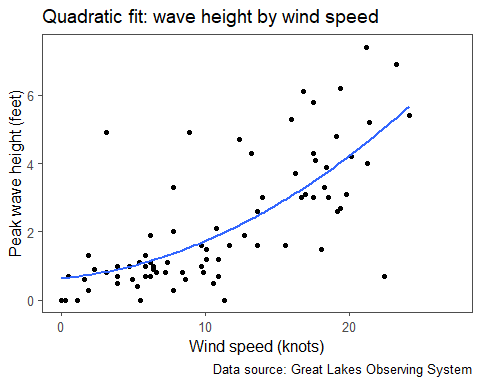
## Warning: Removed 43 rows containing missing values (`geom\_point()`).



# Creating scatter plot of wave height by wind speed with quadratic fit  
greatLakes %>%   
 ggplot(aes(x = wind\_kts,   
 y = waves\_ft)) +   
 geom\_point() +   
stat\_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +  
 labs(title = "Quadratic fit: wave height by wind speed",  
 y = "Peak wave height (feet)",  
 x = "Wind speed (knots)",  
 caption = "Data source: Great Lakes Observing System")

## Warning: Removed 43 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 43 rows containing missing values (`geom\_point()`).



# Fitting a quadratic regression model  
mlrModel2 <- lm(waves\_ft ~ wind\_kts + I(wind\_kts^2),  
 data = greatLakes)  
  
# Printing quadratic model output  
mlrModel2 %>%   
 broom::tidy() %>%   
 make\_flex(caption = "Quadratic MLR model estimates",  
 ndigits = 3)

Table 8: Quadratic MLR model estimates

| term | estimate | std.error | statistic | p.value |
| --- | --- | --- | --- | --- |
| (Intercept) | 0.635 | 0.423 | 1.504 | 0.136 |
| wind\_kts | 0.040 | 0.088 | 0.456 | 0.650 |
| I(wind\_kts^2) | 0.007 | 0.004 | 1.870 | 0.065 |

mlrModel2 %>%   
 broom::glance() %>%   
 make\_flex(caption = "Quadratic MLR model output")

Table 9: Quadratic MLR model output

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.50 | 0.49 | 1.34 | 42.28 | 2.2e-13 | 2.00 | -145.81 | 299.63 | 309.44 | 149.53 | 83 | 86 |

**Based on the quadratic model, what is the estimated peak wave height for a day no wind (0 knots)?**

Intercept value (0.635).

yHat = βHat0 (since xHat = 0 knots) = 0.635.

### Assessing Individual Predictors

**Based on the obtained output, should we include the quadratic term in our model? Why or why not?**

### Assessing Model Fit

# Fitting single predictor model  
mlrModel1 <- lm(waves\_ft ~ wind\_kts,  
 data = greatLakes)  
  
# Model output for single predictor model  
mlrModel1 %>%   
 broom::glance() %>%   
 make\_flex(caption = "Output for SLR model (wind speed).")

Table 10: Output for SLR model (wind speed).

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.48 | 0.48 | 1.36 | 78.73 | 1.1e-13 | 1.00 | -147.59 | 301.17 | 308.54 | 155.83 | 84 | 86 |

# Fitting two-predictor multiple linear regression model  
mlrModel <- lm(waves\_ft ~ wind\_kts + water\_f,  
 data = greatLakes)  
  
# Model output for two-predictor model  
mlrModel %>%   
 broom::glance() %>%   
 make\_flex(caption = "Two-predictor MLR model output")

Table 11: Two-predictor MLR model output

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.45 | 0.43 | 1.49 | 26.98 | 2.7e-09 | 2.00 | -123.88 | 255.76 | 264.70 | 146.49 | 66 | 69 |

# Fitting a quadratic regression model  
mlrModel2 <- lm(waves\_ft ~ wind\_kts + I(wind\_kts^2),  
 data = greatLakes)  
  
# Model output for quadratic model  
mlrModel2 %>%   
 broom::glance() %>%   
 make\_flex(caption = "Quadratic MLR model output")

Table 12: Quadratic MLR model output

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.50 | 0.49 | 1.34 | 42.28 | 2.2e-13 | 2.00 | -145.81 | 299.63 | 309.44 | 149.53 | 83 | 86 |

# Fitting two predictor quadratic model  
mlrModel22 <- lm(waves\_ft ~ water\_f + I(wind\_kts^2),  
 data = greatLakes)  
  
# Model output for two predictor quadratic model  
mlrModel22 %>%   
 broom::glance() %>%   
 make\_flex(caption = "Two predictor quadratic MLR model estimates")

Table 13: Two predictor quadratic MLR model estimates

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | AIC | BIC | deviance | df.residual | nobs |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.47 | 0.45 | 1.46 | 29.38 | 7.5e-10 | 2.00 | -122.53 | 253.06 | 262.00 | 140.87 | 66 | 69 |

**Based on the AIC and BIC values, which model should we use for making predictions?**

Since the two predictor quadratic model with water temperature and wind speed squared has the lowest AIC and lowest BIC values, this is the preferred model for making predictions.

Based on the adjusted r^2 value though, the third model is best since it has the highest largest adjusted r^2 value.