# Math 300 Lesson 35 Notes

Regression Assumptions

YOUR NAME HERE

July, 2022

## Contents

### Objectives

1. Explain and test the assumptions for inference in a linear regression model.

### Reading

Chapter 10.3

### Lesson

Work through the learning check LC 10.1.

- **LINE** is a mnemonic to help remember the assumptions. Linearity, Independence, Normality, and Equality of variance.
- Residuals are the key element used to check assumptions. Note that independence is difficult to check unless there is a time element to the data collection which often happens in experiments.

---

#### Libraries

```
library(tidyverse)
library(infer)
library(moderndive)
```

## Problem

Let's briefly review the ideas from the reading before working the learning check.

### Data and model

Let's get the data and model again.

```
evals_ch5 <- evals %>%
  select(ID, score, bty_avg, age)
glimpse(evals_ch5)
```

```
## Rows: 463
## Columns: 4
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
## $ score   <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3, 2.8, 4.1, 3.4, 4.5, 3.8, 4.5, 4.~
## $ bty_avg <dbl> 5.000, 5.000, 5.000, 5.000, 3.000, 3.000, 3.000, 3.333, 3.333,~
## $ age     <int> 36, 36, 36, 36, 59, 59, 59, 51, 51, 40, 40, 40, 40, 40, 40, 40~
```

```
score_model <- lm(score ~ bty_avg, data = evals_ch5)
```

```
score_regression_points <- get_regression_points(score_model)
```

```
head(score_regression_points)
```

```
## # A tibble: 6 x 5
##      ID score bty_avg score_hat residual
##   <int> <dbl>   <dbl>     <dbl>    <dbl>
## 1     1   4.7       5      4.21    0.486
## 2     2   4.1       5      4.21   -0.114
## 3     3   3.9       5      4.21   -0.314
## 4     4   4.8       5      4.21    0.586
## 5     5   4.6       3      4.08    0.52
## 6     6   4.3       3      4.08    0.22
```

### Linearity

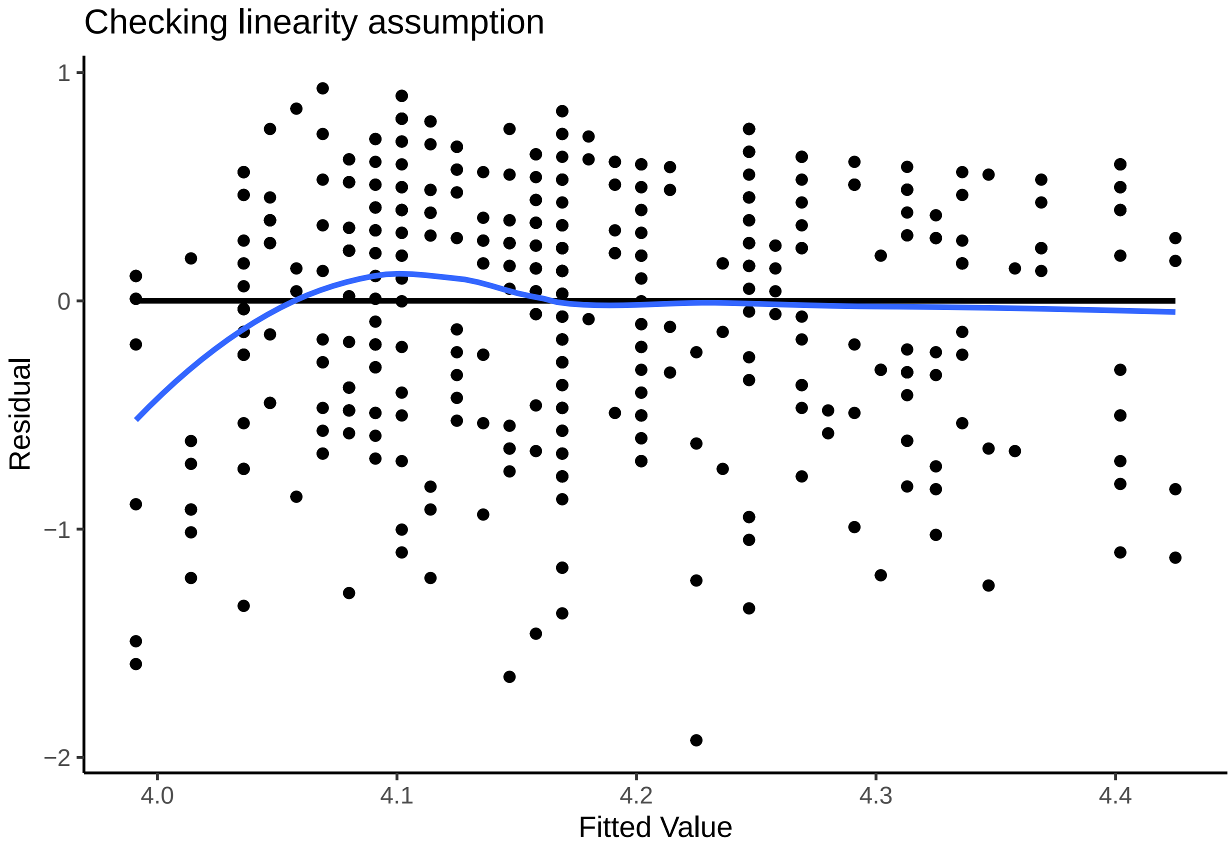
When we have a simple linear regression model, we can check linearity using a scatterplot of the explanatory and response variables. For multiple regression, we need to plot fitted values and residuals. We will do both here for our problem.

```
# Complete the code
# ggplot(evals_ch5,
#        aes(x = ________, y = _______)) +
#   geom_point() +
#   labs(x = "Beauty Score",
#        y = "Teaching Score",
#        title = "Relationship between teaching and beauty scores") +
#   geom_smooth(method = "lm", se = FALSE, color="black") +
#   geom_smooth(method="loess", se=FALSE) +
#   theme_classic()
```

The `loess` function plots a smoother through the data, giving us a good indication as to the true linearity of the observed data. In this case, the linearity assumption is not bad, but there could be a small issue at the lowest beauty scores.

We could also plot the residuals to assess linearity:

```
ggplot(score_regression_points,
       aes(x = score_hat, y = residual)) +
  geom_point() +
  labs(x = "Fitted Value",
       y = "Residual",
       title = "Checking linearity assumption") +
  geom_smooth(method = "lm", se = FALSE, color="black") +
  geom_smooth(method="loess", se=FALSE) +
  theme_classic()
```



We see a similar result; we prefer to use residual plots to check linearity because they allow us to generalize to multiple predictors.

### Normality

We will build a histogram to check this assumption.

```
# Complete the code
# ggplot(________________, aes(x = _______)) +
#   geom_histogram(binwidth = 0.25, fill = "cyan", color="black") +
#   labs(x = "Residual") +
#   theme_classic()
```

There is a skew to the left.

### Equality of Variance

For simple linear regression, we could use the original scatterplot of the data. However, we generally like to use the residuals vs fitted plot because we can extend to multiple regression. We want an equal spread around the horizontal line centered at zero.

```
# Complete the code
# ggplot(____________, aes(x = __________, y = _______)) +
#   geom_point() +
#   labs(x = "__________", y = "___________") +
#   geom_hline(yintercept = 0, col = "blue", size = 1)
```

---

## Learning Check 10.1 (Objective 1)

**(LC 10.1)** Continuing with our regression using `age` as the explanatory variable and teaching `score` as the outcome variable.

- Use the `get_regression_points()` function to get the observed values, fitted values, and residuals for all instructors.

```
# Complete the code to get the regression model and residuals
```

- Perform a residual analysis and look for any systematic patterns in the residuals. Ideally, there should be little to no pattern but comment on what you find here.

The first condition is that the relationship between the outcome variable $y$ and the explanatory variable $x$ must be **L**inear.

```
# Complete the code
set.seed(76)
# evals_ch5 %>%
#   ggplot(aes(x = ________, y = ______)) +
#   geom_point() +
#   labs(x = "_______", y = "_____________") +
#   geom_smooth(method = "lm", se = FALSE, color="black") +
#   geom_smooth(method = "loess", se=FALSE, color="blue") +
#   expand_limits(y = 10)
```

The second condition is that the residuals must be **I**ndependent. In other words, the different observations in our data must be independent of one another. As explained in the reading, "we say there exists *dependence* between observations".

The third condition is that the residuals should follow a **N**ormal distribution.

```
# Complete the code
# ggplot(regression_points, aes(x = ______________)) +
#   geom_histogram(binwidth = 0.25, color = "black", fill = "cyan") +
#   labs(x = "__________") +
#   theme_classic()
```

The fourth and final condition is that the residuals should exhibit **E**qual variance across all values of the explanatory variable $x$. In other words, the value and spread of the residuals should not depend on the value of the explanatory variable $x$.

```
# Complete the code
# ggplot(regression_points, aes(x = _________, y = __________)) +
#   geom_point() +
#   labs(x = "___________", y = "_______________") +
#   geom_hline(yintercept = 0, col = "blue", size = 1)
```

---

## Documenting software

- File creation date: 2022-07-15
- R version 4.1.3 (2022-03-10)
- `tidyverse` package version: 1.3.1
- `moderndive` package version: 0.5.4
- `infer` package version: 1.0.2