IS5 in R: Inferences for Regression (Chapter 20)

Nicholas Horton (nhorton@amherst.edu)

December 13, 2020

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fifth Edition of *Intro Stats* (2018) by De Veaux, Velleman, and Bock. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at http://nhorton.people.amherst.edu/is5.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (https://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

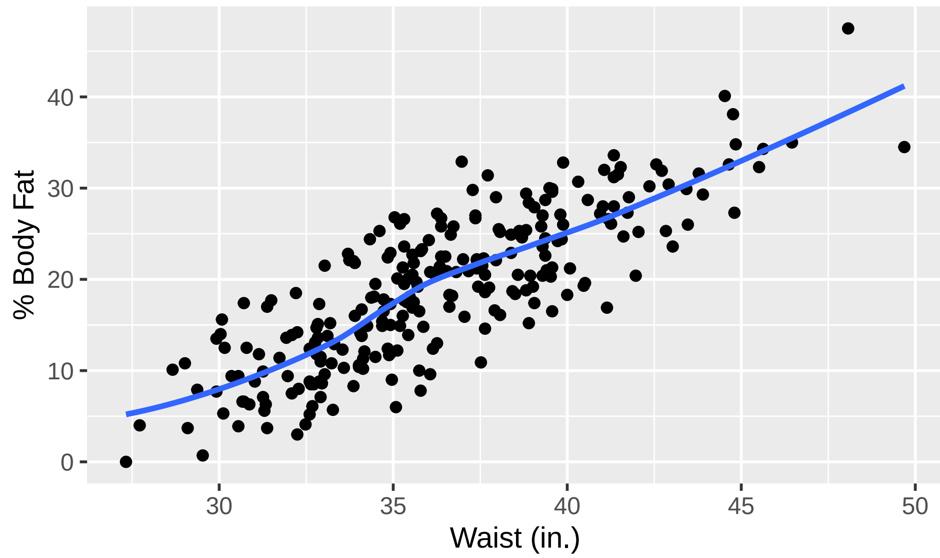
Chapter 20: Inferences for Regression

```
library(mosaic)
library(readr)
library(janitor)
BodyFat <- read_csv("http://nhorton.people.amherst.edu/is5/data/Bodyfat.csv") %>%
  janitor::clean_names()
```

By default, `read_csv()` prints the variable names. These messages have been suppressed using the `message=FALSE` code chunk option to save space and improve readability. Here we use the `clean_names()` function from the `janitor` package to sanitize the names of the columns (which would otherwise contain special characters or whitespace).

```
# Figure 20.1, page 642
gf_point(pct_bf ~ waist, data = BodyFat) %>%
  gf_smooth() %>% # to show linear relationship
  gf_labs(x = "Waist (in.)", y = "% Body Fat")
```

```
## `geom_smooth()` using method = 'loess'
```

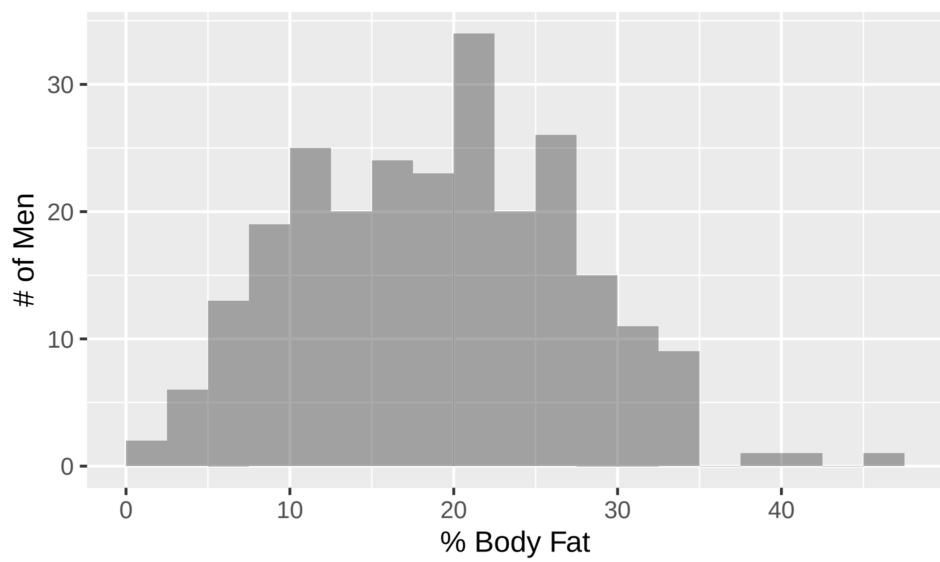


Section 20.1: The Regression Model

```
lm(pct_bf ~ waist, data = BodyFat)
```

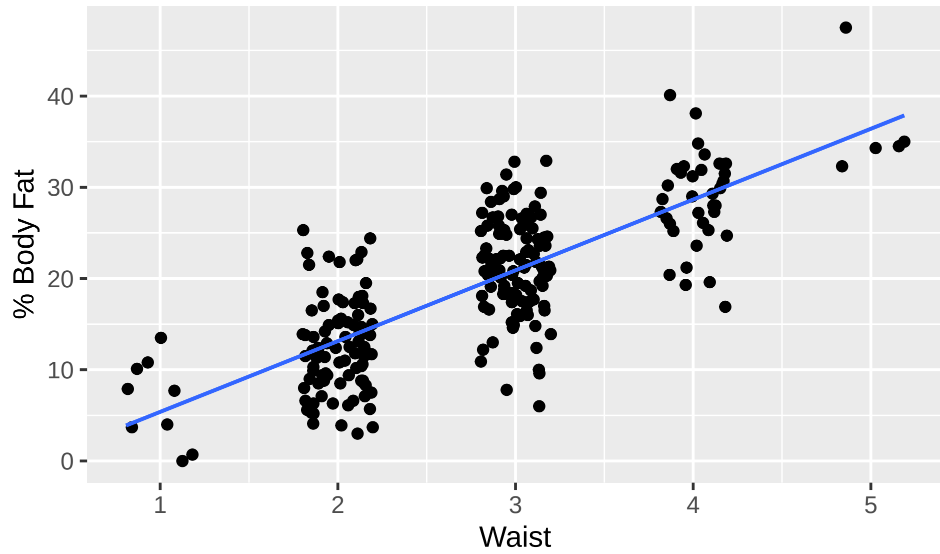
```
##
## Call:
## lm(formula = pct_bf ~ waist, data = BodyFat)
##
## Coefficients:
## (Intercept)        waist
##      -42.73         1.70
```

```
# Figure 20.2
gf_histogram(~pct_bf, data = BodyFat, binwidth = 2.5, center = 1.25) %>%
  gf_labs(x = "% Body Fat", y = "# of Men")
```




```
# Figure 20.3 (reinterpreted with points)
BodyFat <- BodyFat %>%
```

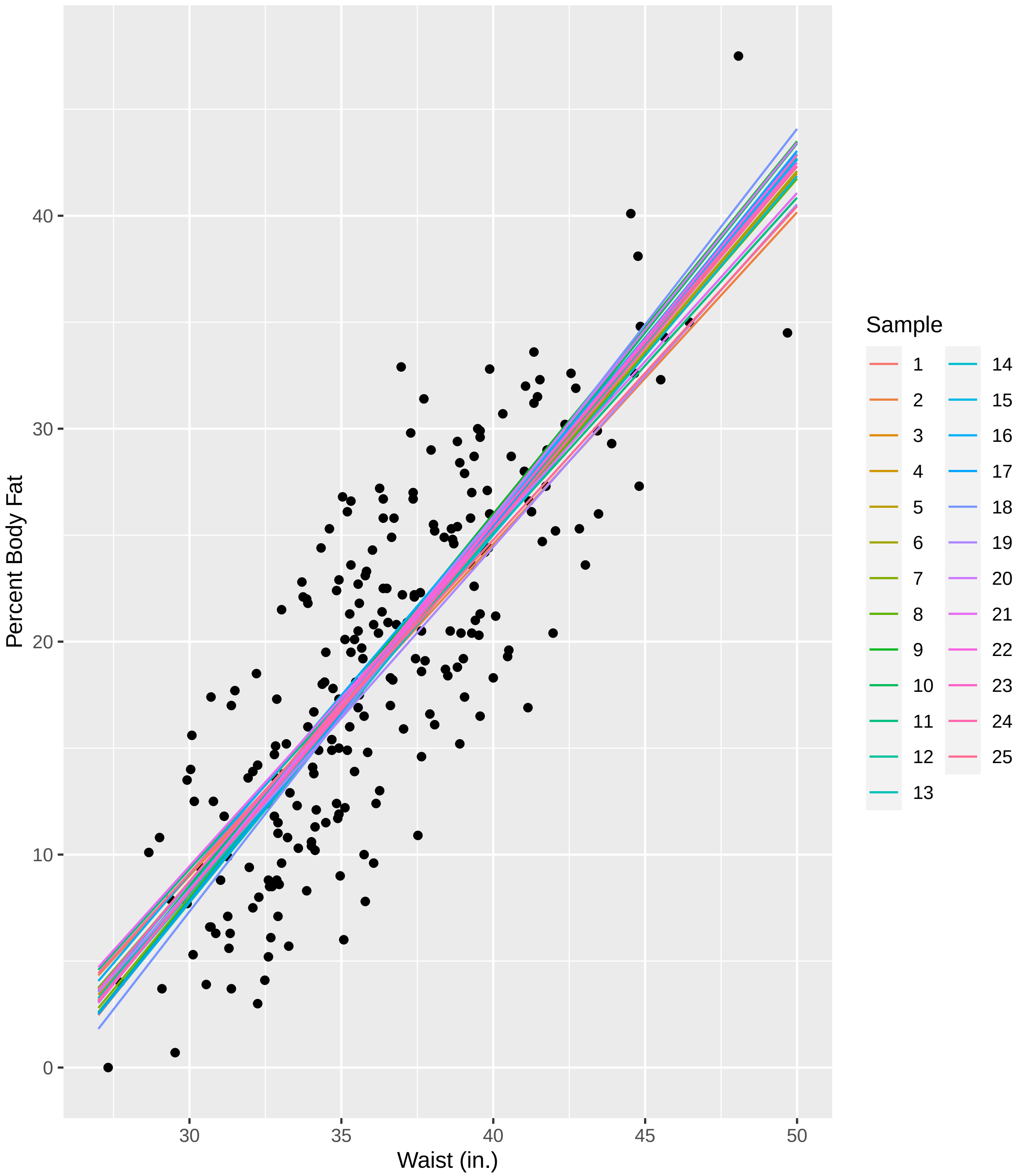
```
  mutate(roundedwaist = cut(waist, breaks = c(0, 30, 35, 40, 45, Inf), labels = c(1:5)))
gf_point(pct_bf ~ jitter(as.numeric(roundedwaist)), data = BodyFat) %>%
  gf_lm() %>%
  gf_labs(y = "% Body Fat", x = "Waist")
```



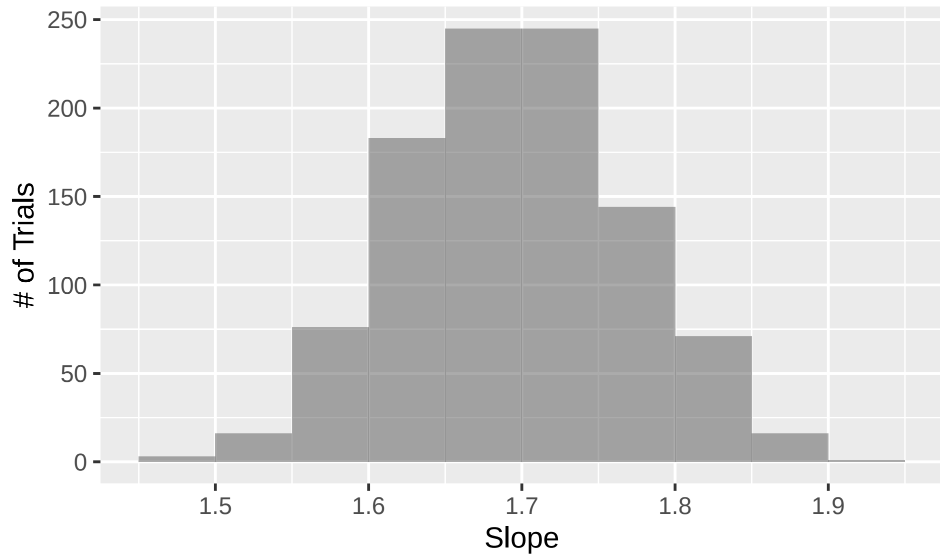

```
numsamp <- 25 # It's too messy to do any more than 25
slopesdata <- do(numsamp) * lm(pct_bf ~ waist, data = resample(BodyFat))
```

Random Matters: Slopes Vary For more information about `resample()`, refer to the resample vignette in mosaic.

```
slopesdata <- slopesdata %>%
  mutate(at27 = Intercept + waist * 27, at50 = Intercept + waist * 50, color = as.factor(1:25))
# Figure 20.4, page 644
gf_point(pct_bf ~ waist, data = BodyFat) %>%
  gf_segment(at27 + at50 ~ 27 + 50, data = slopesdata, color = ~color) %>%
  gf_labs(color = "Sample", x = "Waist (in.)", y = "Percent Body Fat")
```



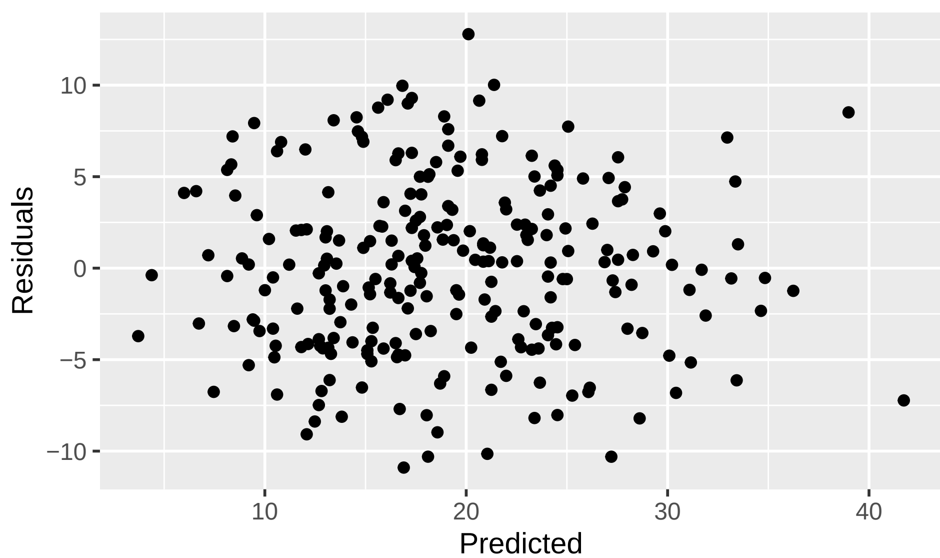

```
numsamp <- 1000 # To see the shape of the histogram
slopesdata <- do(numsamp) * lm(pct_bf ~ waist, data = resample(BodyFat))
# Figure 20.5
gf_histogram(~waist, data = slopesdata, binwidth = .05, center = .025) %>%
  gf_labs(x = "Slope", y = "# of Trials")
```



For the histogram, we use 1,000 trials.

Section 20.2: Assumptions and Conditions

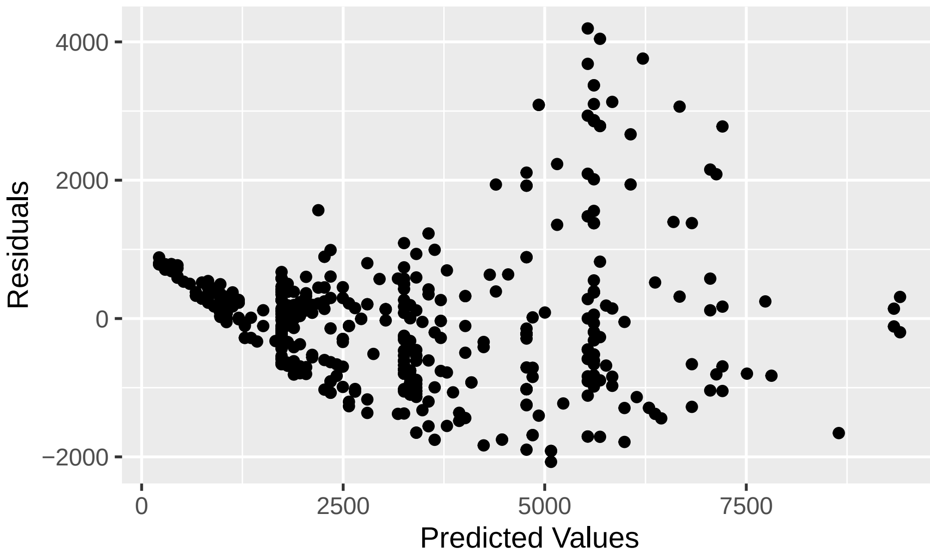
```
# Figure 20.6 is the same as Figure 20.1
# Figure 20.7 (page 645)
bodyfatlm <- lm(pct_bf ~ waist, data = BodyFat)
gf_point(resid(bodyfatlm) ~ fitted(bodyfatlm)) %>%
  gf_labs(x = "Predicted", y = "Residuals")
```



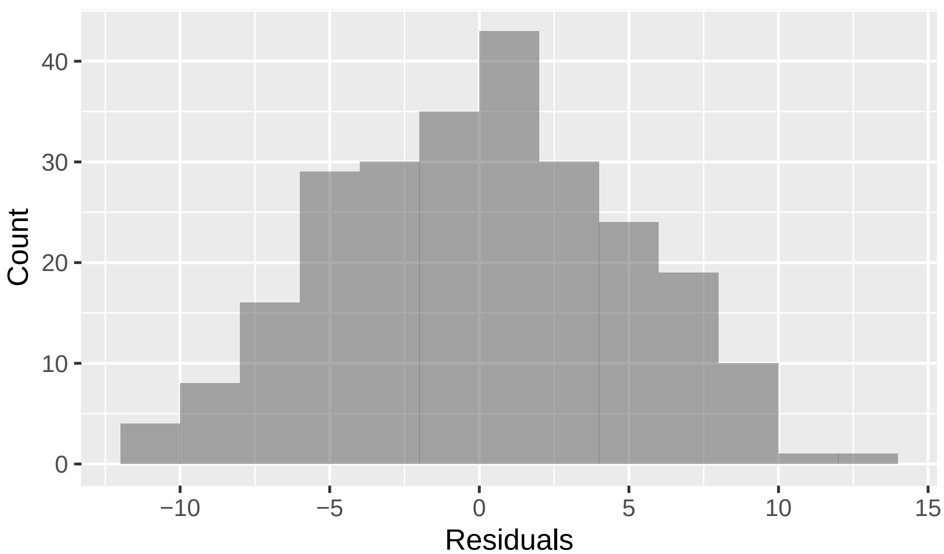

```
Diamonds <- read_csv("http://nhorton.people.amherst.edu/is5/data/Diamonds.csv") %>%
  janitor::clean_names()
```

Here we fit `price` by `carat_size` for diamonds with the color `E`.

```
diamondlm <- lm(price ~ carat_size, data = filter(Diamonds, color == "E"))
# Figure 20.8, page 646
gf_point(resid(diamondlm) ~ fitted(diamondlm)) %>%
  gf_labs(x = "Predicted Values", y = "Residuals")
```



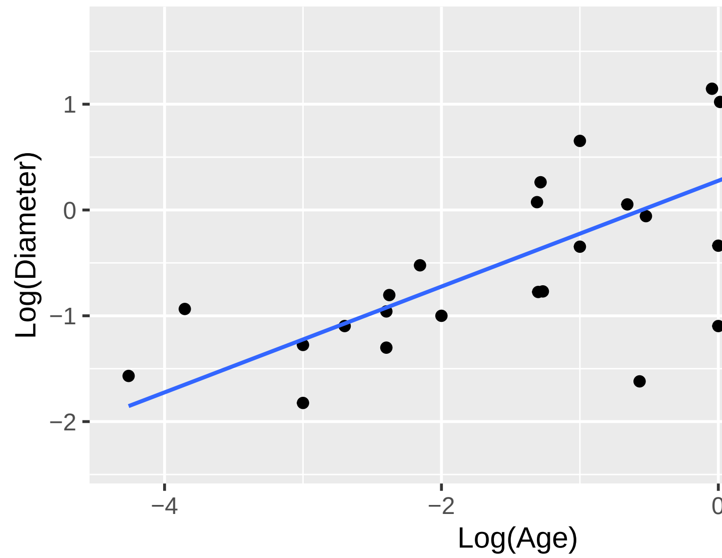

```
# Figure 20.9
gf_histogram(~ resid(bodyfatlm), binwidth = 2, center = 1) %>%
  gf_labs(x = "Residuals", y = "Count")
```




```
# Figure 20.10 is the same idea as Figure 20.3 (page 643)
```

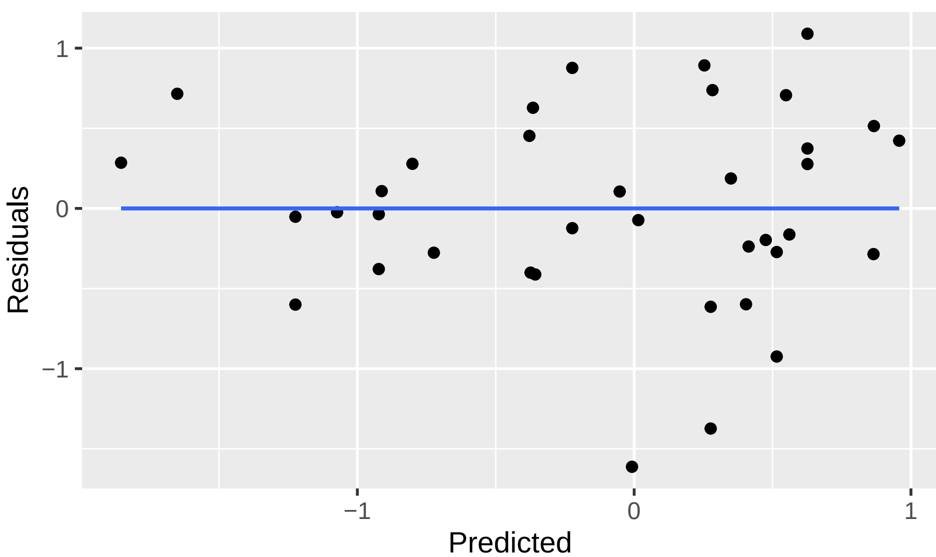
```
Craters <- read_csv("http://nhorton.people.amherst.edu/is5/data/Craters.csv") %>%
  janitor::clean_names() %>%
  filter(log_age <= 1.5) # Removed points to match the textbook
```

```
gf_point(log_diam ~ log_age, data = Craters) %>%
  gf_lm() %>%
  gf_labs(x = "Log(Age)", y = "Log(Diameter)")
```

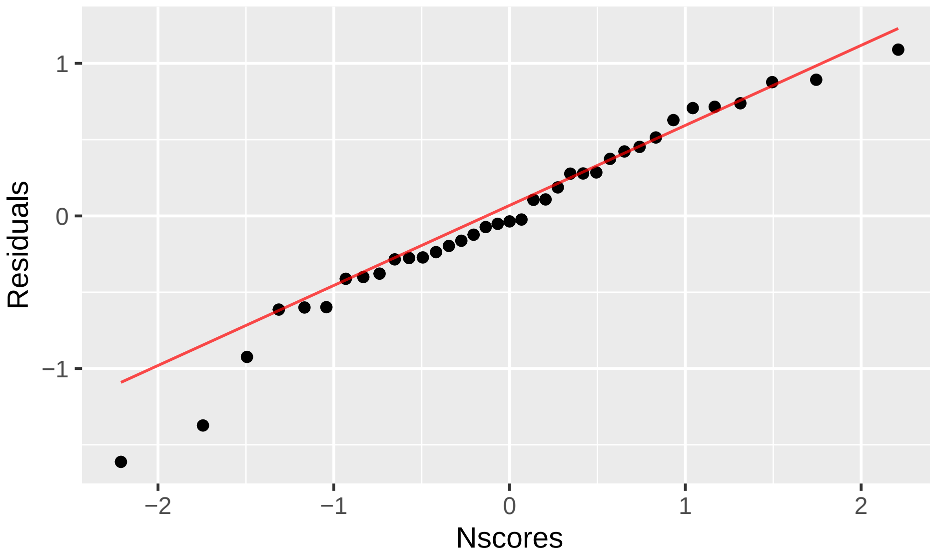


Example 20.1: Checking Assumptions and Conditions

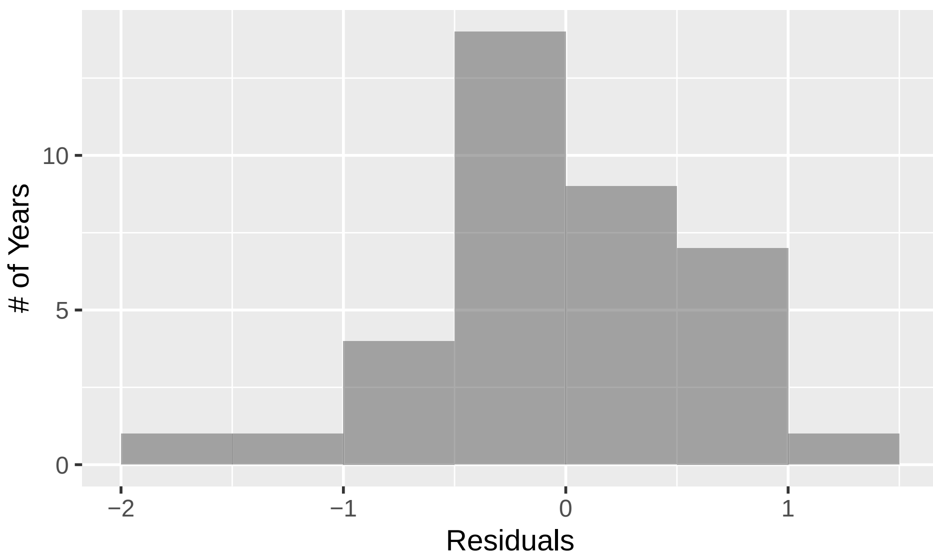
```
craterlm <- lm(log_diam ~ log_age, data = Craters)
gf_point(resid(craterlm) ~ fitted(craterlm)) %>%
  gf_lm() %>%
  gf_labs(x = "Predicted", y = "Residuals")
```




```
gf_qq(~ resid(craterlm)) %>%
  gf_qqline(linetype = "solid", color = "red") %>%
  gf_labs(x = "Nscores", y = "Residuals")
```

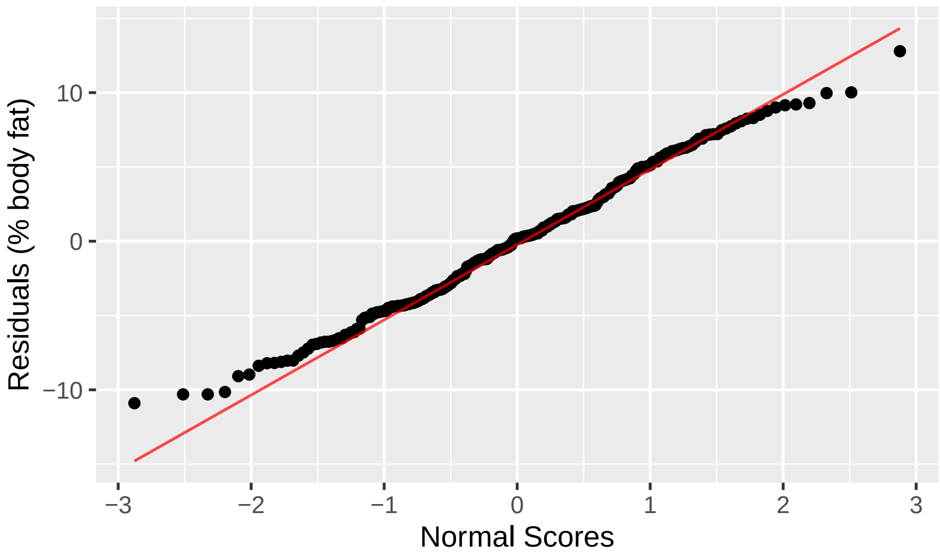



```
gf_histogram(~ resid(craterlm), binwidth = .5, center = 0.25) %>%
  gf_labs(x = "Residuals", y = "# of Years")
```



Step-By-Step Example: Regression Inference The following scatterplot matches Figure 20.1.

```
gf_qq(~ resid(bodyfatlm)) %>%
  gf_qqline(linetype = "solid", color = "red") %>%
  gf_labs(x = "Normal Scores", y = "Residuals (% body fat)")
```




```r
msummary(bodyfatlm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -42.73413    2.71651  -15.73   <2e-16 ***
## waist         1.69997    0.07431   22.88   <2e-16 ***
##
## Residual standard error: 4.713 on 248 degrees of freedom
## Multiple R-squared:  0.6785, Adjusted R-squared:  0.6772
## F-statistic: 523.3 on 1 and 248 DF,  p-value: < 2.2e-16
```

Section 20.3: Regression Inference and Intuition

See the displays on pages 650 and 651.

```r
msummary(bodyfatlm)
```

Example 20.2: Confidence Interval and Hypothesis Test for a Slope

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -42.73413    2.71651  -15.73   <2e-16 ***
## waist         1.69997    0.07431   22.88   <2e-16 ***
##
## Residual standard error: 4.713 on 248 degrees of freedom
## Multiple R-squared:  0.6785, Adjusted R-squared:  0.6772
## F-statistic: 523.3 on 1 and 248 DF,  p-value: < 2.2e-16
```

```r
mean <- 1.70
se <- .074
tstats <- qt(p = c(.025, .975), df = 248)
tstats
```

```
## [1] -1.969576  1.969576
```

```r
mean + tstats * se
```

```
## [1] 1.554251 1.845749
```

```r
t <- (mean - 0.00) / se
t
```

```
## [1] 22.97297
```

Section 20.4: The Regression Table

```r
# Table 20.1, page 654
msummary(bodyfatlm)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -42.73413    2.71651  -15.73   <2e-16 ***
## waist         1.69997    0.07431   22.88   <2e-16 ***
##
## Residual standard error: 4.713 on 248 degrees of freedom
## Multiple R-squared:  0.6785, Adjusted R-squared:  0.6772
## F-statistic: 523.3 on 1 and 248 DF,  p-value: < 2.2e-16
```

Section 20.5: Multiple Regression Inference

```r
# Table 20.2, page 655
bodyfatmlm <- lm(pct_bf ~ waist + height, data = BodyFat)
msummary(bodyfatmlm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.10088    7.68611  -0.403    0.687
## waist        1.77309    0.07158  24.770  < 2e-16 ***
## height      -0.60154    0.10994  -5.472 1.09e-07 ***
##
## Residual standard error: 4.46 on 247 degrees of freedom
## Multiple R-squared:  0.7132, Adjusted R-squared:  0.7109
## F-statistic: 307.1 on 2 and 247 DF,  p-value: < 2.2e-16
```

```r
Mouth <- read_csv("http://nhorton.people.amherst.edu/is5/data/Mouth_volume.csv")
```

Just Checking

```
##
## -- Column specification --------------------------------------------------------------------------------
## cols(
##   Mouth_Volume = col_double(),
##   Age = col_double(),
##   Sex = col_double(),
##   Height = col_double(),
##   Weight = col_double()
## )
```

```r
mouthlm <- lm(Mouth_Volume ~ Height, data = Mouth) # simple linear model
df_stats(~Mouth_Volume, data = Mouth)
```

```
##       response    min     Q1 median     Q3     max     mean      sd  n missing
## 1 Mouth_Volume 35.839 47.647  57.31 69.665 111.181 60.27038 16.8777 61       0
```

```r
msummary(mouthlm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -44.71      32.16  -1.390  0.16966
## Height         61.38      18.77   3.271  0.00179 **
```

```
##
## Residual standard error: 15.66 on 59 degrees of freedom
## Multiple R-squared:  0.1535, Adjusted R-squared:  0.1391
## F-statistic:  10.7 on 1 and 59 DF,  p-value: 0.001794
```

```
mouthmlm <- lm(Mouth_Volume ~ Age + Height, data = Mouth) # multiple linear model
msummary(mouthmlm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -51.0122    31.8843  -1.600  0.11505
## Age           0.4373     0.2588   1.690  0.09646 .
## Height       58.1009    18.5791   3.127  0.00276 **
##
## Residual standard error: 15.42 on 58 degrees of freedom
## Multiple R-squared:  0.1932, Adjusted R-squared:  0.1654
## F-statistic: 6.945 on 2 and 58 DF,  p-value: 0.001978
```

```
Coasters <- read_csv("http://nhorton.people.amherst.edu/is5/data/Coasters_2015.csv")
```

Collinearity

```
##
## -- Column specification --------------------------------------------------------------------------------
## cols(
##   Name = col_character(),
##   Park = col_character(),
##   Track = col_character(),
##   Speed = col_double(),
##   Height = col_double(),
##   Drop = col_double(),
##   Length = col_double(),
##   Duration = col_double(),
##   Inversions = col_double()
## )
```

```
Coasters <- Coasters %>%
  filter(Name != "Tower of Terror", Name != "Xcelerator") %>%
  # Removed artificially accelerated coasters and Tower of Terror
  filter(Drop != "NA", Duration != "NA") %>%
  mutate(Inversions = as.factor(Inversions))
coasterlm <- lm(Duration ~ Drop, data = Coasters) # simple linear model
msummary(coasterlm)
```

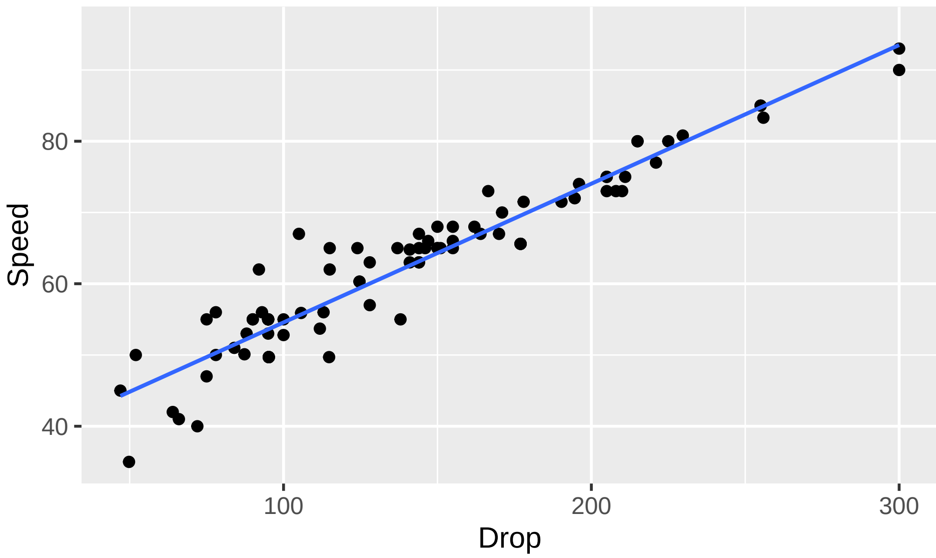
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 88.48688    9.52406   9.291 1.14e-14 ***
## Drop         0.38634    0.06279   6.153 2.26e-08 ***
##
## Residual standard error: 33.27 on 87 degrees of freedom
## Multiple R-squared:  0.3032, Adjusted R-squared:  0.2952
## F-statistic: 37.86 on 1 and 87 DF,  p-value: 2.264e-08
```

```
coastermlm <- lm(Duration ~ Drop + Speed, data = Coasters) # multiple linear regression model
msummary(coastermlm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.3932    34.0567  -0.188  0.85154
```

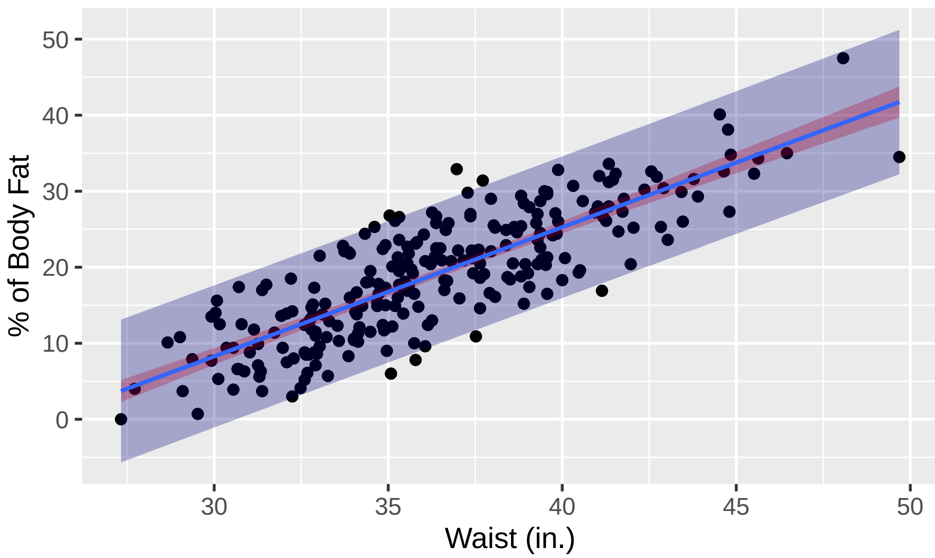
```
## Drop          -0.1399     0.1917  -0.730  0.46754
## Speed          2.7030     0.9346   2.892  0.00484 **
##
## Residual standard error: 31.94 on 86 degrees of freedom
## Multiple R-squared:  0.365,  Adjusted R-squared:  0.3502
## F-statistic: 24.71 on 2 and 86 DF,  p-value: 3.314e-09
```

```
gf_point(Speed ~ Drop, data = Coasters) %>%
  gf_lm()
```



Section 20.6: Confidence and Prediction Intervals

```
# Figure 20.16, page 659
gf_point(pct_bf ~ waist, data = BodyFat) %>%
  gf_lm(interval = "confidence", fill = "red") %>%
  gf_lm(interval = "prediction", fill = "navy") %>%
  gf_labs(x = "Waist (in.)", y = "% of Body Fat")
```

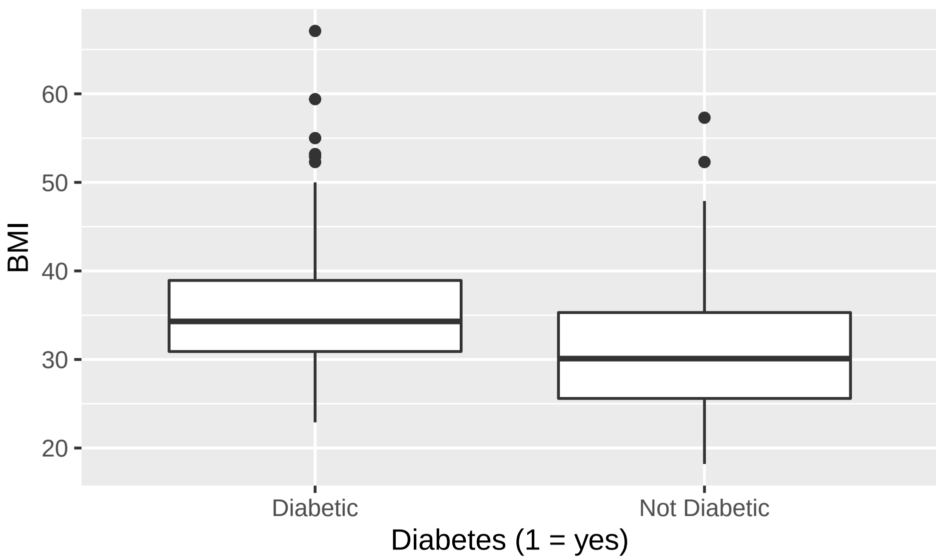


Section 20.7: Logistic Regression

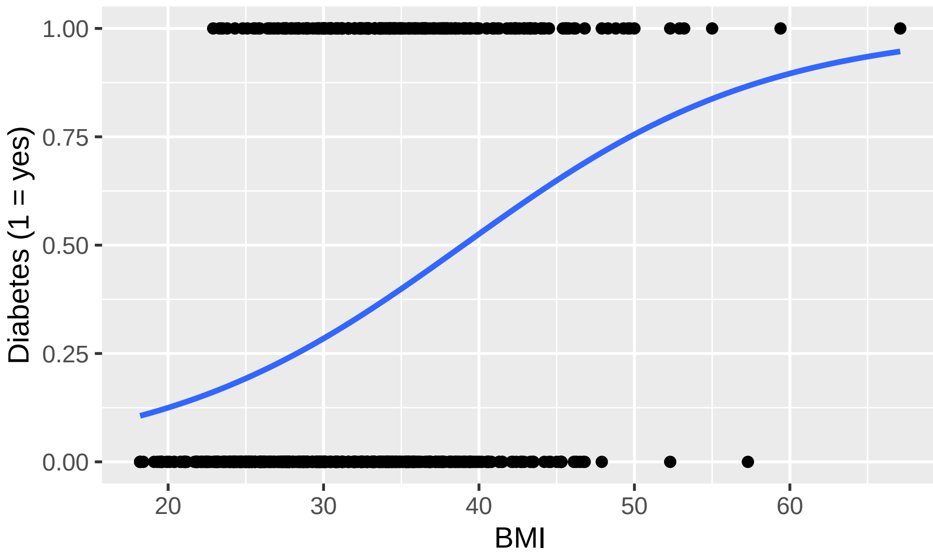
```
PimaIndians <- read_csv("http://nhorton.people.amherst.edu/is5/data/Pima_indians.csv")
```

```
##
## -- Column specification --------------------------------------------------------------------------
## cols(
##   Diabetes = col_double(),
##   BMI = col_double(),
##   Age = col_double()
## )
```

```
PimaIndians <- PimaIndians %>%
  filter(BMI != 0)
# Figure 20.17, page 661
PimaIndians %>%
  mutate(Diabetes = ifelse(Diabetes == 1, "Diabetic", "Not Diabetic")) %>%
  gf_boxplot(BMI ~ as.factor(Diabetes), xlab = "Diabetes (1 = yes)")
```




```
# Figure 20.21, page 663
gf_point(Diabetes ~ BMI, data = PimaIndians, ylab = "Diabetes (1 = yes)") %>%
  gf_smooth(method = "glm", method.args = list(family = "binomial"))
```



Section 20.8: More About Regression