Stat 344 – HW 8

Trey Tipton

March 23, 2022

Problem 6.33

a.)

```
model1 <- lm(y1 ~ x1, data = anscombe)
msummary(model1)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x1            0.5001     0.1179   4.241  0.00217 **
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
model2 <- lm(y2 ~ x2, data = anscombe)
msummary(model2)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x2             0.500      0.118   4.239  0.00218 **
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
model3 <- lm(y3 ~ x3, data = anscombe)
msummary(model3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

```
model4 <- lm(y4 ~ x4, data = anscombe)
msummary(model4)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x4            0.4999     0.1178   4.243  0.00216 **
##
## Residual standard error: 1.236 on 9 degrees of freedom
```

```
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

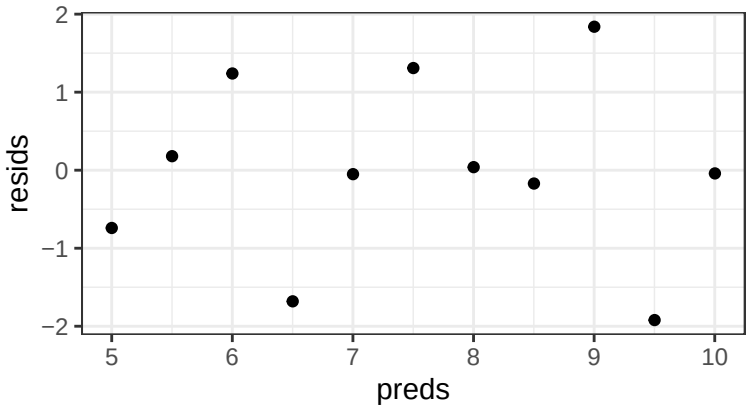
All four models are fairly similar and have slope and intercept estimates that are close to one another.

b.) Model 1:

```
anscombe <- anscombe %>%
  mutate(preds = predict(model1), resids = resid(model1))
```

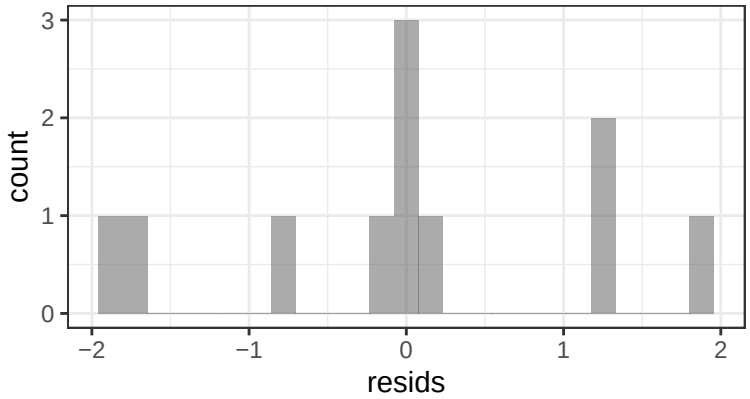
Residuals vs. Fitted Plot

```
gf_point(resids ~ preds, data = anscombe)
```



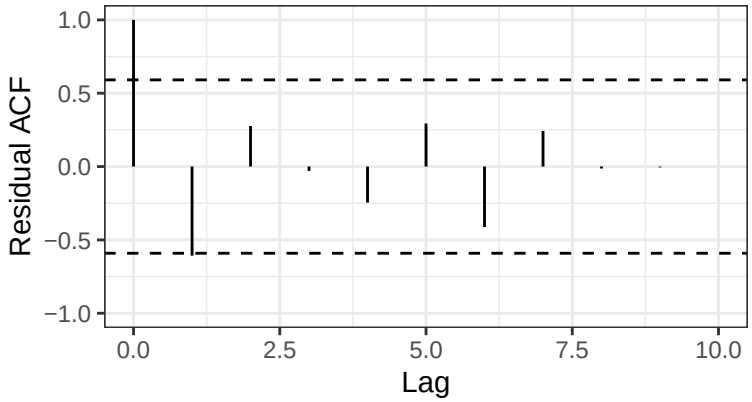
Histogram of Residuals

```
gf_histogram(~resids, data = anscombe)
```



ACF Plot

```
s245::gf_acf(~model1) %>%
  gf_lims(y = c(-1, 1))
```



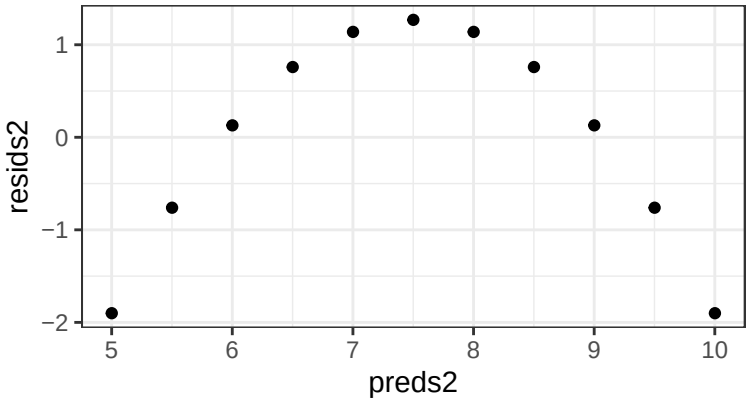
For model one, the residuals seem to be scattered randomly and the histogram of residuals seems to be normal. This model seems to pass these conditions.

Model 2:

```
anscombe <- anscombe %>%
  mutate(preds2 = predict(model2), resids2 = resid(model2))
```

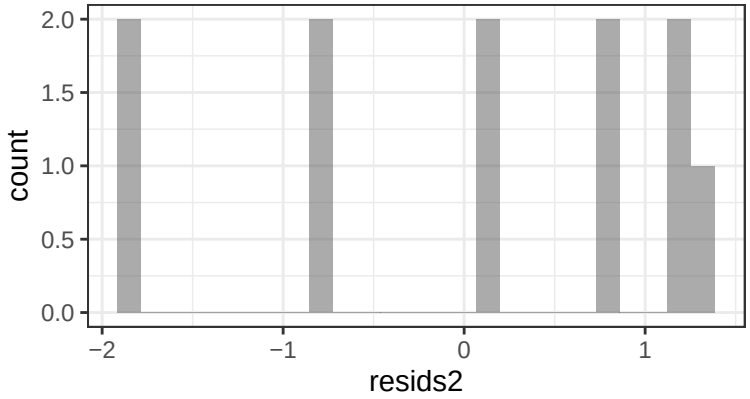
Residuals vs. Fitted Plot

```
gf_point(resids2 ~ preds2, data = anscombe)
```



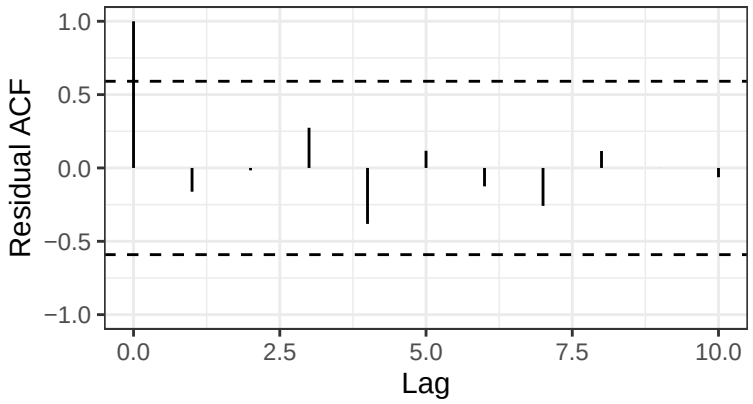
Histogram of Residuals

```
gf_histogram(~resids2, data = anscombe)
```



ACF Plot

```
s245::gf_acf(~model2) %>%
  gf_lims(y = c(-1, 1))
```



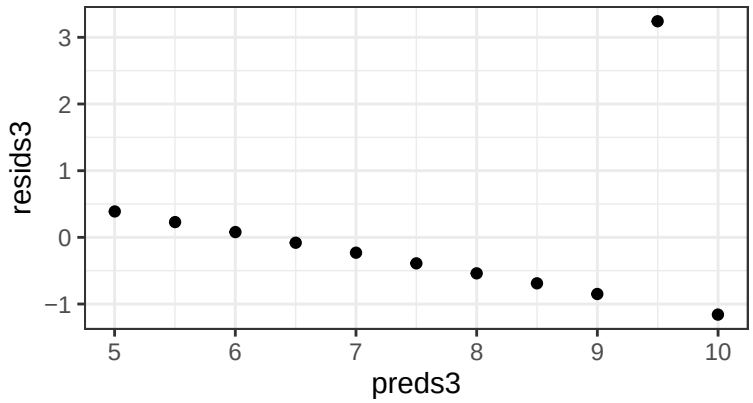
For model two, the residuals are not scattered randomly and the histogram of residuals is not normal. The ACF is okay, but the others make it clear that this model does not pass the conditions.

Model 3:

```
anscombe <- anscombe %>%
  mutate(preds3 = predict(model3), resids3 = resid(model3))
```

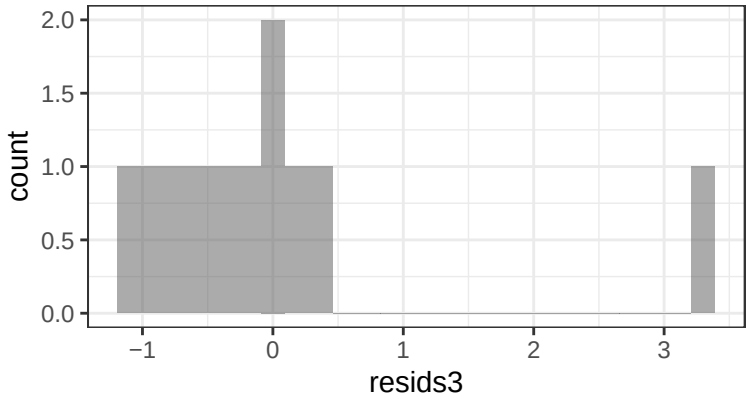
Residuals vs. Fitted Plot

```
gf_point(resids3 ~ preds3, data = anscombe)
```



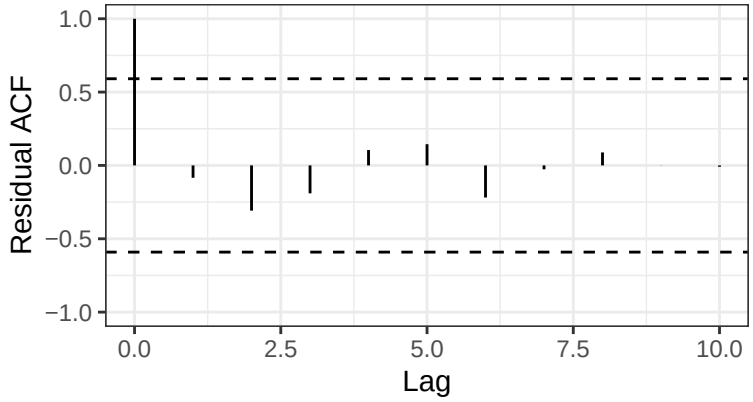
Histogram of Residuals

```
gf_histogram(~resids3, data = anscombe)
```



ACF Plot

```
s245::gf_acf(~model3) %>%
  gf_lims(y = c(-1, 1))
```



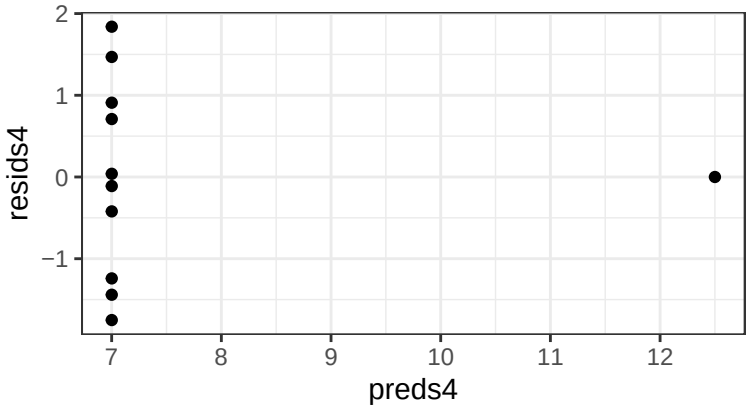
Once again, the ACF plot is okay, but the residuals are not scattered randomly and the histogram of residuals is not normal.

Model 4:

```
anscombe <- anscombe %>%
  mutate(preds4 = predict(model4), resids4 = resid(model4))
```

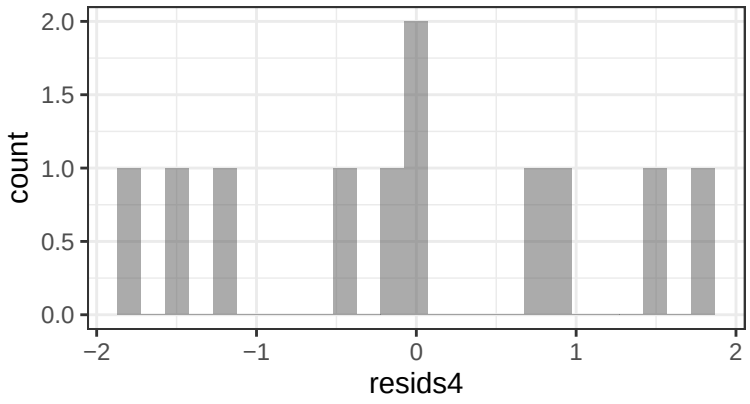
Residuals vs. Fitted Plot

```
gf_point(resids4 ~ preds4, data = anscombe)
```



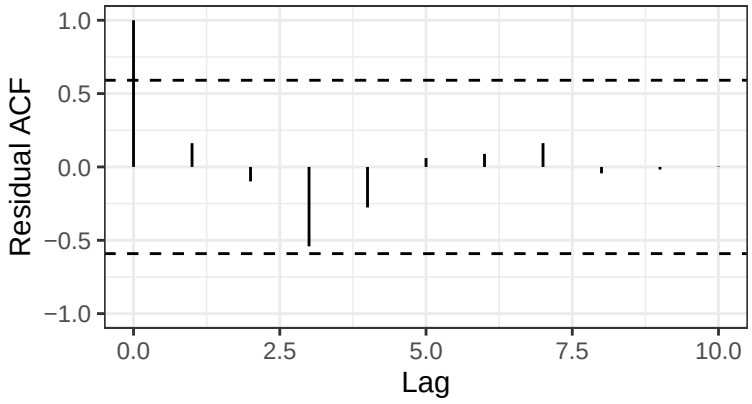
Histogram of Residuals

```
gf_histogram(~resids4, data = anscombe)
```



ACF Plot

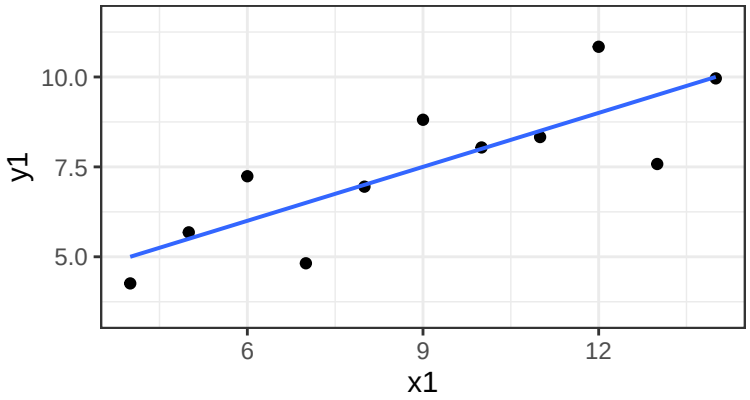
```
s245::gf_acf(~model4) %>%
  gf_lims(y = c(-1, 1))
```



Similar to two and three, this model does not pass the conditions because the residuals have a trend and the histogram of residuals are not normal.

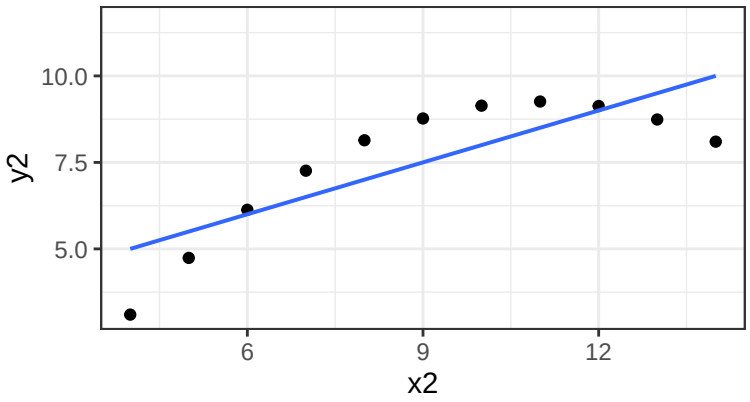
c.)

```
gf_point(y1 ~ x1, data = anscombe)%>%
  gf_lm()
```



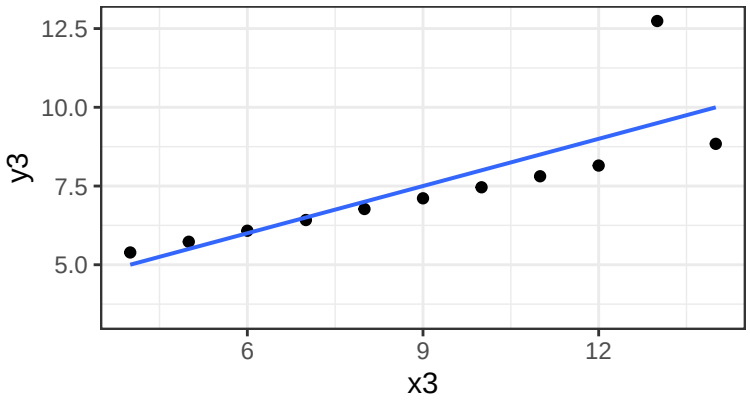
The scatter plot seems to show a linear relationship for this one, and shows that fitting a linear model for this made sense.

```
gf_point(y2 ~ x2, data = anscombe)%>%
  gf_lm()
```



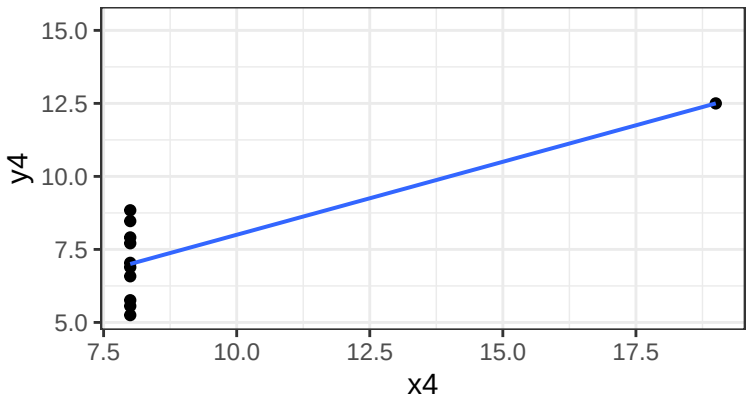
For this one, a linear model clearly does not make sense, and I would not have know that without seeing this scatterplot.

```
gf_point(y3 ~ x3, data = anscombe)%>%
  gf_lm()
```



For this one, there is a clear outlier that is changing the regression line. Looking at this scatterplot beforehand would have been helpful to note that there was an outlier, but the rest of the points have a clear linear trend.

```
gf_point(y4 ~ x4, data = anscombe)%>%
  gf_lm()
```



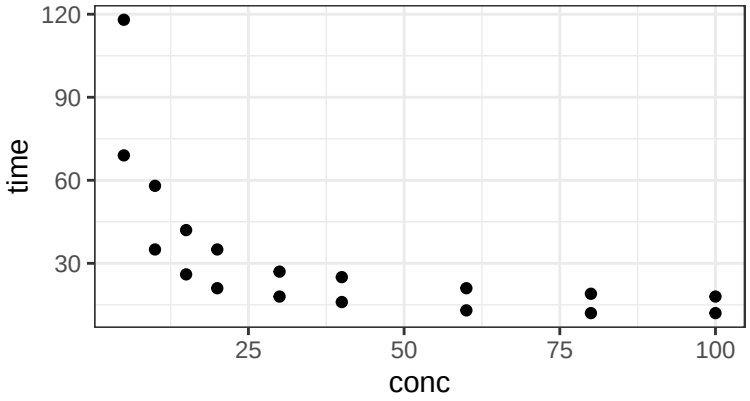
Once again, this data would not make sense to fit a linear model to, and since the model we got seemed fine, we should look at these plots before fitting an lm().

d.) Overall, since all of the models had very similar slope and intercept estimates, it would have seemed that the data for each came from the same population or that all four datasets had similarities. After looking at the scatter plots and checking conditions, it is clear that some of these would not make sense to fit a linear model to or an outlier would affect the model. The scatterplots showed that these data sets had little in common.

Problem 6.59

e.)

```
gf_point(time ~ conc, data = clot)
```



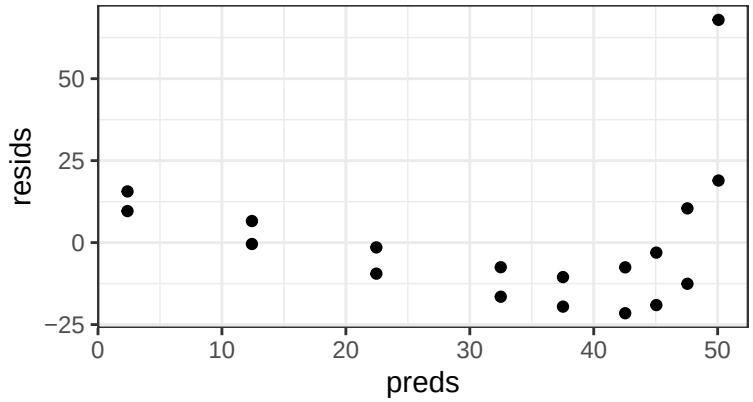

```
clot.model <- lm(time ~ conc, data = clot)
summary(clot.model)
```

```
##
## Call:
## lm(formula = time ~ conc, data = clot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.540 -12.049  -5.275   8.859  67.931
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.5791     8.2295   6.389 8.97e-06 ***
## conc         -0.5020     0.1619  -3.100  0.00688 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.54 on 16 degrees of freedom
## Multiple R-squared:  0.3753, Adjusted R-squared:  0.3362
## F-statistic: 9.612 on 1 and 16 DF,  p-value: 0.006876
```

```
clot <- clot %>%
  mutate(preds = predict(clot.model), resids = resid(clot.model))
```

Residuals vs. Fitted Plot

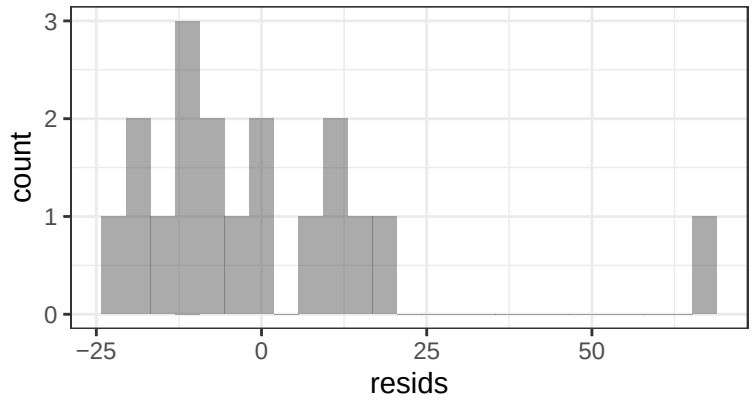
```
gf_point(resids ~ preds, data = clot)
```



This residual plot checks the lack of non-linearity condition. There seems to be a trend in the scatterplot of residuals vs. predictions, so the model does not pass this condition.

Histogram of Residuals

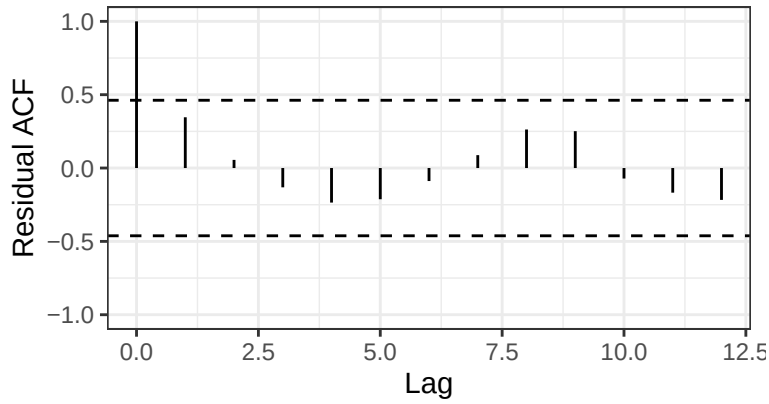
```
gf_histogram(~resids, data = clot)
```



This histogram of residuals checks the normality of residuals condition. The histogram is not normal, so it would not pass this condition, however, if it were not for the other conditions not being passed, I might be able to let this one slide.

ACF Plot

```
s245::gf_acf(~clot.model) %>%
  gf_lims(y = c(-1, 1))
```



This ACF plot checks independence of residuals and it seems to pass this condition, since none of the bars go past the limits.

Our model can not provide reliable conclusions because the conditions are not met.

f.)

```
new_data <- data.frame(conc = 30)
conf_int <- predict(clot.model, newdata = new_data,
        interval = 'confidence',
        level = 0.95)
conf_int
```

```
##        fit      lwr      upr
## 1 37.51977 26.22227 48.81728
```

In this scenario I would want to find an interval estimate for the average time in seconds it takes for blood to clot if the concentration is 30% prothrombin-free plasma. I would use a confidence interval because I would want to find how long it typically takes to clot, not the interval for one individual case. In this case, at a concentration of 30%, the 95% confidence interval for the average time it takes to clot is between 26.22 and 48.82 seconds.

Problem 6.37

c.)

```
act.model <- lm(GPA ~ ACT, data = ACTgpa)
summary(act.model)
```

```
##
## Call:
## lm(formula = GPA ~ ACT, data = ACTgpa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78010 -0.22341  0.05116  0.17520  0.51488
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.11263    0.34120   3.261  0.00331 **
## ACT          0.08702    0.01290   6.747  5.6e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2918 on 24 degrees of freedom
## Multiple R-squared:  0.6548, Adjusted R-squared:  0.6404
## F-statistic: 45.52 on 1 and 24 DF,  p-value: 5.601e-07
```

Using an adjusted R-squared of 0.6404, we can say that 64.04% of the variation in GPAs is explained by a student's ACT score.

d.)

```
new_data <- data.frame(ACT = 25)
conf_int <- predict(act.model, newdata = new_data,
        interval = 'confidence',
        level = 0.95)
conf_int
```

```
##        fit      lwr      upr
## 1 3.288243 3.166694 3.409792
```

A 95% confidence interval for the average GPA for a student who scored 25 on the ACT is (3.17, 3.41).

e.)

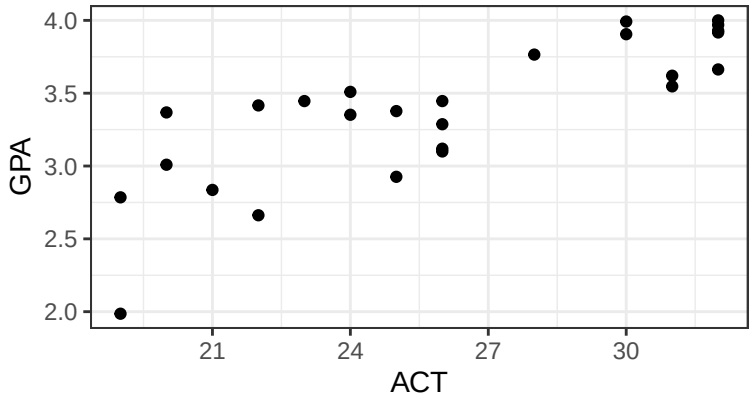
```
new_data <- data.frame(ACT = 30)
conf_int <- predict(act.model, newdata = new_data,
        interval = 'prediction',
        level = 0.95)
conf_int
```

```
##        fit      lwr      upr
## 1 3.723365 3.100775 4.345955
```

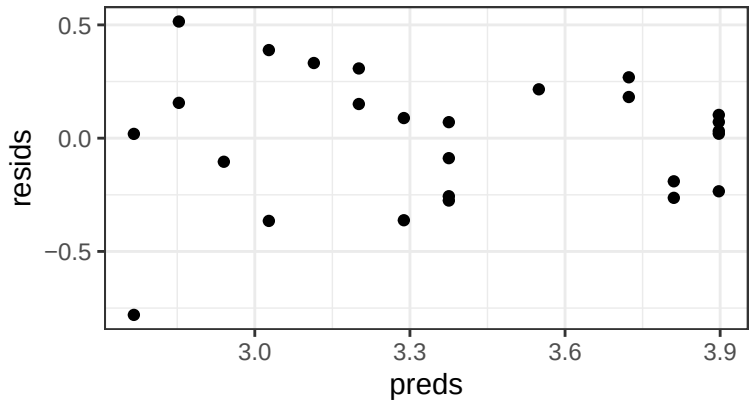
A 95% prediction interval interval for the GPA for a student who scored 30 on the ACT is (3.10, 4.35).

f.)

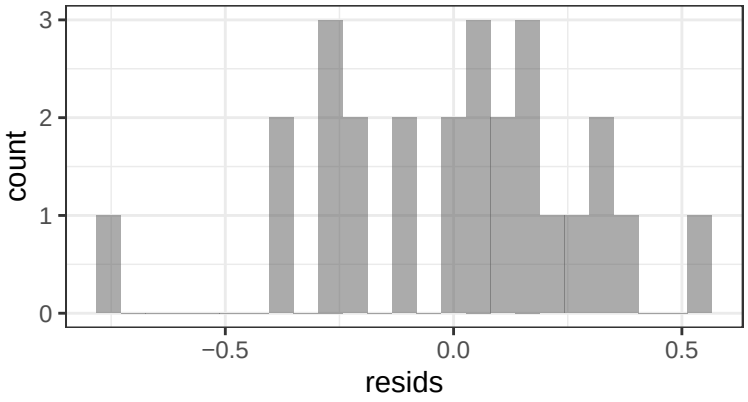
```
ACTgpa <- ACTgpa %>%
  mutate(preds = predict(act.model), resids = resid(act.model))

gf_point(GPA ~ ACT, data = ACTgpa)
```



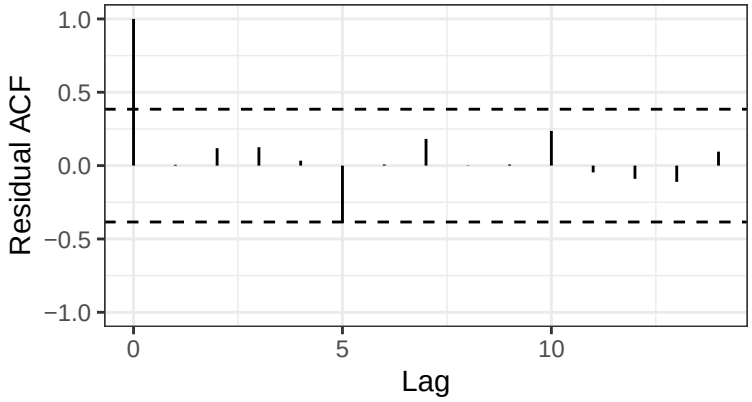

```
gf_point(resids ~ preds, data = ACTgpa)
```




```
gf_histogram(~resids, data = ACTgpa)
```




```
s245::gf_acf(~act.model) %>%
  gf_lims(y = c(-1, 1))
```



Looking at the scatterplot of GPA vs. ACT, there does seem to be a linear relationship, so it makes sense to fit a linear model. After checking conditions, there does not seem to be any reason to be concerned about the analyses we have done, because the histogram of residuals is mostly normal, the ACF plot does not have any that pass the limits, and the scatterplot of residuals vs. predictions is randomly scattered with no trend. It is fair to say that this model passes conditions for normality of residuals, independence of residuals, and lack od non-linearity respectively.

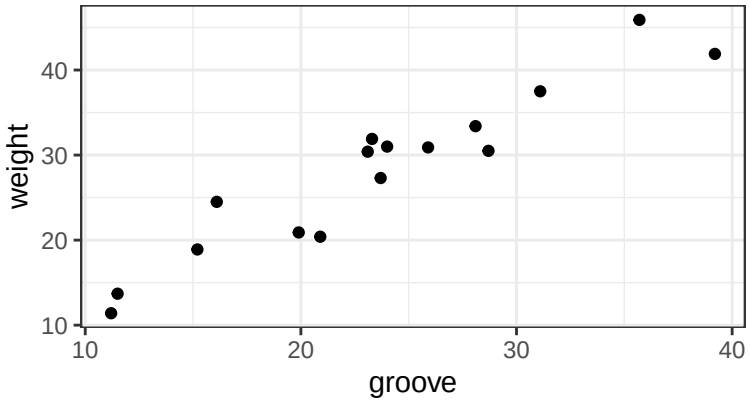
Problem 6.45a

Is there a difference between the two different ways of measuring tread wear: tire weight and groove depth?

Null Hypothesis: $\hat{\beta}_1 = 1$, the slope between tire weight and groove depth is 1.

Alternate Hypothesis: $\hat{\beta}_1 \neq 1$, the slope between the tire weight and groove depth is not 1.

```
gf_point(weight ~ groove, data = TireWear)
```



Since there appears to be a linear relationship, let us create a linear regression model.

```
tire.model <- lm(weight ~ groove, data = TireWear)
summary(tire.model)
```

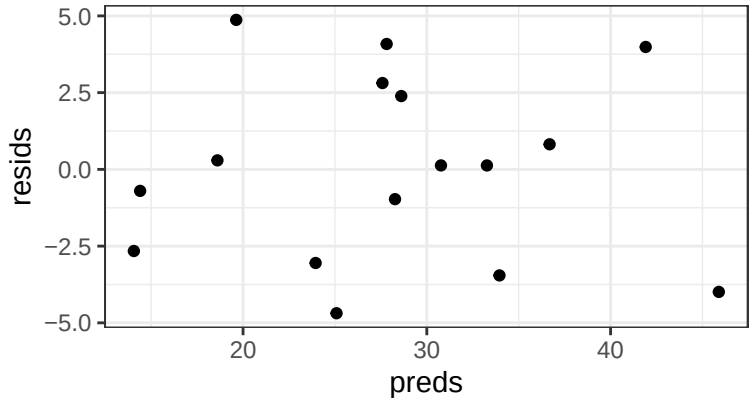
```
##
## Call:
## lm(formula = weight ~ groove, data = TireWear)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6867 -2.7569  0.1284  2.4948  4.8702
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3265     2.5367   0.523    0.609
## groove        1.1369     0.1022  11.124 2.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.143 on 14 degrees of freedom
## Multiple R-squared:  0.8984, Adjusted R-squared:  0.8911
## F-statistic: 123.7 on 1 and 14 DF,  p-value: 2.46e-08
```

Checking Conditions:

```
TireWear <- TireWear %>%
  mutate(preds = predict(tire.model), resids = resid(tire.model))
```

Residuals vs. Fitted Plot

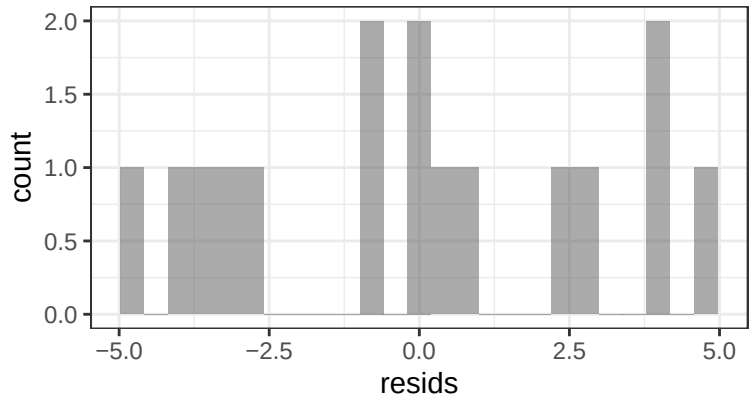
```
gf_point(resids ~ preds, data = TireWear)
```



This residual plot checks the lack of non-linearity condition. There seems to be no trend in the scatterplot of residuals vs. predictions, so the model passes the condition.

Histogram of Residuals

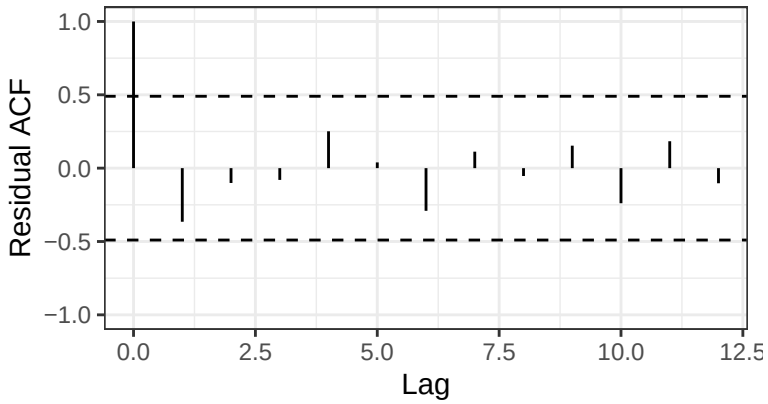
```
gf_histogram(~resids, data = TireWear)
```



This histogram of residuals checks the normality of residuals condition. The histogram is mostly normal with leniency, so the model passes this condition.

ACF Plot

```
s245::gf_acf(~tire.model) %>%
  gf_lims(y = c(-1, 1))
```



This ACF plot checks independence of residuals and it seems to pass this condition, since none of the bars go past the limits.

```
test_stat <- (1.1369 - 1)/0.1022
test_stat
```

```
## [1] 1.33953
```

```
2*(1 - pt(test_stat, 14))
```

```
## [1] 0.2017387
```

Test Stat: Using $\hat{\beta}_1 - \beta_1 / SE(\beta_1)$, we get (1.1369 -1)/0.1022.

p-value and conclusion:

With a p-value of 0.201, we fail to reject the null hypothesis that $\hat{\beta}_1 = 1$. This means that we have enough evidence to say that the weight of the tires and the depth of the grooves provide comparable results.