# Chapter 5: Linear models with a 2-level categorical (factor) explanatory variable

STATS 201/8

University of Auckland

# Learning Outcomes

In this chapter you will learn about:

- Using a 2-level categorical variable as an explanatory variable in a linear model by using indicator variables
- Putting the two-sample $t$-test into the linear model framework
- Relevant `R`-code.

## Section 5.1
## Using categorical variables as explanatory variables by using indicator variables

# New Example – Exam marks vs Attendance

We have gained some understanding about STATS 20x students' final exam marks and how they are related to test and assignment scores, both of which are numeric explanatory variables.

Here, we are going to see if class attendance helps to explain exam score, where class attendance (did or did not) is a categorical explanatory variable.

I am pretty sure we know there is going to be a relationship, but let us answer the question anyway. This also lets us estimate the magnitude of the "attendance effect".

# Exam marks vs Attendance

## The particular variables of interest

| | |
|---|---|
| `Exam` | the student's Exam mark (out of 100) |
| `Attend` | whether the student regularly attended lectures or not – *Yes* or *No*.[1] |

**Note:** As always, our question really is about the 'typical' or average relationship. Some students may attend regularly but not do well in the exam, and vice-versa. That is part of the statistical variability, which we can deal with.

[1] This was measured by taking 4 rolls throughout the semester of lecture attendance. If a student was present for at least 3 of those 4 rolls, then they were recorded as a regular 'attender'.

# Exam marks vs Attendance. . .

## Preliminary exploration of the data

```
> ## Invoke the s20x library
> library(s20x)
> ## Importing data into R
> Stats20x.df = read.table("Data/STATS20x.txt", header=T)
> ## Change Attend from a character variable to a factor variable
> Stats20x.df$Attend = as.factor(Stats20x.df$Attend)
> ## Examine the data
> Stats20x.df$Attend[1:20]
 [1] Yes Yes Yes Yes No  Yes Yes No  Yes Yes No  Yes No  No  No  Yes Yes No  Yes
[20] Yes
Levels: No Yes
```

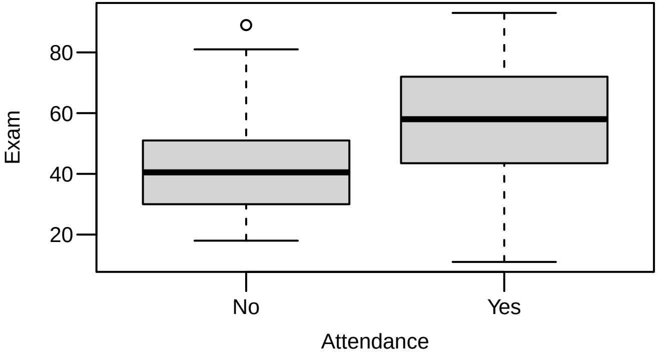
The `Attend` variable has been formatted as a factor variable with two levels - `Levels: No Yes`.

By default, `No` is the first level and `Yes` is the second level, because of alphabetical order – but this can be changed if need be. In this case, we wish to contrast the attenders (`Yes`) against the non-attenders (`No`) and (as you will see later) this ordering of the levels suits us.

# Exam marks vs Attendance...

## Preliminary exploration of the data...

```
> summaryStats(Stats20x.df$Exam,Stats20x.df$Attend)
    Sample Size     Mean Median  Std Dev Midspread
No           46 42.21739   40.5 16.34206     20.50
Yes         100 57.78000   58.0 17.67757     28.25
> plot(Exam~ Attend, data = Stats20x.df, xlab="Attendance")
```




**NOTE:** `Attend` is a factor, so `R` used a boxplot to display the data.

# Exam marks vs Attendance...

## Preliminary exploration of the data...

Here, we are asking how a student's attendance mark ($x$) is related to a their exam ($y$) mark.

As we are interested in the 'typical' student, we want to see what the underlying trend is and how students vary (scatter) about that trend.

Looking at the boxplot, it seems that regular attendance is associated with higher exam scores. Also, the equality of variance assumption seems okay.

# Exam marks vs Attendance. . .

## Preliminary exploration of the data. . .

If you wish to use trendscatter, we need to create a new $x$ variable that is numerical. We can do this by recoding non-attenders as 0s and attenders as 1s.

Here is the code for doing this recoding, with some checks to see that it has been successful.

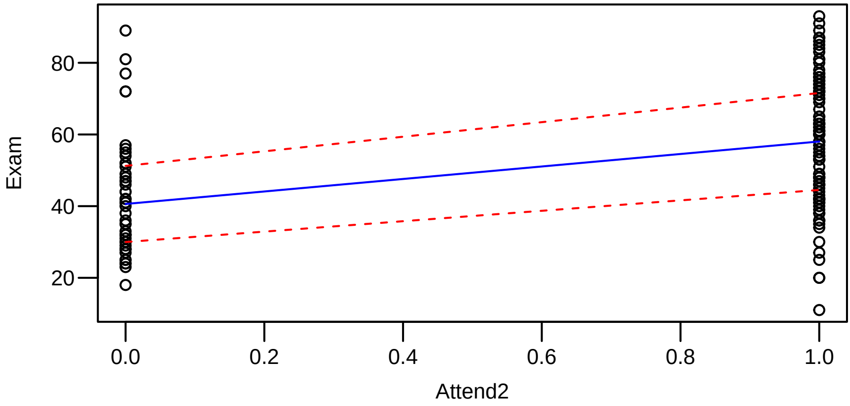
```
> #Make a new variable Attend2 which is 1 if Attend = "Yes" and 0 otherwise
>
> #Note how we use two equal signs, ==, to test equality
> Stats20x.df$Attend2 = as.numeric(Stats20x.df$Attend=="Yes")
> with(Stats20x.df, table(Attend, Attend2))
      Attend2
Attend   0   1
   No   46   0
   Yes   0 100
```

The with function lets us use the variables in the dataframe without having to type the dataframe name every time.

# Exam marks vs Attendance. . .

## Preliminary exploration of the data. . .

```
> trendscatter(Exam~ Attend2, data = Stats20x.df)
```




EOV assumption seems valid.

# Exam marks vs Attendance...

## Fitting a linear model using `Attend2`

Note that `Attend2` is a numeric explanatory variable (albeit with only two different values). So, we can fit a simple linear regression model to see how well `Attend2` explains `Exam`. What would this model tell us?

The linear model for the expected value of `Exam` is

$$\texttt{E[Exam|Attend2]} = \beta_0 + \beta_1 \texttt{Attend2} \ .$$

and so the full model equation is

$$\texttt{Exam}_i = \beta_0 + \beta_1 \texttt{Attend2}_i + \varepsilon_i \text{ where } \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2) \ .$$

If we use the notation `E[Exam|Attend2=0]` to denote the expected value of `Exam` when `Attend2`=0 then we have

$$\texttt{E[Exam|Attend2=0]} = \beta_0 \ .$$

Similarly, when `Attend2`=1

$$\texttt{E[Exam|Attend2=1]} = \beta_0 + \beta_1 \ .$$

# Exam marks vs Attendance...

## Fitting a linear model using `Attend2`...

We see that $\beta_0$ is the mean exam mark for non-attenders and $\beta_0 + \beta_1$ is the mean mark for attenders, and so $\beta_1$ is the difference in mean mark for attenders compared to non-attenders.

Our question of interest was to make inference about the "attendance effect" – this "attendance effect" is simply $\beta_1$, so we can use the methods of inference about the slope of a linear model that we have already seen.

In particular, we know how to test the null hypothesis of no attendance effect, $H_0 : \beta_1 = 0$, and how to obtain confidence intervals. Let's give it a go...

# Exam marks vs Attendance...

## Fitting a linear model using Attend2...

```
> examattend2.fit = lm(Exam~ Attend2, data = Stats20x.df)
> summary(examattend2.fit)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   42.217      2.547  16.578  < 2e-16 ***
Attend2       15.563      3.077   5.058 1.27e-06 ***
---
Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.1508,Adjusted R-squared:  0.145
F-statistic: 25.58 on 1 and 144 DF,  p-value: 1.271e-06
```

Having to create the indicator variable[2] `Attend2` is a nuisance. The good news is that the linear model function `lm` implicitly does this for us.

Here we made the choice to recode non-attenders as 0s and attenders as 1s, rather than the other way around. This was deliberate – it is the same choice that `lm` would make....see below.

[2] So-called because it indicates whether attendance is No or Yes. Some people call these dummy variables.

# Exam marks vs Attendance...

## Fitting a linear model using Attend

We now fit a linear model to these data using the factor variable Attend. The lm function automatically attributes Attend=="No" the zero value [3] because **N**o comes before **Y**es in the alphabet, and Attend=="Yes" is attributed the value 1.

```
> examattend.fit = lm(Exam~ Attend, data = Stats20x.df)
> summary(examattend.fit)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   42.217      2.547  16.578  < 2e-16 ***
AttendYes     15.563      3.077   5.058 1.27e-06 ***
---
Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.1508,Adjusted R-squared:  0.145
F-statistic: 25.58 on 1 and 144 DF,  p-value: 1.271e-06
```

What is the difference? Nothing really – slightly different formats of the coefficient names as Attend2 is numerical and Attend is a categorical variable (factor).

[3] This is called the **baseline** or reference level of the factor variable.

## Section 5.2
## Model checking and inference

# Exam marks vs Attendance...

## The fitted model

Note that the p-value for `Attend` is very small, so we conclude that there is indeed a significant effect of attendance.

The estimates of $\beta_0$ and $\beta_1$ are

```
(Intercept)    AttendYes
   42.21739     15.56261
```

That is, (formatted to 2 decimal places) $\hat{\beta}_0 = 42.22$ and $\hat{\beta}_1 = 15.56$.

Using these estimated coefficients[4] our estimated values for the exam score of a 'typical' student are:

$$\widehat{\texttt{Exam}} = 42.22 + 15.56 \times \texttt{Attend2}$$

or

$$\widehat{\texttt{Exam}} = \begin{cases} 42.22 & \text{, for non-attenders and} \\ 42.22 + 15.56 & \text{, for attenders.} \end{cases}$$

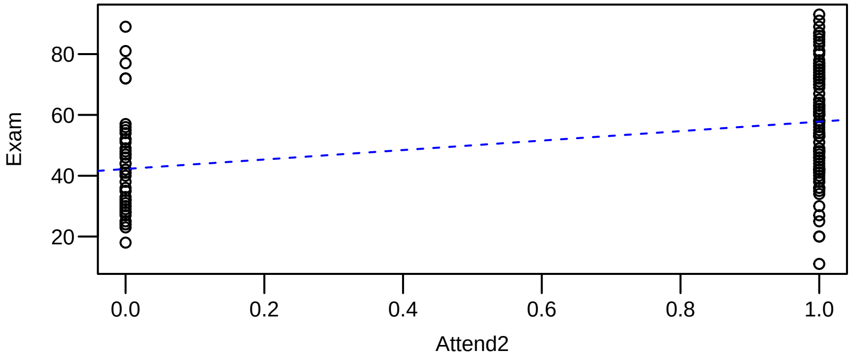
[4]Subject to verification of the model assumptions - stay tuned

# Exam marks vs Attendance...

## The fitted model...

Let's visualize the fit. Here we are plotting the 'best' estimated straight line that we obtained from fitting our model using the indicator variable `Attend2`.

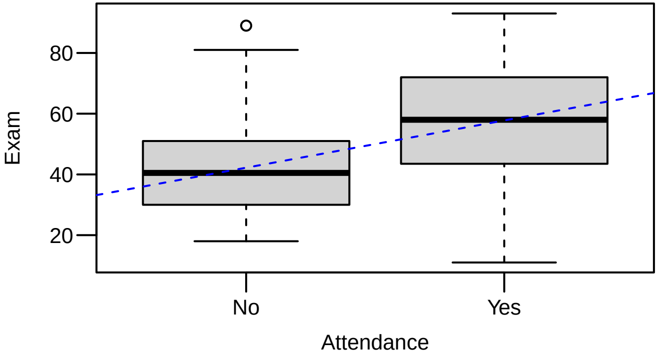
```
> plot(Exam ~ Attend2, data = Stats20x.df)
> ## Add the lm estimated line to this  plot  where a=intercept, b=slope
> abline(coef(examattend.fit),lty=2, col="blue")
```



# Exam marks vs Attendance...

## The fitted model...

Here is the same thing using the factor variable `Attend`, with the fit overlaid on the boxplots.




The two plots above essentially present the same information, repackaged in a slightly different way.

# Exam marks vs Attendance...

## Checking assumptions

We should check that our assumptions hold before we report (or use) the results from this analysis. Remember, we require independence, identical distribution and normality of the random components

$$\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2).$$

The assumptions (in order of importance):

**i**id – independence. We check this by investigating how we obtained the data.

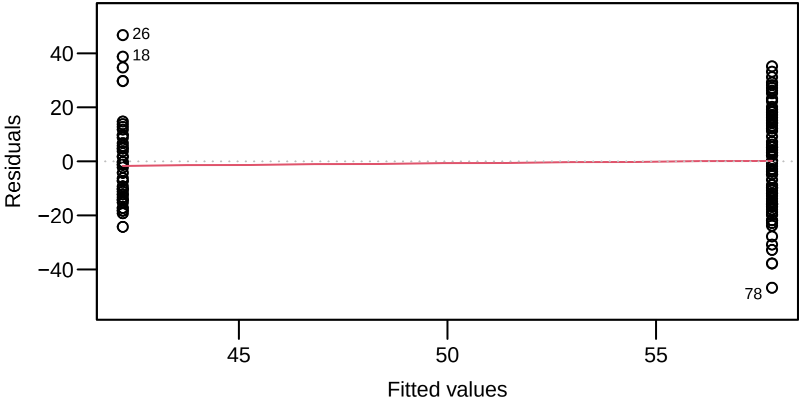
i**i**d – identically distributed. This should result in the variation of the residuals being roughly constant (regardless of the fitted value) and the residuals more-or-less averaging around zero. We can use `plot()` with the `which=1` argument.

ii**d** – Normality. We only check this having validated the first two assumptions, using `normcheck`.

# Exam marks vs Attendance...

## Checking our assumptions

```
> plot(examattend.fit,which=1)
```

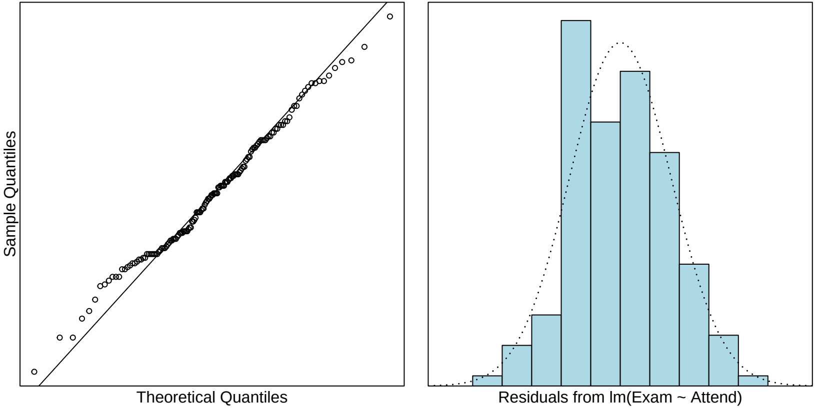



No cause for concern.

# Exam marks vs Attendance...

## Checking our assumptions...

```
> normcheck(examattend.fit)
```

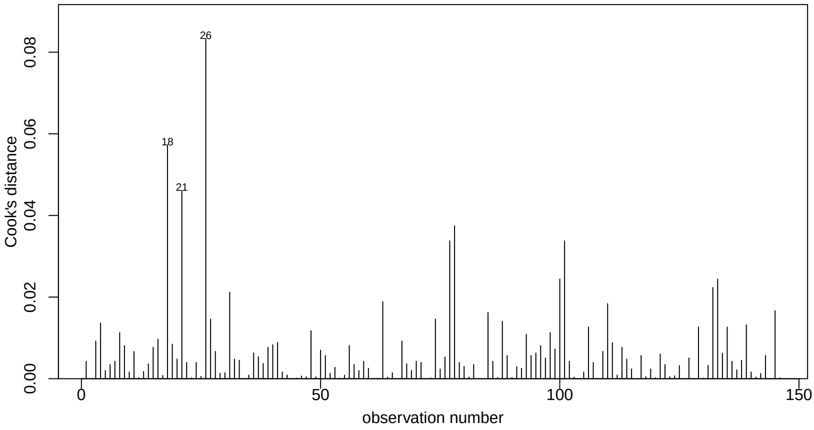



Looks good - the residuals appear to have a near normal distribution.

# Exam marks vs Attendance. . .

## Checking our assumptions. . .

```
> cooks20x(examattend.fit)
```




No unduly influential points.

# Inference about Exam marks vs Attendance

## Testing the null hypothesis

The model assumptions look to be reasonably well satisfied, so we can use the fitted model to make statistical inference (i.e., to answer questions of interest).

We begin by testing the null hypothesis that there is no effect of the explanatory variable `Attend`.[5] Recall that this is $H_0 : \beta_1 = 0$.

```
> coef(summary(examattend.fit))
            Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 42.21739   2.546517 16.578482 3.358318e-35
AttendYes   15.56261   3.076969  5.057773 1.271370e-06
```

The $P$-value for `AttendYes` is testing the null hypothesis $H_0 : \beta_1 = 0$. It is highly statistically significant $p = 1.27 \times 10^{-6}$, which is just over 1 in a million. We have extremely strong evidence that attendance is related to exam score.

[5] In this case, the null hypothesis corresponds to the `Exam` scores being iid.

# Inference about Exam marks vs Attendance...

## Calculating confidence intervals for effect size

We can get confidence intervals for $\beta_1$ (and $\beta_0$ if need be) by:

```
> confint(examattend.fit)
                2.5 %   97.5 %
(Intercept) 37.184009 47.25077
AttendYes    9.480749 21.64447
```

Here we can say that, on average, regular attenders will obtain an increased exam mark of between 9.5 to 21.6 compared to non-attenders.

Alternative wording would be: the expected exam mark of a student who regularly attends class is 9.5 to 21.6 marks higher than that of a non-attendee.

# Inference about Exam marks vs Attendance. . .

## Calculating confidence and prediction intervals

We might also want to estimate and/or predict exam marks based on attendance status.

Recall that we need to make a new dataframe containing the values of the explanatory variable to be used for the prediction. In this case, that is just "No" and "Yes".

```
> ## Create data frame of values of interest: Attend=="Yes" and "No"
> ## Make sure that the names of vars are exactly the same as in the data frame
> preds.df = data.frame(Attend = c("No", "Yes"))
> predict(examattend.fit, preds.df, interval = "confidence")
       fit      lwr      upr
1 42.21739 37.18401 47.25077
2 57.78000 54.36619 61.19381
> predict(examattend.fit, preds.df, interval = "prediction")
       fit       lwr      upr
1 42.21739  7.710259 76.72452
2 57.78000 23.471673 92.08833
```

# Inference about Exam marks vs Attendance...

## Confidence and prediction intervals

Here, we estimate the exam mark for non-regular attenders is between 37.2 and 47.3 on average, whereas for those who regularly attend it is between 54.4 and 61.2.

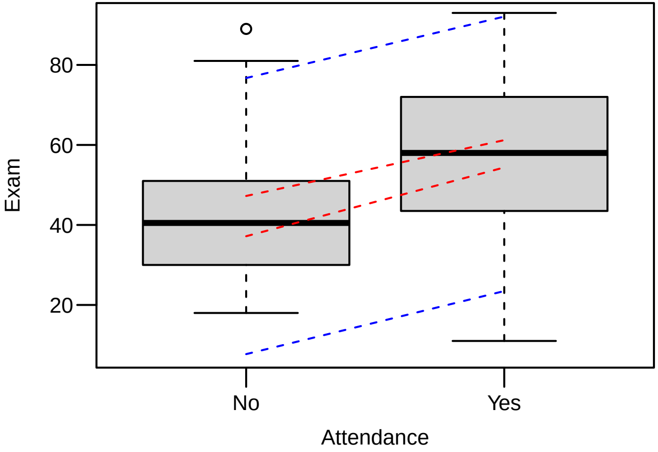
For any individual student, if they are a non-regular attender then we predict their exam mark to be between 7.7 and 76.7, or between 23.5 and 92.1 if they do attend regularly.

The prediction intervals are very wide – which is not surprising as regular attendance only explains 15% of the variation in exam score, and there is plenty of variability between individual students.

# Inference about Exam marks vs Attendance. . .

## Confidence and prediction intervals

Here is what the confidence (red)/prediction intervals (blue) look like:



## Section 5.3
## Putting the two-sample t-test into the linear model framework

# Two-sample *t*-test in disguise

We have just done an analysis to see if two groups (attendees and non-attendees) differ in expected value. You have encountered this scenario previously – the two-sample *t*-test.

By default, the `t.test` function in `R` relaxes the equality of variance assumption, so we have to explicitly tell it not to (using the `var.equal=TRUE` argument) if we want to reproduce our `lm` results exactly.

```
> t.test(Exam~ Attend, var.equal=TRUE, data = Stats20x.df)

Two Sample t-test

data:  Exam by Attend
t = -5.0578, df = 144, p-value = 1.271e-06
alternative hypothesis: true difference in means between group No and group Yes is 
95 percent confidence interval:
 -21.644468  -9.480749
sample estimates:
 mean in group No mean in group Yes
         42.21739          57.78000
```

# Exam marks vs Attendance...

## Two-sample *t*-test

The two-sample *t*-test has a variant which relaxes the EOV assumption. This is known as the **Welch** form of the *t*-test, and is the default in the `t.test` function.

```
> t.test(Exam~ Attend, var.equal=FALSE, data = Stats20x.df)
```

or just

```
> t.test(Exam~ Attend, data = Stats20x.df)

Welch Two Sample t-test

data:  Exam by Attend
t = -5.2076, df = 94.09, p-value = 1.122e-06
alternative hypothesis: true difference in means between group No and group Yes is
95 percent confidence interval:
 -21.496121  -9.629096
sample estimates:
 mean in group No mean in group Yes
         42.21739          57.78000
```

# Exam marks vs Attendance...

## Two-sample $t$-test...

The Welch form loses some degrees of freedom because it adds more uncertainty as we have to now estimate the variability of the sample in each group (attenders and non-attenders) rather than 'pooling' them based on the EOV assumption.

In this case, there is negligible difference between the standard and **Welch** forms of the two-sample $t$-test.

# Summary

We have now seen that `lm` can be used when the explanatory variable is numeric (e.g., `Test` or `Assign`), and also when the explanatory variable is categorical (e.g., `Attend`).

When the explanatory variable is categorical, `lm` automatically creates numeric indicator variables to use in the model formula. The indicator variables indicate the category level (relative to the baseline level) and take the value 0 or 1 – for this reason they are sometimes referred to as indicator variables.

The natural question to ask here is: can we use `Test` and `Attend` *together* to explain exam score?

Stay tuned...

# Section 5.4
# R tips and relevant code

# R tips and tricks

## Use of -1 in model formulae

In some situations it is useful to fit the model without a baseline level for the intercept.

This is easy, just add -1 to the model formula.

```
> NoBaseline.fit=lm(Exam~ Attend-1, data = Stats20x.df)
> summary(NoBaseline.fit)
```

```
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
AttendNo    42.217      2.547   16.58   <2e-16 ***
AttendYes   57.780      1.727   33.45   <2e-16 ***
---
Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.9064,Adjusted R-squared:  0.9051
F-statistic:   697 on 2 and 144 DF,  p-value: < 2.2e-16
> confint(NoBaseline.fit)
             2.5 %   97.5 %
AttendNo  37.18401 47.25077
AttendYes 54.36619 61.19381
```

Note: $R^2$ has no meaning when there is no intercept term in the model.

# Most of the R-code you need for this chapter

You do not need to create indicator variables as R does this for you. It will choose the baseline for you, so be careful. You can change this if needed – you will see an example of this soon.

```
> examattend.fit = lm(Exam~ Attend, data = Stats20x.df)
```

This is equivalent to

```
> t.test(Exam~ Attend, var.equal=TRUE, data = Stats20x.df)
```

If it is clear that the two groups have massively different variances then one approach would be to abandon the use of a linear model and use the modified t-test without the equality of variance assumption[6]

```
> t.test(Exam~ Attend, var.equal=FALSE, data = Stats20x.df)
```

**However**, in most cases the technique shown in the next Chapter is a better way to cope with inequality of variance.

[6]The modified t-test approach will **never** be used in this class.