MAP501 Coursework Submission

Preamble

```
library("rio")
library("dplyr")
library("tidyr")
library("magrittr")
library("ggplot2")
library("pROC")
library("car")
library("nnet")
library("caret")
library("lme4")
library("AmesHousing")
library("here")
library("tidyverse")
library("janitor")
library("ggrepel")
library("sandwich")
library("investr")
library("rcompanion")
library("ggcorrplot")
library("corrr")
library("effects")
library("lindia")

Ames <- make_ames()
```

1. Data Preparation

1a)

Import the soccer.csv dataset as “footballer_data”. (2 points)

```
footballer_data <-
  read_csv(here("data/soccer.csv"))
```

1b)

Ensure all character variables are treated as factors and where variable names have a space, rename the variables without these. (3 points)

```
footballer_data <-
  footballer_data %>%
  clean_names() %>%
  mutate_at(vars(full_name, birthday_gmt, position, current_club, nationality),
            list(factor))

footballer_data
```

```
# A tibble: 570 × 45
   full_…¹   age birth…² birth…³ posit…⁴ curre…⁵ minut…⁶ minut…⁷ minut…⁸ natio…⁹
   <fct>   <dbl>   <dbl> <fct>   <fct>   <fct>     <dbl>   <dbl>   <dbl> <fct>
 1 Aaron …    32  6.30e8 15/12/… Defend… West H…    1589     888     701 England
 2 Aaron …    35  5.46e8 16/04/… Midfie… Burnley    1217     487     730 England
 3 Aaron …    31  6.53e8 15/09/… Midfie… Hudder…    2327    1190    1137 Austra…
 4 Aaron …    31  6.62e8 26/12/… Midfie… Arsenal    1327     689     638 Wales
 5 Aaron …    22  9.68e8 07/09/… Forward Hudder…      69      14      55 England
 6 Aaron …    24  8.81e8 26/11/… Midfie… Crysta…    3135    1605    1530 England
 7 Abdelh…    25  8.49e8 28/11/… Midfie… Hudder…      49       0      49 Morocco
 8 Abdoul…    29  7.26e8 01/01/… Midfie… Watford    3062    1566    1496 France
 9 Abouba…    27  7.95e8 07/03/… Forward Fulham      687     468     219 France
10 Adalbe…    25  8.65e8 31/05/… Forward Watford       0       0       0 Venezu…
# … with 560 more rows, 35 more variables: appearances_overall <dbl>,
#   appearances_home <dbl>, appearances_away <dbl>, goals_overall <dbl>,
#   goals_home <dbl>, goals_away <dbl>, assists_overall <dbl>,
#   assists_home <dbl>, assists_away <dbl>, penalty_goals <dbl>,
#   penalty_misses <dbl>, clean_sheets_overall <dbl>, clean_sheets_home <dbl>,
#   clean_sheets_away <dbl>, conceded_overall <dbl>, conceded_home <dbl>,
#   conceded_away <dbl>, yellow_cards_overall <dbl>, red_cards_overall <dbl>, …
```

Running the footballer_data code, allows you to confirm that these changes have actually happened, as the variables selected are now seen as factors, and the names have been cleaned using clean_names().

1c)

Remove the columns birthday and birthday_GMT. (2 points)

```
footballer_data2 <-
  footballer_data %>%
  select(-c(birthday, birthday_gmt))

footballer_data2
```

```
# A tibble: 570 × 43
   full_…¹   age posit…² curre…³ minut…⁴ minut…⁵ minut…⁶ natio…⁷ appea…⁸ appea…⁹
   <fct>   <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl> <fct>     <dbl>   <dbl>
 1 Aaron …    32 Defend… West H…    1589     888     701 England      20      11
 2 Aaron …    35 Midfie… Burnley    1217     487     730 England      16       7
 3 Aaron …    31 Midfie… Hudder…    2327    1190    1137 Austra…      29      15
 4 Aaron …    31 Midfie… Arsenal    1327     689     638 Wales        28      14
 5 Aaron …    22 Forward Hudder…      69      14      55 England       2       1
 6 Aaron …    24 Midfie… Crysta…    3135    1605    1530 England      35      18
 7 Abdelh…    25 Midfie… Hudder…      49       0      49 Morocco       2       0
 8 Abdoul…    29 Midfie… Watford    3062    1566    1496 France       35      18
 9 Abouba…    27 Forward Fulham      687     468     219 France       13       8
10 Adalbe…    25 Forward Watford       0       0       0 Venezu…       0       0
# … with 560 more rows, 33 more variables: appearances_away <dbl>,
#   goals_overall <dbl>, goals_home <dbl>, goals_away <dbl>,
#   assists_overall <dbl>, assists_home <dbl>, assists_away <dbl>,
#   penalty_goals <dbl>, penalty_misses <dbl>, clean_sheets_overall <dbl>,
#   clean_sheets_home <dbl>, clean_sheets_away <dbl>, conceded_overall <dbl>,
#   conceded_home <dbl>, conceded_away <dbl>, yellow_cards_overall <dbl>,
#   red_cards_overall <dbl>, goals_involved_per_90_overall <dbl>, …
```

Running footballer_data2 shows that the columns have been removed as they had initially came after age in the orignial table, yet are not present now.

1d)

Remove the cases with age<=15 and age>40. (2 points)

```
footballer_data3 <-
  footballer_data2 %>%
  filter(age > 15 & age <= 40)

max(footballer_data3$age)
min(footballer_data3$age)
footballer_data3
```

```
[1] 40
[1] 20
# A tibble: 565 × 43
   full_…¹   age posit…² curre…³ minut…⁴ minut…⁵ minut…⁶ natio…⁷ appea…⁸ appea…⁹
   <fct>   <dbl> <fct>   <fct>     <dbl>   <dbl>   <dbl> <fct>     <dbl>   <dbl>
 1 Aaron …    32 Defend… West H…    1589     888     701 England      20      11
 2 Aaron …    35 Midfie… Burnley    1217     487     730 England      16       7
 3 Aaron …    31 Midfie… Hudder…    2327    1190    1137 Austra…      29      15
 4 Aaron …    31 Midfie… Arsenal    1327     689     638 Wales        28      14
 5 Aaron …    22 Forward Hudder…      69      14      55 England       2       1
 6 Aaron …    24 Midfie… Crysta…    3135    1605    1530 England      35      18
 7 Abdelh…    25 Midfie… Hudder…      49       0      49 Morocco       2       0
 8 Abdoul…    29 Midfie… Watford    3062    1566    1496 France       35      18
 9 Abouba…    27 Forward Fulham      687     468     219 France       13       8
10 Adalbe…    25 Forward Watford       0       0       0 Venezu…       0       0
# … with 555 more rows, 33 more variables: appearances_away <dbl>,
#   goals_overall <dbl>, goals_home <dbl>, goals_away <dbl>,
#   assists_overall <dbl>, assists_home <dbl>, assists_away <dbl>,
#   penalty_goals <dbl>, penalty_misses <dbl>, clean_sheets_overall <dbl>,
#   clean_sheets_home <dbl>, clean_sheets_away <dbl>, conceded_overall <dbl>,
#   conceded_home <dbl>, conceded_away <dbl>, yellow_cards_overall <dbl>,
#   red_cards_overall <dbl>, goals_involved_per_90_overall <dbl>, …
```

By using the code for max and min age, you can confirm that the code has correctly carried out its task, as it has removed all players older than 40, and all players younger than 15 (although the youngest player is 20 in the data set).

2. Linear Regression

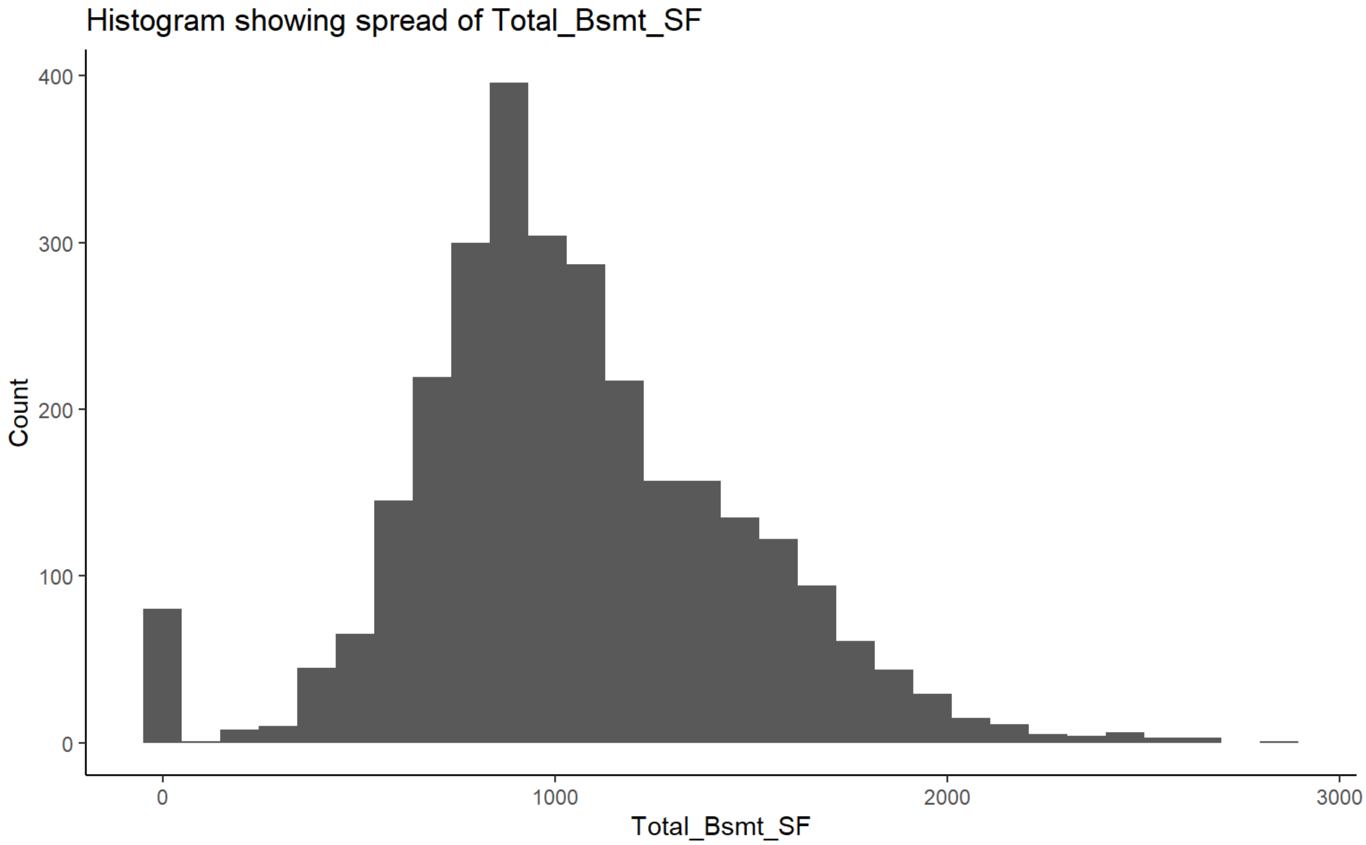
In this problem, you are going to investigate the response variable Total_Bsmt_SF in “Ames” dataset through linear regression.

2a)

By adjusting x axis range and number of bars, create a useful histogram of Total_Bsmt_SF on the full dataset. Ensure that plot titles and axis labels are clear. (4 points)

As most of the data points are between 0 - 3000 Total_Bsmt_SF, the values beyond this are assumed to be outliers and will be removed to allow for a more representative histogram. There is an outlier value of 6110 too, which disrupts the histogram plot. The number of bars is selected to be 30 so that each bar contains fewer data points, making each bar more accurate to the actual size of the Total_Bsmt_SF.

```
Ames %>%
  filter(Total_Bsmt_SF <= 3000) %>%
  ggplot(mapping = aes(x = Total_Bsmt_SF)) +
  geom_histogram(bins = 30) +
  labs(y = "Count",
       title = "Histogram showing spread of Total_Bsmt_SF") +
  theme_classic()
```



2b)

Using “Ames” dataset to create a new dataset called “Ames2” in which you remove all cases corresponding to:

i. MS_Zoning categories of A_agr (agricultural), C_all (commercial) and I_all (industrial),

ii. BsmtFin_Type_1 category of “No_Basement”.

iii. Bldg_Type category of “OneFam”

and drop the unused levels from the dataset “Ames2”. (4 points)

```
Ames2 <-
  Ames %>%
  filter(MS_Zoning != "A_agr", MS_Zoning != "C_all", MS_Zoning != "I_all") %>%
  filter(BsmtFin_Type_1 != "No_Basement") %>%
  filter(Bldg_Type != "OneFam") %>%
  droplevels()
```

To confirm that these changes have been made, the following summaries are carried out to show that the omitted categories are no longer present.

```
summary(Ames2$MS_Zoning)
summary(Ames2$BsmtFin_Type_1)
summary(Ames2$Bldg_Type)
```

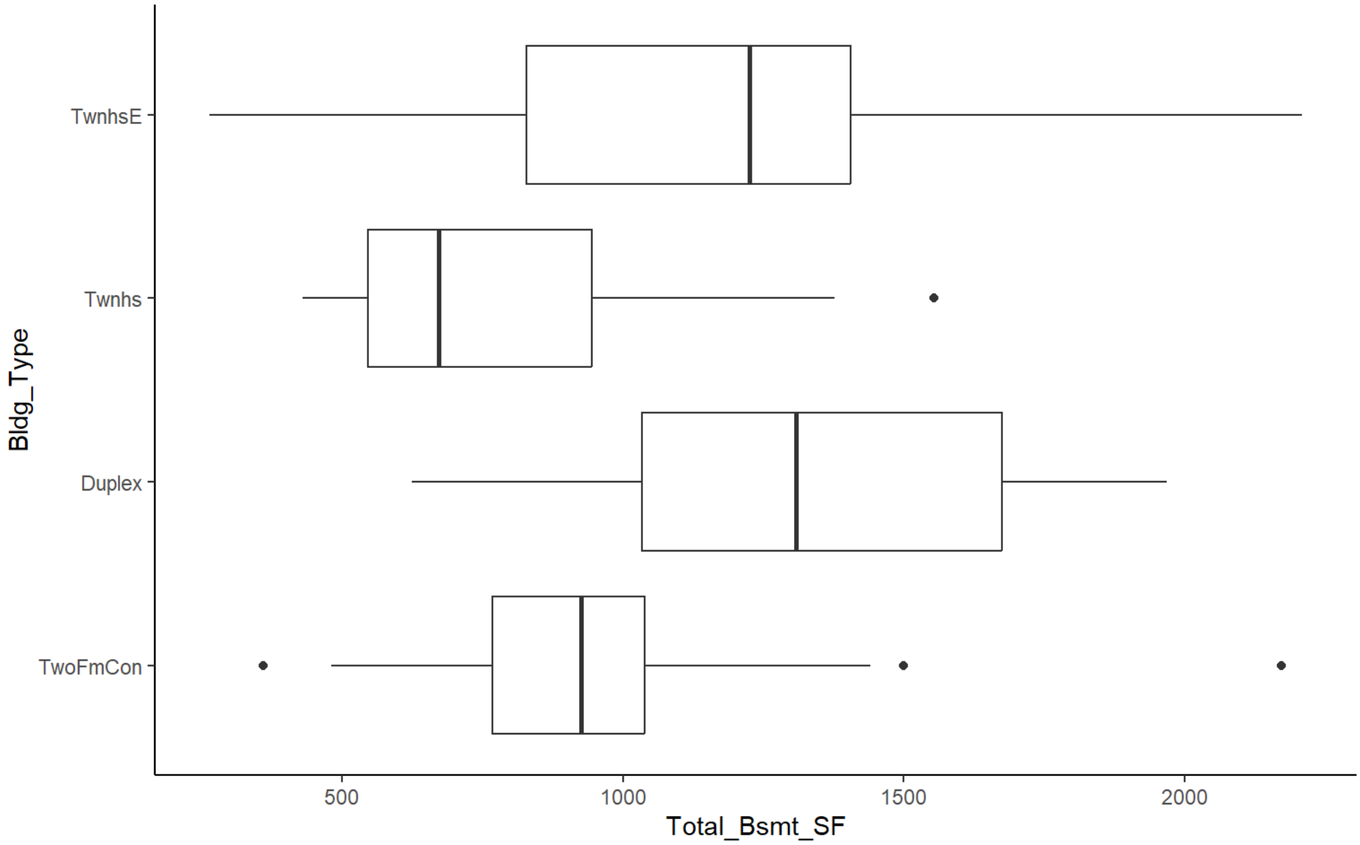
```
Floating_Village_Residential     Residential_High_Density 
                          62                           13 
     Residential_Low_Density   Residential_Medium_Density 
                         240                          157 
ALQ BLQ GLQ LwQ Rec Unf 
 66  25 220  22  28 111 
TwoFmCon   Duplex    Twnhs   TwnhsE 
      57       82      101      232 
```

2c)

Choose an appropriate plot to investigate the relationship between Bldg_Type and Total_Bsmt_SF in Ames2. (2 points)

A boxplot is the most appropriate plot to investigate this relationship as it shows the distribution of Total_Bsmt_SF based on the building type chosen. All other graphs (line, point, histogram, bar) are not suitable for this investigation.

```
Ames2 %>%
  select(Bldg_Type, Total_Bsmt_SF) %>%
  ggplot(mapping = aes(x = Total_Bsmt_SF, y = Bldg_Type)) +
  geom_boxplot() +
  theme_classic()
```



2d)

Choose an appropriate plot to investigate the relationship between Year_Built and Total_Bsmt_SF in Ames2. Color points according to the factor Bldg_Type. Ensure your plot has a clear title, axis labels and legend. What do you notice about how Basement size has changed over time? Were there any slowdowns in construction over this period? When? Can you think why? (4 points)

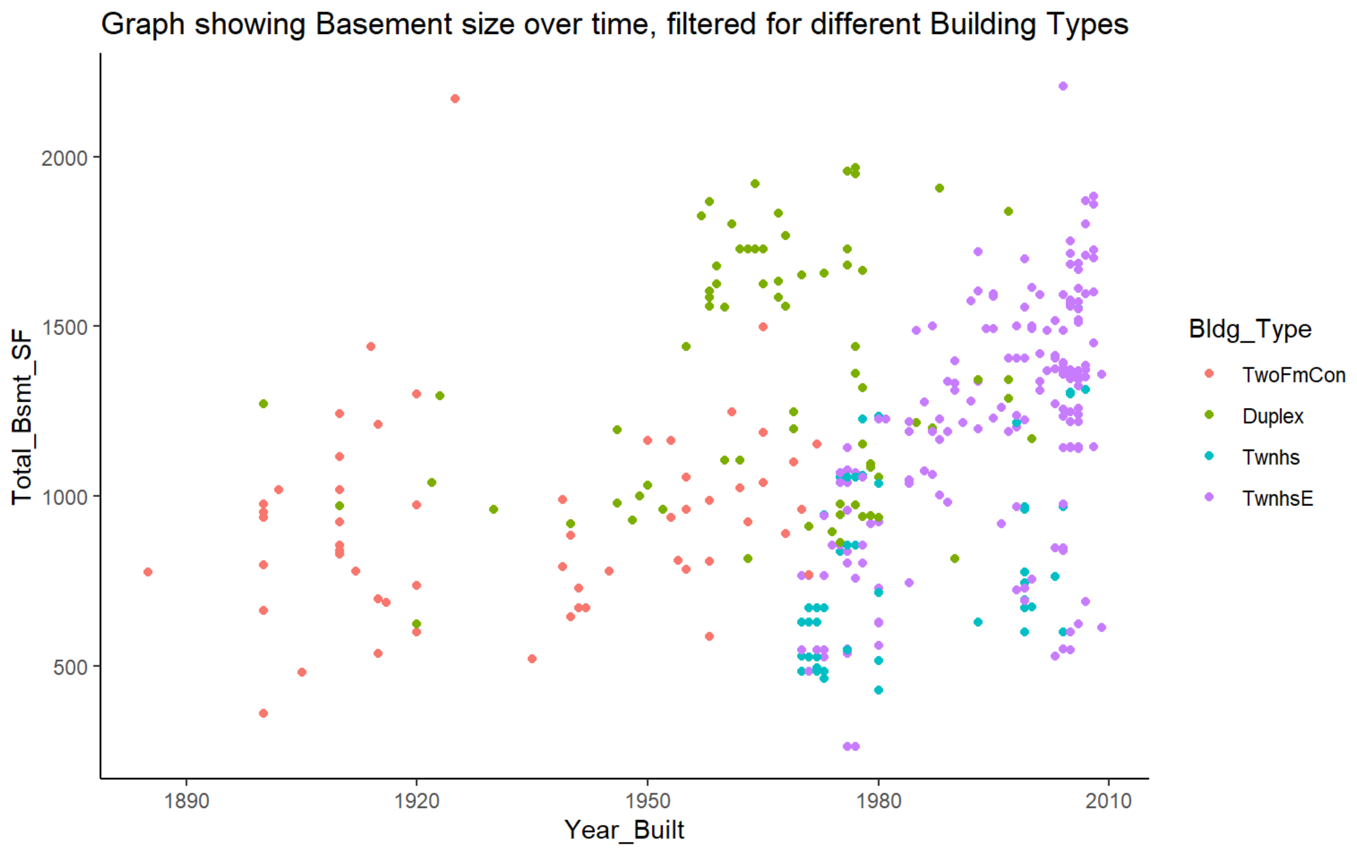
A scatter graph was chosen as the most appropriate plot to investigate the relationship between Year_Built & Total_Bsmt_SF as it shows how the basement size has varied over time from 1870 to 2010. The legend shows that the data is filtered by colour, to separate data by Building type to add extra depth into the graph and its analysis.

It is clear that the basement size has slowly been increasing over time as there is an increase in number of points with bigger basements after around 1980, as well as a large sum of houses with a basement size of 1000+ after the year 2000.

The slowdowns in construction during the time period of the data occurred during 1912 to 1945 as there are very few data points present in this time period, suggesting little to no construction occurring. This is likely due to the two appalling world wars that occurred: WW1 (1914 - 1918) & WW2 (1939 - 1945). Due to this men and women left Iowa and joined the war efforts in Europe, Asia, and Africa.

(Note: The Ames dataset records data for houses in Iowa.)

```
Ames2 %>%
  select(Year_Built, Total_Bsmt_SF, Bldg_Type) %>%
  ggplot(mapping = aes(x = Year_Built, y = Total_Bsmt_SF, colour = Bldg_Type)) +
  geom_point() +
  labs(title = "Graph showing Basement size over time, filtered for different Building Types"
) +
  theme_classic()
```



2e)

Why do we make these plots? Comment on your findings from these plots (1 sentence is fine). (2 points)

The above [plots 2c), 2d)] are used to investigate the relationship between 2 factors to see if they have any effect on each other, in which case perhaps one variable could be predicted if the other variable is known.

The findings from the boxplot show that the basement size is highly reliant on the building type, as basements tend to be smaller for a townhouse, and larger for a duplex.

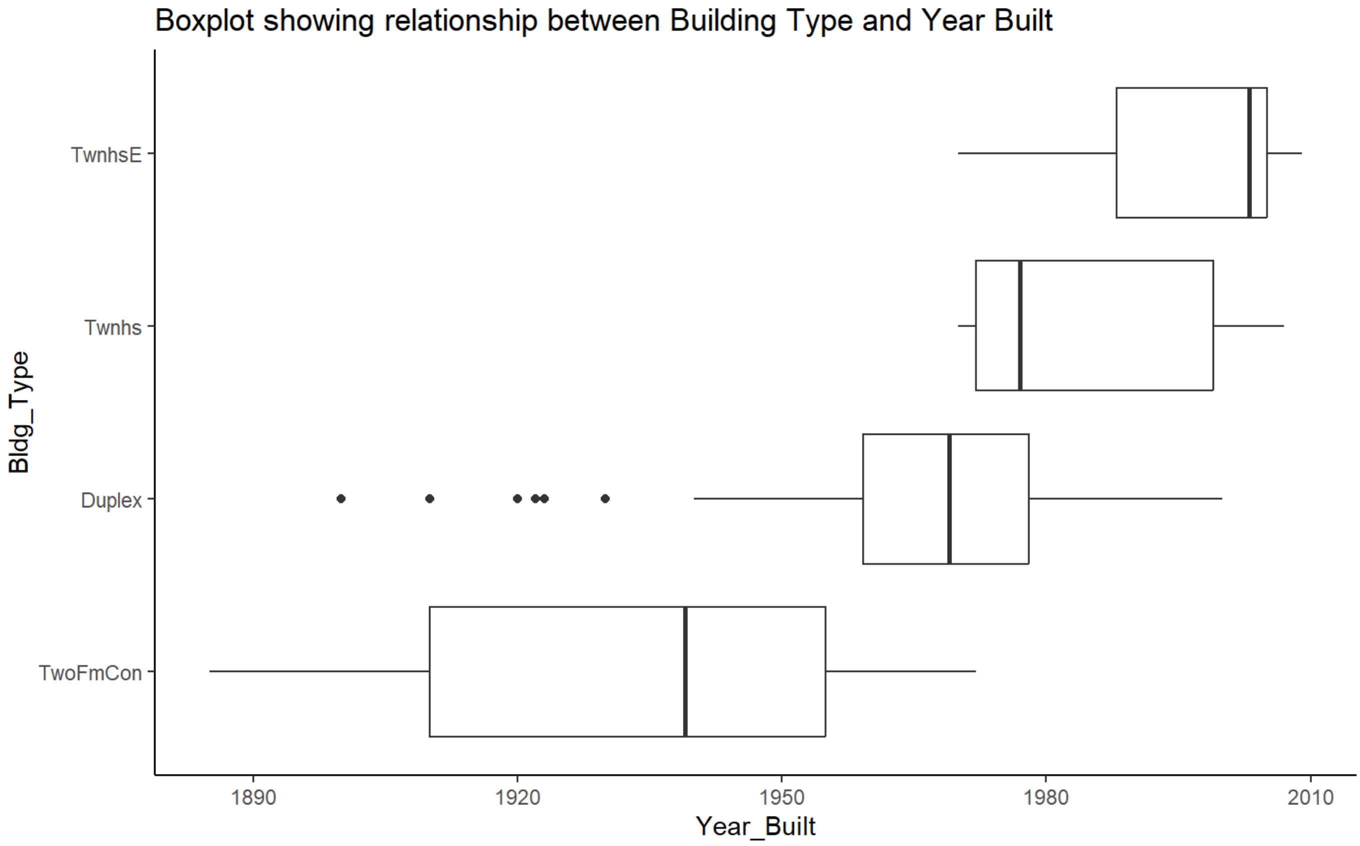
The findings from the scatter graph show that there is an increase in basement size over time, however, no TwoFmCon houses were built after 1970, after which most building types were TownHouseE which all had a relatively larger basement size.

2f)

Now choose an appropriate plot to investigate the relationship between Bldg_Type and Year_Built in Ames2. Why should we consider this? What do you notice? (3 points)

The boxplot below is an important plot as it shows when each building type was first introduced into construction, when it was the most popular building type being contructed, as well as when it was no longer being constructed. It is interesting to note that no two building types were highly popular to construct at one single time period. For example, a TwoFmCon was the first building type, being most popular during 1940, after which Duplex's began to go under construction. As Duplex buildings started increasing in construction, the TwoFmCon slowly stopped being built during 1960 (with the exception of a few outliers until 1965), as most new buildings being constructed were Duplex. This same trend follows accordingly for both Townhouse and TownhouseE as typically one building type is most popular at one singe time, hence will have more data points, increasing the count on the boxplot. TownhouseE was mainly built during 2000, altough the building type was introduced in 1970, however during 1970-2000 most buildings being constructed were Townhouses.

```
Ames2 %>%
  ggplot(mapping = aes(x = Year_Built, Bldg_Type)) +
  geom_boxplot() +
  labs(title = "Boxplot showing relationship between Building Type and Year Built") +
  theme_classic()
```



2g)

Use the lm command to build a linear model, linmod1, of Total_Bsmt_SF as a function of the predictors Bldg_Type and Year_Built for the “Ames2” dataset. (2 points)

```
linmod1 <-
  lm(Total_Bsmt_SF~Bldg_Type + Year_Built, data = Ames2)

linmod1
summary(linmod1)
```

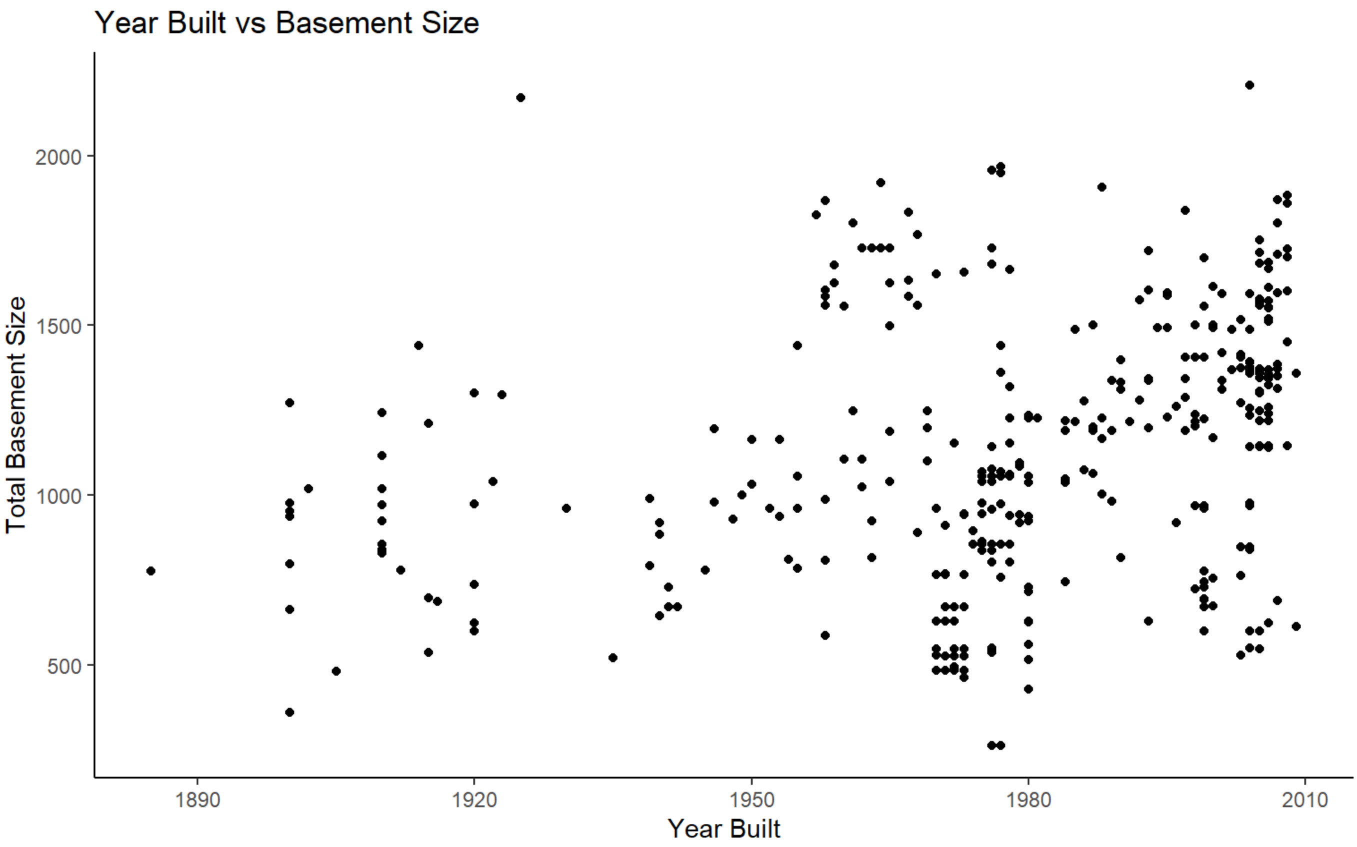
```


Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built, data = Ames2)

Coefficients:
    (Intercept)  Bldg_TypeDuplex   Bldg_TypeTwnhs  Bldg_TypeTwnhsE  
     -12929.190          186.983         -531.366         -234.859  
     Year_Built  
          7.166  



Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built, data = Ames2)

Residuals:
    Min      1Q  Median      3Q     Max 
-738.53 -223.35    7.68  238.36 1306.23 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -1.293e+04  1.833e+03  -7.054 6.30e-12 ***
Bldg_TypeDuplex  1.870e+02  6.504e+01   2.875  0.00422 ** 
Bldg_TypeTwnhs  -5.314e+02  7.252e+01  -7.327 1.04e-12 ***
Bldg_TypeTwnhsE -2.349e+02  7.678e+01  -3.059  0.00235 ** 
Year_Built       7.166e+00  9.478e-01   7.560 2.15e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 327.1 on 467 degrees of freedom
Multiple R-squared:  0.3339,    Adjusted R-squared:  0.3282 
F-statistic: 58.54 on 4 and 467 DF,  p-value: < 2.2e-16
```

2h)

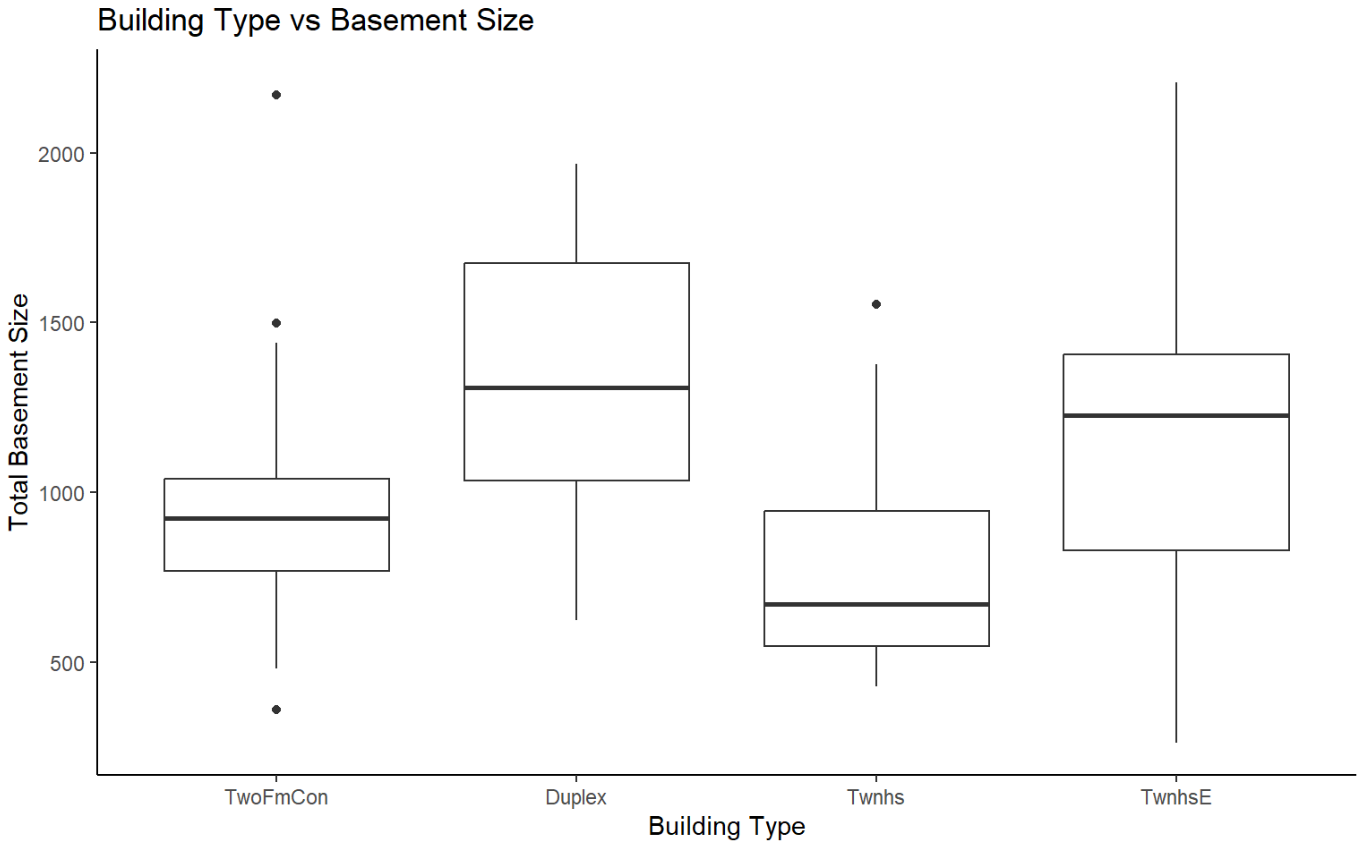
State and evaluate the assumptions of the model. (6 points)

To ensure that the model created is reasonable, it must be evaluated to assess the following assumptions: Linearity, Homoscedasticity, and Normality.

```
Ames2 %>%
  ggplot(mapping = aes(x = Year_Built, y = Total_Bsmt_SF)) +
  geom_point() +
  labs(x = "Year Built",
       y = "Total Basement Size") +
  labs(title = "Year Built vs Basement Size") +
  theme_classic()
```



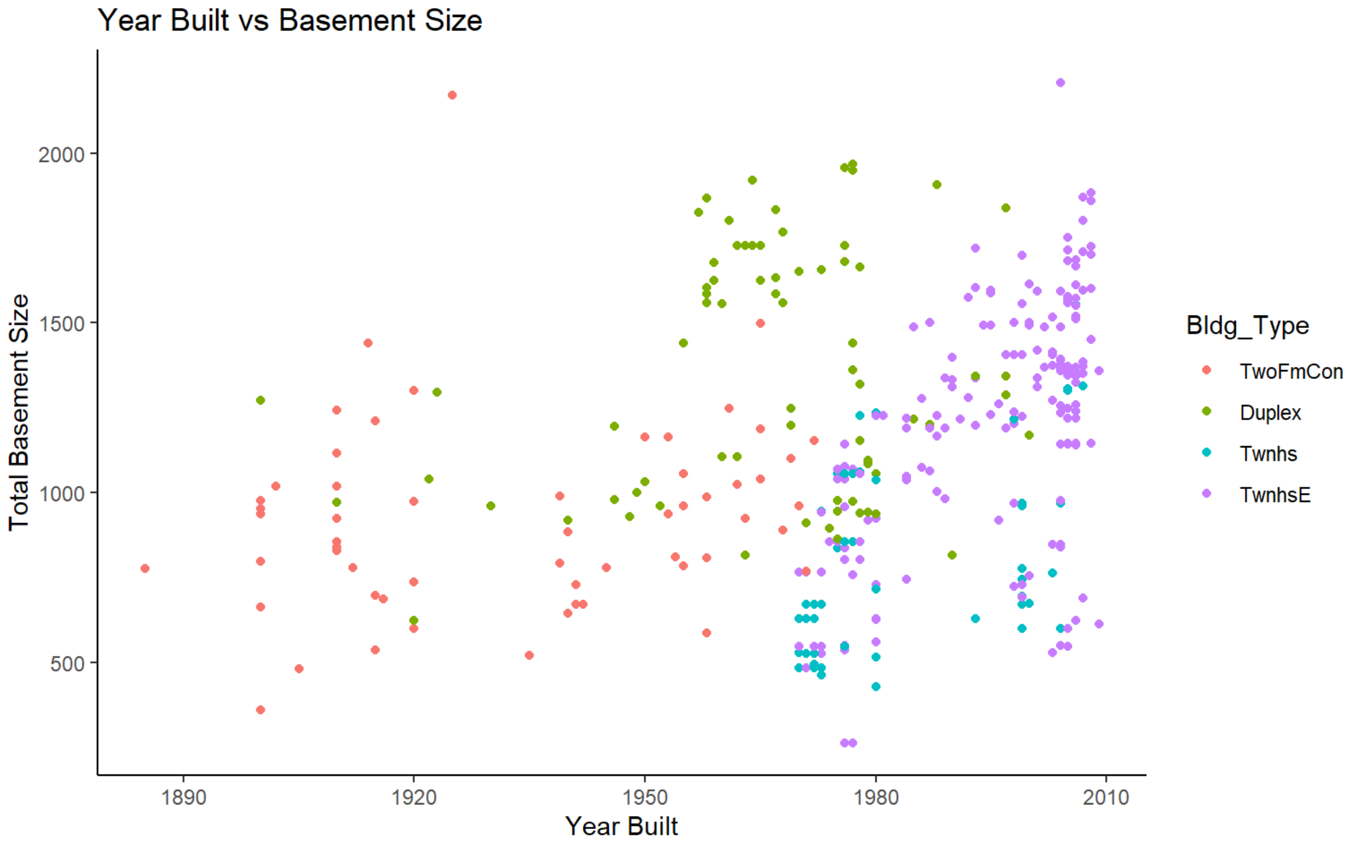
```
Ames2 %>%
  ggplot(mapping = aes(x = Bldg_Type, y = Total_Bsmt_SF)) +
  geom_boxplot() +
  labs(x = "Building Type",
       y = "Total Basement Size") +
  labs(title = "Building Type vs Basement Size") +
  theme_classic()
```



Firstly, to test linearity, a plot can be made for each predictor (Building type & Year built) against Basement size (response variable). The first figure above (Year Built vs Basement size) has a roughly linear trend to it, allowing us to assume that a linear model is somewhat reasonable. The graph can be made more linear using log values although this will not be done here.

The second figure (Building Type vs Basement size) is difficult to assess linearity from as it makes use of categorical data. Instead, the two graphs can be combined, and Bldg_Type can be assigned to colour as follows (same as figure in Q2d):

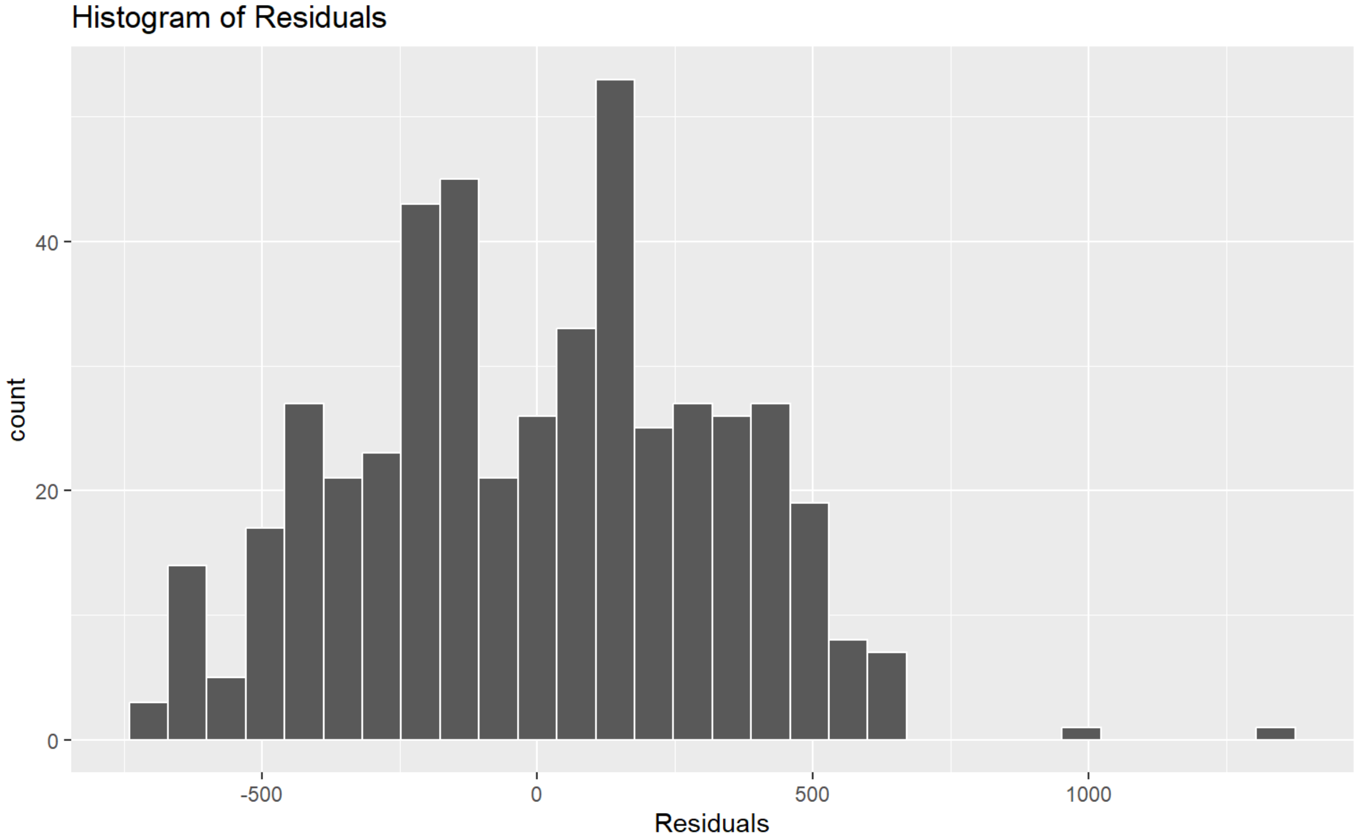
```
Ames2 %>%
  ggplot(mapping = aes(x = Year_Built, y = Total_Bsmt_SF, colour = Bldg_Type)) +
  geom_point() +
  labs(x = "Year Built",
       y = "Total Basement Size") +
  labs(title = "Year Built vs Basement Size") +
  theme_classic()
```

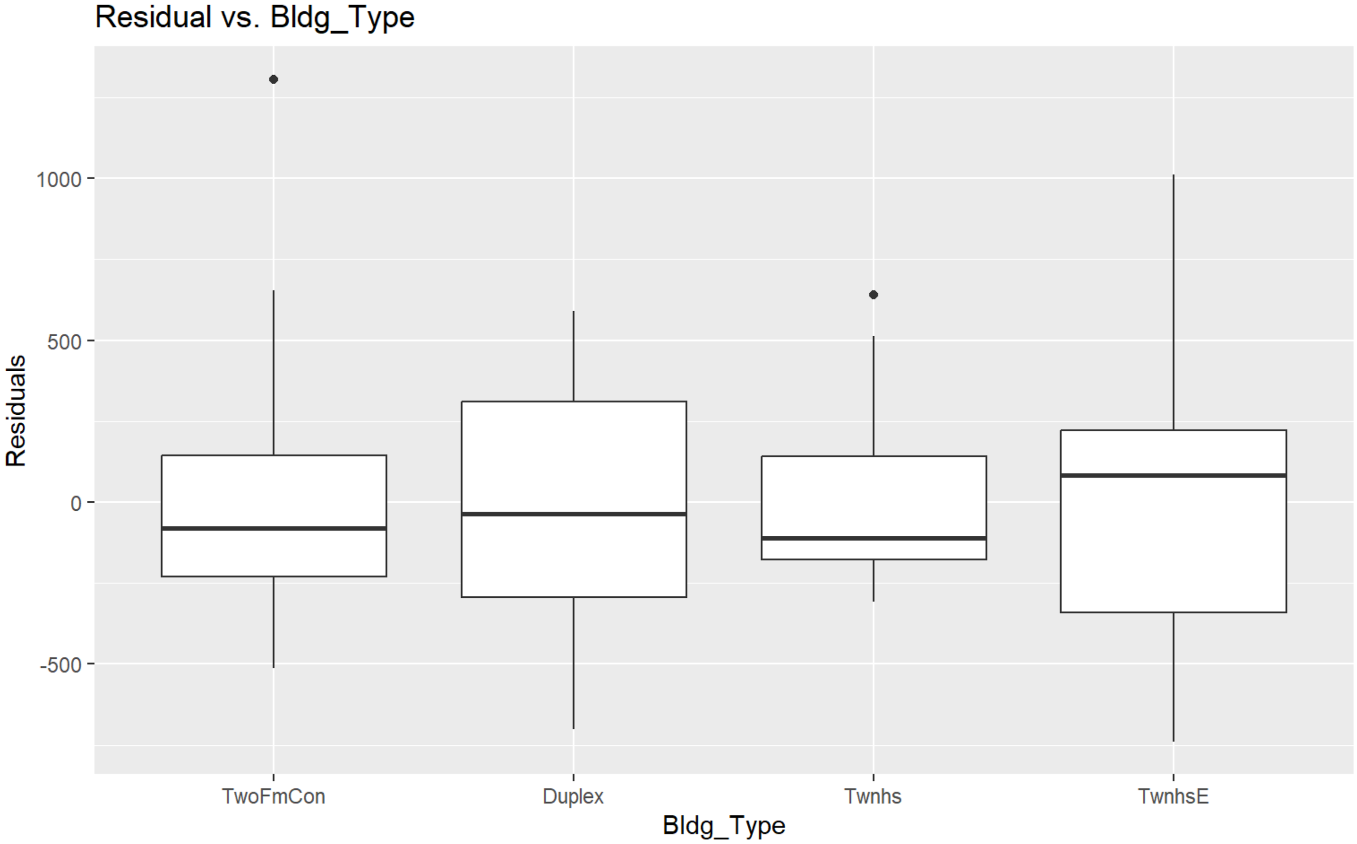


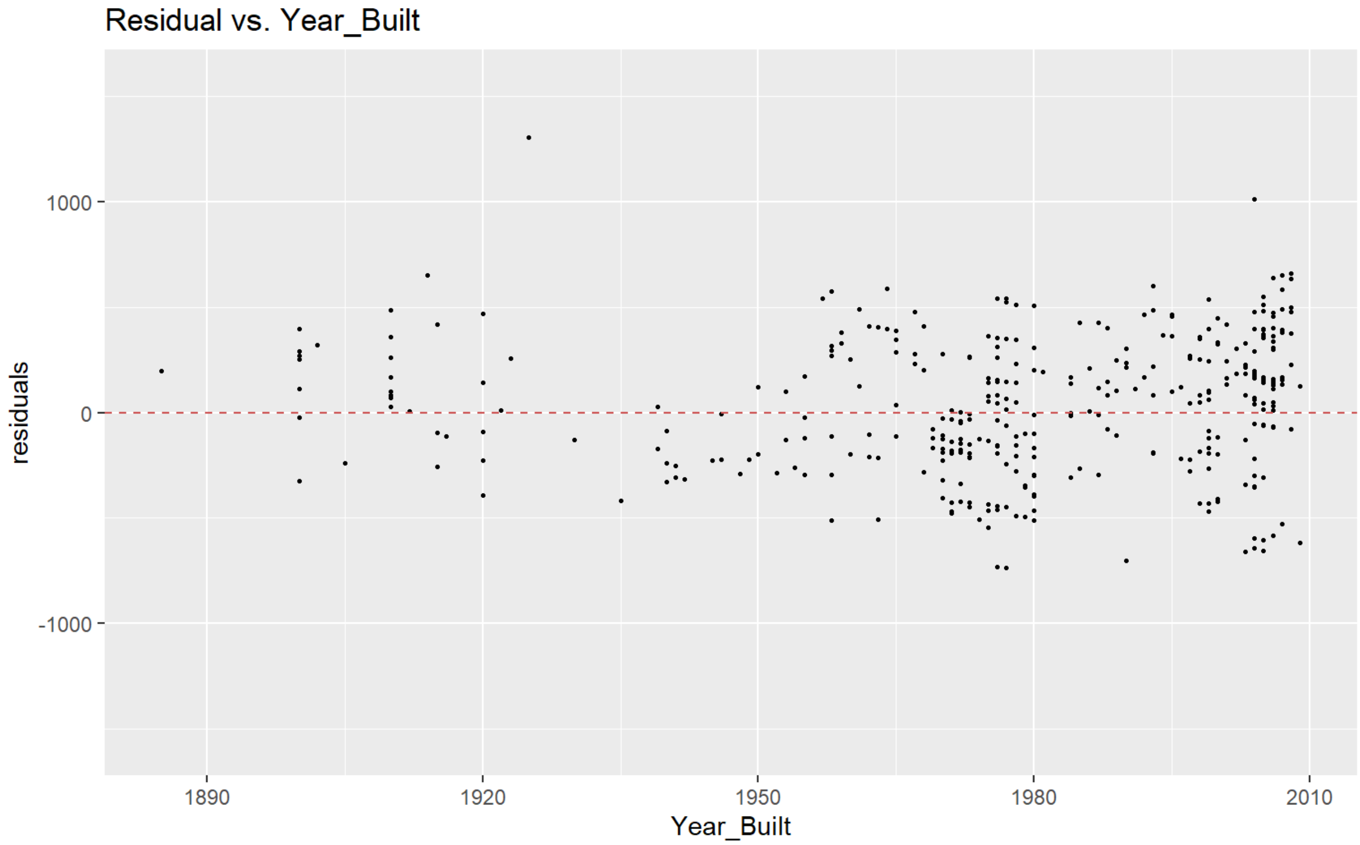
The plot above shows a relatively linear trend upwards, as well as a large proportion of the data points being situated on the right hand side of the graph. This suggests that it could be seen as an exponential relationship rather than linear too, although it is up to interpretation. To conclude, the assumption of linearity is barely correct for this model, as there is linearity in the figure above, although not perfect.

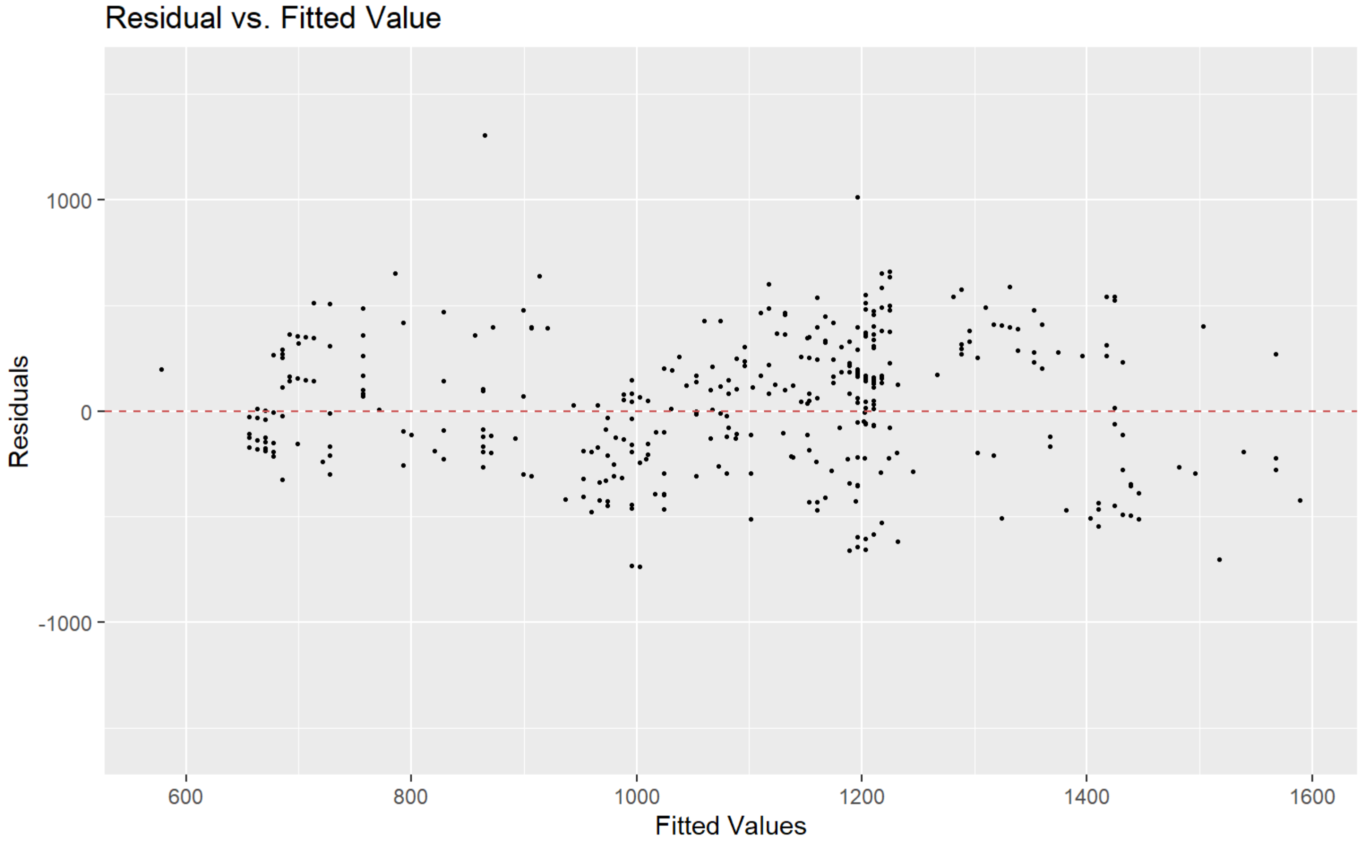
To do a more in depth analysis for the remaining 2 assumptions, the following code is used to generate a set of figures:

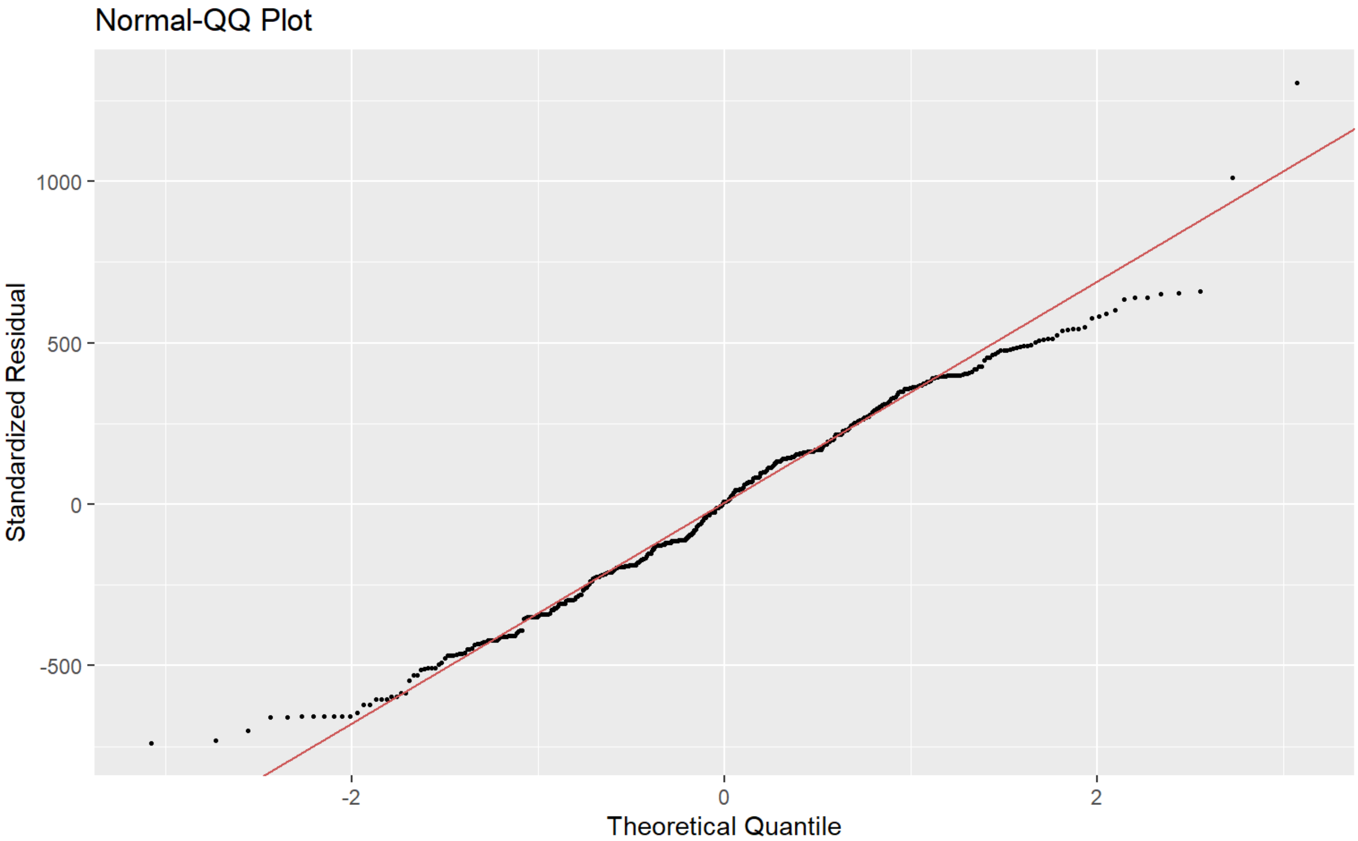
```
linmod1 %>%
  gg_diagnose(max.per.page = 1)
```

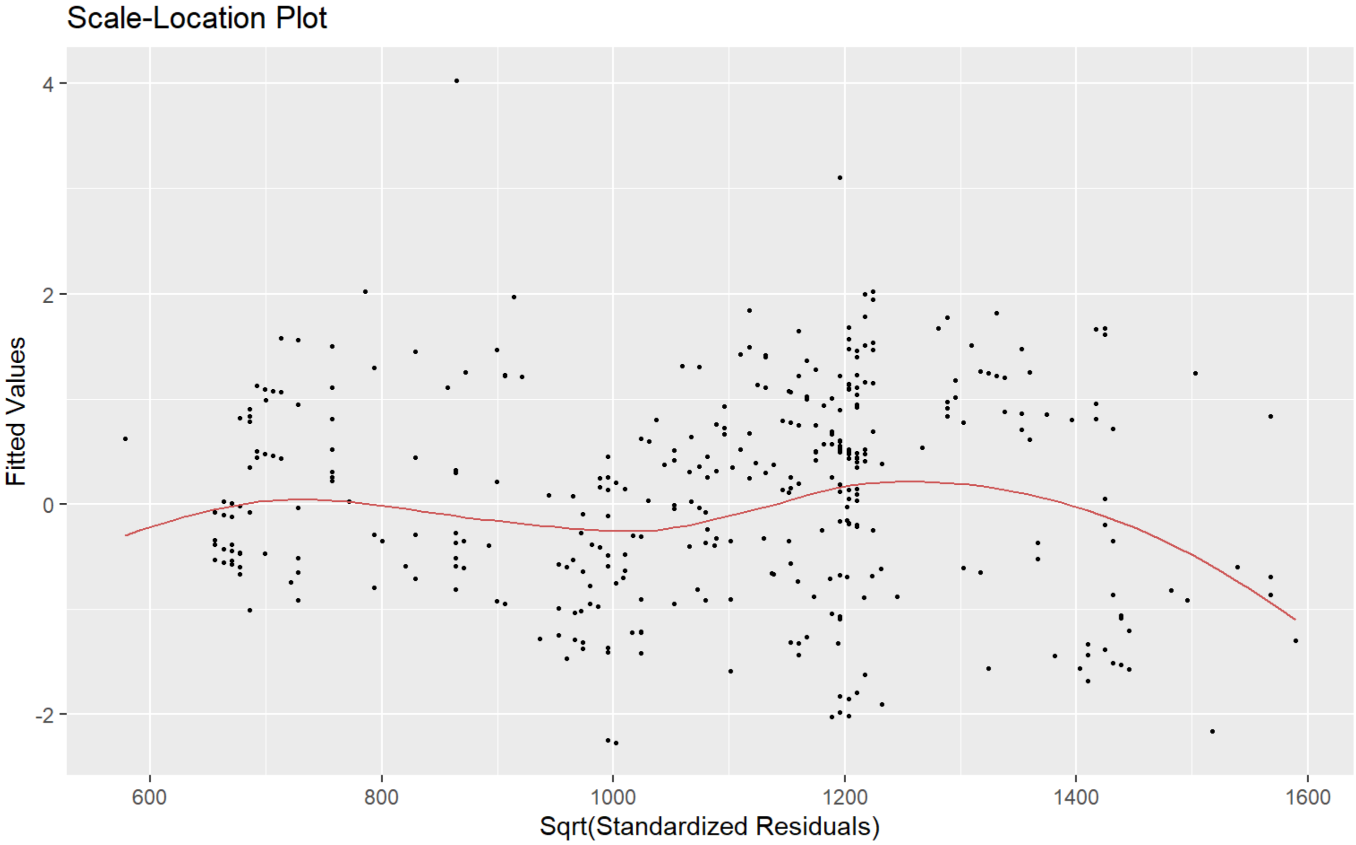

Histogram of Residuals
count
40
20
0
-500
0
500
1000
Residuals

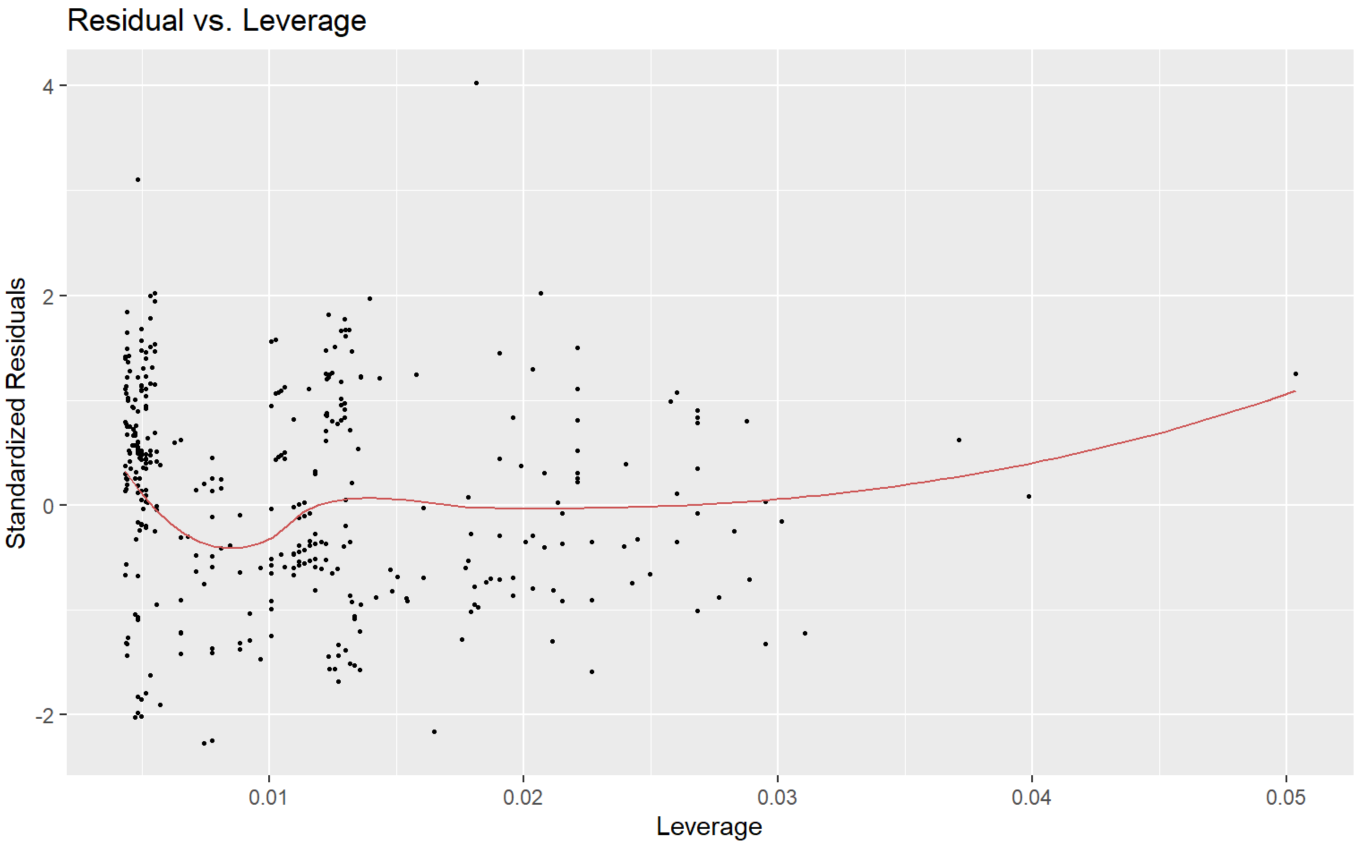

Residual vs. Bldg_Type
Residuals
1000
500
0
-500
TwoFmCon
Duplex
Twnhs
TwnhsE
Bldg_Type

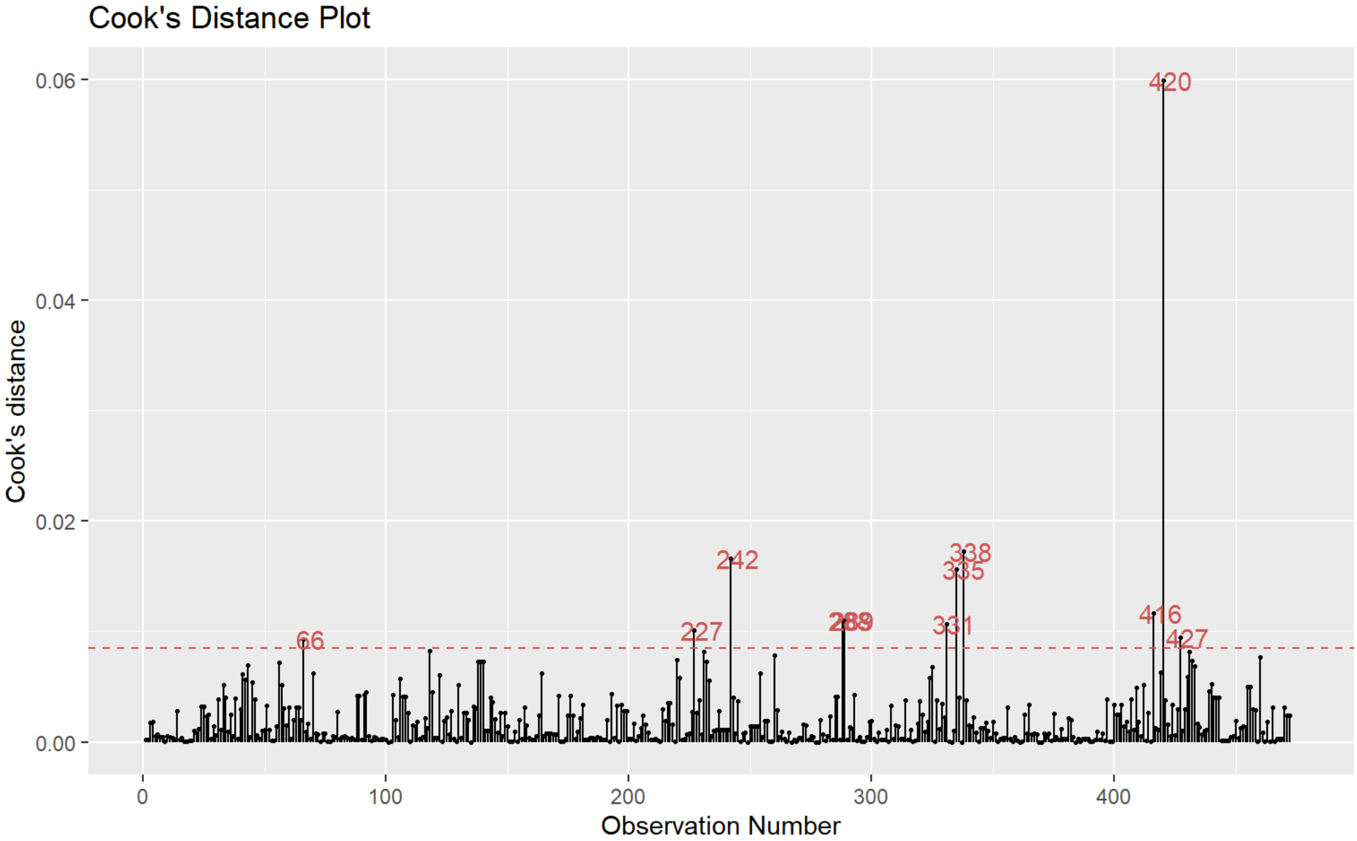

Residual vs. Year_Built
residuals
1000
0
-1000
1890
1920
1950
1980
2010
Year_Built


Residual vs. Fitted Value
Residuals
1000
0
-1000
600
800
1000
1200
1400
1600
Fitted Values


Normal-QQ Plot
Standardized Residual
1000
500
0
-500
-2
0
2
Theoretical Quantile


Scale-Location Plot
Fitted Values
4
2
0
-2
600
800
1000
1200
1400
1600
Sqrt(Standardized Residuals)





The next assumption is Homoscedasticity. This makes use of the graphs plotting residuals against predictors (Building Type & Year Built). In the figure plotting Residuals vs Year Built, that scatter is roughly the same width along the x-axis, although there is an indication for an upwards trend due to the number of data points on right hand side of the graph. The spread is not very even as most data points are towards 1970+. Although not perfect, the graph still suggests that the homoscedasticity assumption has been satisfied.

The figure plotting Residuals vs Building Type is not great and is not perfectly in line with the model assumptions as the box plot interquartile ranges are quite different, with some box plots being smaller than others. For the assumption to have been satisfied, the interquartile ranges of each category needed to be similar/same sizes (which is not the case here).

Therefore, while the ‘Residuals vs Year Built’ figure is passable in confirming the assumption of Homoscedasticity, the ‘Residuals vs Bldg_Type’ figure does not satisfy the assumption.

The third key assumption is Normality which can be assessed using two of the figures above:

1. Using the histogram of residuals, it resembles a Gaussian distribution model following a bell curve shape. However, while a bell shape curve peaks in the middle of the x-axis, the residuals in the figure above drop at this point, weakening the model assumption of Normality. This model barely confirms the assumption of Normality as it does resemble a Gaussian model for the most part.
2. Using the qqplot of residuals, the model data is strongly in line with the Normality assumption as the qqplot line is linear, although there is a slight tail towards the start and end of the data set. Overall, this figure supports the model assumption as the qqplot is linear for majority of the data set.

2i)

Use the lm command to build a second linear model, linmod2, for Total_Bsmt_SF as a function of Bldg_Type, Year_Built and Lot_Area. (2 points)

```
linmod2 <-
  lm(Total_Bsmt_SF~Bldg_Type + Year_Built + Lot_Area, data = Ames2)

linmod2
summary(linmod2)
```

```
Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built + Lot_Area, 
    data = Ames2)

Coefficients:
    (Intercept)  Bldg_TypeDuplex   Bldg_TypeTwnhs  Bldg_TypeTwnhsE  
     -1.176e+04        2.378e+02       -4.120e+02       -1.265e+02  
     Year_Built         Lot_Area  
      6.509e+00        7.793e-03  



Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built + Lot_Area, 
    data = Ames2)

Residuals:
    Min      1Q  Median      3Q     Max 
-810.32 -212.07   -5.72  233.88 1232.65 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -1.176e+04  1.828e+03  -6.435 3.08e-10 ***
Bldg_TypeDuplex  2.378e+02  6.529e+01   3.642 0.000301 ***
Bldg_TypeTwnhs  -4.120e+02  7.745e+01  -5.319 1.62e-07 ***
Bldg_TypeTwnhsE -1.265e+02  8.035e+01  -1.575 0.115942    
Year_Built       6.509e+00  9.476e-01   6.868 2.09e-11 ***
Lot_Area         7.793e-03  1.960e-03   3.977 8.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 322.1 on 466 degrees of freedom
Multiple R-squared:  0.3558,	Adjusted R-squared:  0.3489 
F-statistic: 51.48 on 5 and 466 DF,  p-value: < 2.2e-16
```

2j)

Use Analysis of variance (ANOVA) and Adjusted R-squared to compare these two models, and decide which is a better model. (6 points)

```
Anova(linmod1)
Anova(linmod2)
```

```
Anova Table (Type II tests)

Response: Total_Bsmt_SF
             Sum Sq  Df F value    Pr(>F)    
Bldg_Type  21391017   3  66.624 < 2.2e-16 ***
Year_Built  6117408   1  57.159 2.147e-13 ***
Residuals  49980160 467                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Anova Table (Type II tests)

Response: Total_Bsmt_SF
             Sum Sq  Df F value    Pr(>F)    
Bldg_Type  16215684   3  52.107 < 2.2e-16 ***
Year_Built  4893518   1  47.174 2.086e-11 ***
Lot_Area    1640455   1  15.814 8.099e-05 ***
Residuals  48339705 466                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Carrying out an ANOVA test on linmod1 obtains a p-value for each variable in the model (Bldg_Type, Year_Built). Both of these values are extremely small (2.2^(-16) and 2.147^(-13)), suggesting that this full model is a lot better than a null hypothesis model which does not make use of predictors.

Similarly, the ANOVA test on linmod2 also obtains extremely small p-values for each variable in the model (Bldg_Type, Year_Built, Lot_Area) with p-values of 2.2^(-16), 2.086^(-11), 8.099^(-5) respectively. Extremely small p-values here shows we have good evidence again, that this model is an improvement over a null model with no predictors.

```
anova(linmod1, linmod2)
```

```
Analysis of Variance Table

Model 1: Total_Bsmt_SF ~ Bldg_Type + Year_Built
Model 2: Total_Bsmt_SF ~ Bldg_Type + Year_Built + Lot_Area
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)    
1    467 49980160                                  
2    466 48339705  1   1640455 15.814 8.099e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To compare the 2 models using ANOVA, the above code is used to generate a single p-value that is the test comparing the null hypothesis of linmod1 against the alternative hypothesis of linmod2. The resulting p-value is 8.099^(-5) which is also an extremely small p-value, which suggests that the model (linmod2) which accounts for 3 predictors is a more accurate model to estimate the Basement size than linmod1 which only makes use of 2 predictors.

Note: The p-value (in anova(linmod1,linmod2)) is actually equal to the p-value for Lot_Area in linmod2, as the addition of this predictor is the only difference between the two models.

If the p-value would have been >0.05, the opposite conclusion would have been made and the extra predictor would not have been a useful addition to the model.

Next, the two models are compared by using their Adjusted R squared values:

```
summary(linmod1)
summary(linmod2)
```

```

Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built, data = Ames2)

Residuals:
    Min      1Q  Median      3Q     Max 
-738.53 -223.35    7.68  238.36 1306.23 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -1.293e+04  1.833e+03  -7.054 6.30e-12 ***
Bldg_TypeDuplex  1.870e+02  6.504e+01   2.875  0.00422 ** 
Bldg_TypeTwnhs  -5.314e+02  7.252e+01  -7.327 1.04e-12 ***
Bldg_TypeTwnhsE -2.349e+02  7.678e+01  -3.059  0.00235 ** 
Year_Built       7.166e+00  9.478e-01   7.560 2.15e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 327.1 on 467 degrees of freedom
Multiple R-squared:  0.3339,	Adjusted R-squared:  0.3282 
F-statistic: 58.54 on 4 and 467 DF,  p-value: < 2.2e-16


Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built + Lot_Area, 
    data = Ames2)

Residuals:
    Min      1Q  Median      3Q     Max 
-810.32 -212.07   -5.72  233.88 1232.65 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -1.176e+04  1.828e+03  -6.435 3.08e-10 ***
Bldg_TypeDuplex  2.378e+02  6.529e+01   3.642 0.000301 ***
Bldg_TypeTwnhs  -4.120e+02  7.745e+01  -5.319 1.62e-07 ***
Bldg_TypeTwnhsE -1.265e+02  8.035e+01  -1.575 0.115942    
Year_Built       6.509e+00  9.476e-01   6.868 2.09e-11 ***
Lot_Area         7.793e-03  1.960e-03   3.977 8.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 322.1 on 466 degrees of freedom
Multiple R-squared:  0.3558,	Adjusted R-squared:  0.3489 
F-statistic: 51.48 on 5 and 466 DF,  p-value: < 2.2e-16
```

Linmod1, which estimates Basement size based on Building Type and Year Built has an adjusted R squared value of 0.3282, suggesting that 32.82% of the variance in basement size is explained by the differences in the two mentioned predictor variables.

Linmod 2, which estimates Basement size based on Building Type, Year Built, and Lot Area has an adjusted R squared value of 0.3489, suggesting that 34.89% of the variance in basement size is explained by the differences in the three mentioned predictor variables.

Therefore, linmod2 which considers more predictors, is a slightly better model than linmod1 as it is has a greater adjusted R squared value so is a more explanatory model (by just 2.07%).

2k)

Construct a confidence interval and a prediction interval for the basement area of a Twnhs built in 1980, with a lot Area of 7300. Explain what these two intervals mean. (6 points)

The model considers Basement area based on 3 predictor variables:

1. Bldg_Type = Townhouse 2. Year_Built = 1980 3. Lot_Area = 7300

Therefore, it is not possible to create nice plots of prediction and confidence intervals on a graph. However, both of these can still be calculated for the model as follows.

To calculate the confidence interval, the following code is used:

```
predict(linmod2, newdata = data.frame(Bldg_Type = "Twnhs", Year_Built = 1980, Lot_Area = 7300
), interval = "confidence")
```

```
       fit      lwr      upr
1 768.7589 702.1423 835.3755
```

To calculate the prediction interval, the following code is used:

```
predict(linmod2, newdata = data.frame(Bldg_Type = "Twnhs", Year_Built = 1980, Lot_Area = 7300
), interval = "prediction")
```

```
       fit      lwr      upr
1 768.7589 132.3605 1405.157
```

The fitted value is the same for both the prediction and interval models, as this value shows the resulting Basement size (obtained from the model) to be 768.759 for a Townhouse built in 1980, with a Lot Area of 7300.

The lwr (702.142) and upr (835.376) values for the confidence band show that although the coefficients in the model have resulted in the fitted value, the data we observed could also have come from any other regression line that could fit inside the confidence interval region between 702.142 and 835.376. It shows the mean value for basement size and an acceptable value range based on the model.

Most data points are likely not within the confidence interval as this only represents the uncertainty around the mean. To calculate the range that most observations are present in, a prediction interval is used instead as it will take into account the uncertainty in the estimate of the mean, and the variance in the residuals.

The lwr (702.142) and upr (835.376) values for the prediction band show the range of values in which most of the data points are present within. This band is much wider than the confidence band as it covers most data points in the data set for this set of variable specifications (Townhouse, Year 1980, Lot Area of 7300.

2l)

Now build a linear mixed model, linmod3, for Total_Bsmt_SF as a function of Year_Built,MS_Zoning and Bldg_Type. Use Neighborhood as random effect. What is the critical number to pull out from this, and what does it tell us? (4 points)

```
linmod3 <-
  lmer(Total_Bsmt_SF~Year_Built + MS_Zoning + Bldg_Type +  (1|Neighborhood), data = Ames2)
linmod3
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
Total_Bsmt_SF ~ Year_Built + MS_Zoning + Bldg_Type + (1 | Neighborhood)
   Data: Ames2
REML criterion at convergence: 6566.758
Random effects:
 Groups       Name        Std.Dev.
 Neighborhood (Intercept) 187.4   
 Residual                 261.8   
Number of obs: 472, groups:  Neighborhood, 27
Fixed Effects:
                        (Intercept)                           Year_Built  
                          -4890.652                                2.876  
  MS_ZoningResidential_High_Density     MS_ZoningResidential_Low_Density  
                            148.504                              288.369  
MS_ZoningResidential_Medium_Density                      Bldg_TypeDuplex  
                            109.234                              264.530  
                     Bldg_TypeTwnhs                      Bldg_TypeTwnhsE  
                            -63.140                              105.171  
```

Using the fixed effects values and variables, an equation can be derived to estimate Total_Bsmt_SF.

The critical value that can be pulled from this is the standard deviation (τ), which shows the effect of the neighborhood being 187.4. This shows that the effect of the neighborhood contributes 187.4 to the overall Basement size. It is evident that this value is of large significance to the Basement size as it has a similar value to the fixed effect values for MS_Zoning & Bldg_Type.

The other key critical value is the residual standard deviation (σ) which has a value of 261.8, hence, further emphasising the importance of Neighborhood as a parameter to consider when looking into Basement sizes.

2m)

Construct 95% confidence intervals around each parameter estimate for linmod3. What does this tell us about the significance of the random effect? (3 points)

```
confint(linmod3)
```

```
                                           2.5 %     97.5 %
.sig01                              114.916972  253.19244
.sigma                              244.221572  278.77404
(Intercept)                       -9699.022207 -595.99553
Year_Built                            0.691417    5.33084
MS_ZoningResidential_High_Density  -254.190137  549.38121
MS_ZoningResidential_Low_Density    -91.829020  665.77648
MS_ZoningResidential_Medium_Density -266.136920  487.72182
Bldg_TypeDuplex                     145.073466  377.00462
Bldg_TypeTwnhs                     -249.148897  102.22240
Bldg_TypeTwnhsE                     -68.266514  263.18536
```

The confidence intervals of the random effect (Neighborhood) represented by .sig01 (standard deviation of the random effect) & .sigma (standard deviation of the residuals) had a very small interval range, increasing its accuracy and reliability.

Neither of these confidence intervals contain zero as a value, hence, they are both “significant” at the 95% confidence interval level, as well as being a very important factor in its effects to Basement size.

2n)

Write out the full mathematical expression for the model in linmod2 and for the model in linmod3. Round to the nearest integer in all coefficients with modulus (absolute value) > 10 and to three decimal places for coefficients with modulus < 10. (4 points)

```
summary(linmod2)
```

```


Call:
lm(formula = Total_Bsmt_SF ~ Bldg_Type + Year_Built + Lot_Area,
    data = Ames2)

Residuals:
    Min      1Q  Median      3Q     Max
-810.32 -212.07   -5.72  233.88 1232.65

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.176e+04  1.828e+03  -6.435 3.08e-10 ***
Bldg_TypeDuplex  2.378e+02  6.529e+01   3.642 0.000301 ***
Bldg_TypeTwnhs  -4.120e+02  7.745e+01  -5.319 1.62e-07 ***
Bldg_TypeTwnhsE -1.265e+02  8.035e+01  -1.575 0.115942
Year_Built       6.509e+00  9.476e-01   6.868 2.09e-11 ***
Lot_Area         7.793e-03  1.960e-03   3.977 8.10e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 322.1 on 466 degrees of freedom
Multiple R-squared:  0.3558,    Adjusted R-squared:  0.3489
F-statistic: 51.48 on 5 and 466 DF,  p-value: < 2.2e-16
```

$$\begin{aligned}\mathrm{E}(\mathrm{Total\_Bsmt\_SF}) = -11760 + 238 \times isDuplex \\ -412 \times isTwnhs - 127 \times isTwnhsE \\ +6.509 \times \mathrm{Year\_Built} + 0.008 \times \mathrm{Lot\_Area}\end{aligned}$$

$$\mathrm{Total\_Bsmt\_SF} \sim N(\mathrm{E}(\mathrm{Total\_Bsmt\_SF}), 322)$$

```
summary(linmod3)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
Total_Bsmt_SF ~ Year_Built + MS_Zoning + Bldg_Type + (1 | Neighborhood)
   Data: Ames2

REML criterion at convergence: 6566.8

Scaled residuals: 
    Min      1Q  Median      3Q     Max 
-2.1502 -0.5804 -0.0394  0.6330  4.4359 

Random effects:
 Groups       Name        Variance Std.Dev.
 Neighborhood (Intercept) 35128    187.4   
 Residual                 68517    261.8   
Number of obs: 472, groups:  Neighborhood, 27

Fixed effects:
                                     Estimate Std. Error t value
(Intercept)                         -4890.652   2262.148  -2.162
Year_Built                              2.876      1.151   2.499
MS_ZoningResidential_High_Density     148.504    211.630   0.702
MS_ZoningResidential_Low_Density      288.369    198.824   1.450
MS_ZoningResidential_Medium_Density   109.234    197.814   0.552
Bldg_TypeDuplex                       264.530     59.046   4.480
Bldg_TypeTwnhs                        -63.140     87.020  -0.726
Bldg_TypeTwnhsE                       105.171     83.438   1.260

Correlation of Fixed Effects:
            (Intr) Yr_Blt MS_ZR_H MS_ZR_L MS_ZR_M Bld_TD Bld_TT
Year_Built  -0.996                                             
MS_ZnnR_H_D -0.158  0.081                                      
MS_ZnnR_L_D -0.169  0.085  0.912                               
MS_ZnnR_M_D -0.142  0.061  0.901   0.963                       
Bldg_TypDpl  0.378 -0.395  0.014  -0.017  -0.001               
Bldg_TypTwn  0.546 -0.570  0.004   0.059  -0.006   0.617       
Bldg_TypTwE  0.602 -0.626 -0.005   0.052  -0.010   0.662  0.911
```

$$
\begin{aligned}
\mathrm{E}(\mathrm{Total\_Bsmt\_SF}) = -4891 + 2.876 \times \mathrm{Year\_Built} \\
+149 \times isHighDensity + 288 \times isLowDensity \\
+109 \times isMediumDensity + 265 \times isDuplex \\
-63 \times isTwnhs + 105 \times isTwnhsE \\
+U
\end{aligned}
$$

$$U \sim N(0, 35128)$$

$$\mathrm{Total\_Bsmt\_SF} \sim N(E(\mathrm{Total\_Bsmt\_SF}), 68517)$$

3. Logistic Regression

3ai)

Create a new dataset called “Ames3” that contains all data in “Ames” dataset plus a new variable “excellent_heating” that indicates if the heating quality and condition “Heating_QC” is excellent or not. (2 points)

```
summary(Ames$Heating_QC)

Ames3 <-
  Ames %>%
  mutate(excellent_heating = case_when(
    Heating_QC == "Excellent" ~ "Excellent",
    Heating_QC == "Fair" ~ "Not_Excellent",
    Heating_QC == "Good" ~ "Not_Excellent",
    Heating_QC == "Poor" ~ "Not_Excellent",
    Heating_QC == "Typical" ~ "Not_Excellent"
  )) %>%
   mutate_at(vars(excellent_heating),
            list(factor))

summary(Ames3$excellent_heating)
```

```
Excellent      Fair      Good      Poor   Typical 
     1495        92       476         3       864 
    Excellent Not_Excellent 
         1495          1435 
```

The summary codes show that initially there were 5 measures of quality, but this has been changed to two factors which are excellent & not_excellent.

3aii)

In “Ames3” dataset, remove all cases “3” and “4” corresponding to the Fireplaces variable.

Remove all cases where Lot_Frontage is greater than 130 or smaller than 20.

Drop the unused levels from the data set . (2 points)

```
summary(Ames$Fireplaces) # shows there are values from 0 to 4.

Ames3ii <-
  Ames3 %>%
  filter(Fireplaces != 3, Fireplaces != 4) %>% # remove 3 and 4.
  filter(Lot_Frontage >= 20 & Lot_Frontage <= 130) %>%
  droplevels() # drops all unused levels from the data set.

summary(Ames3ii$Fireplaces) # shows 3 and 4 have been removed.
max(Ames3ii$Lot_Frontage) # checks Lot frontage max value
min(Ames3ii$Lot_Frontage) # checks Lot frontage min value
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  1.0000  0.5993  1.0000  4.0000 
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0000  0.5487  1.0000  2.0000 
[1] 130
[1] 21
```

The first summary shows that there are initially fireplace options from of 0,1,2,3,4. The second summary shows that now the only options are 0,1,2. The max and min codes for lot frontage confirm that values are only within requested range.

3aiii)

Save “Fireplaces” as factor in “Ames3” dataset (1 point)

```
Ames3iii <-
  Ames3ii %>%
  mutate_at(vars(Fireplaces),
            list(factor))

summary(Ames3iii$Fireplaces) # 3 category results shows that Fireplace is
                             # considered a factor, not a numerical data point.
```

```
   0    1    2 
1236 1011  153 
```

The summary shows that Fireplaces is now viewed as a factor, as the result is 3 categories of Fireplaces.

3aiv)

Construct a logistic regression model glmod for excellent_heating as a function of Lot_Frontage and Fireplaces for the dataset “Ames3”. (2 points)

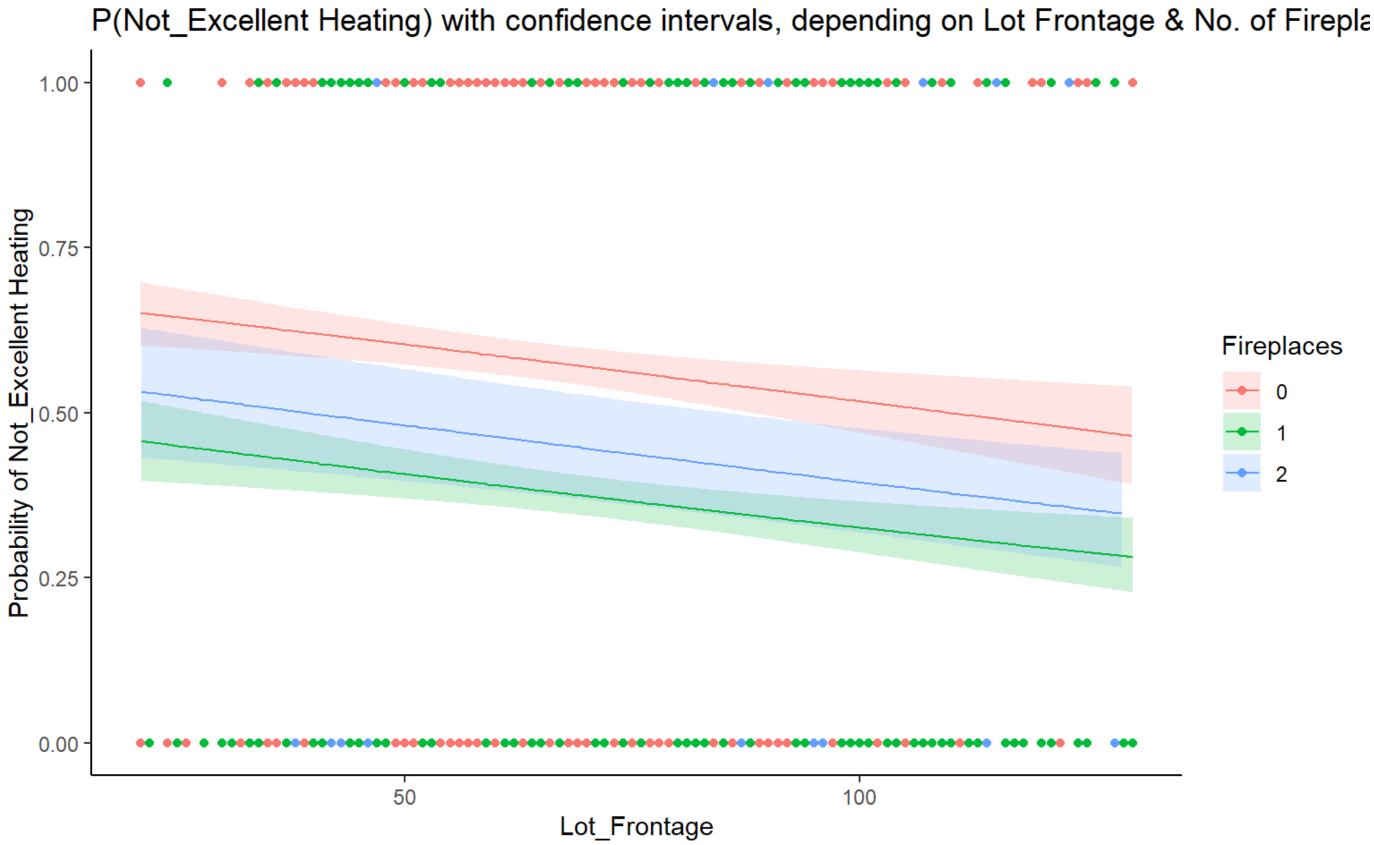
```
glmod <- glm(excellent_heating~Lot_Frontage + Fireplaces, family = "binomial",
             data = Ames3iii)
glmod
```

```
Call:  glm(formula = excellent_heating ~ Lot_Frontage + Fireplaces, 
    family = "binomial", data = Ames3iii)

Coefficients:
 (Intercept)  Lot_Frontage   Fireplaces1   Fireplaces2  
    0.769387     -0.007018     -0.796183     -0.494887  

Degrees of Freedom: 2399 Total (i.e. Null);  2396 Residual
Null Deviance:      3324 
Residual Deviance: 3213     AIC: 3221
```

3b)

Construct confidence bands for the variable excellent_heating as a function of Lot_Frontage for each number of Fireplaces (hint: create a new data frame for each number of Fireplaces). Colour these with different transparent colours for each number of Fireplaces and plot them together on the same axes. Put the actual data on the plot, coloured to match the bands, and jittered in position to make it possible to see all points. Ensure you have an informative main plot title, axes labels and a legend. (7 points)

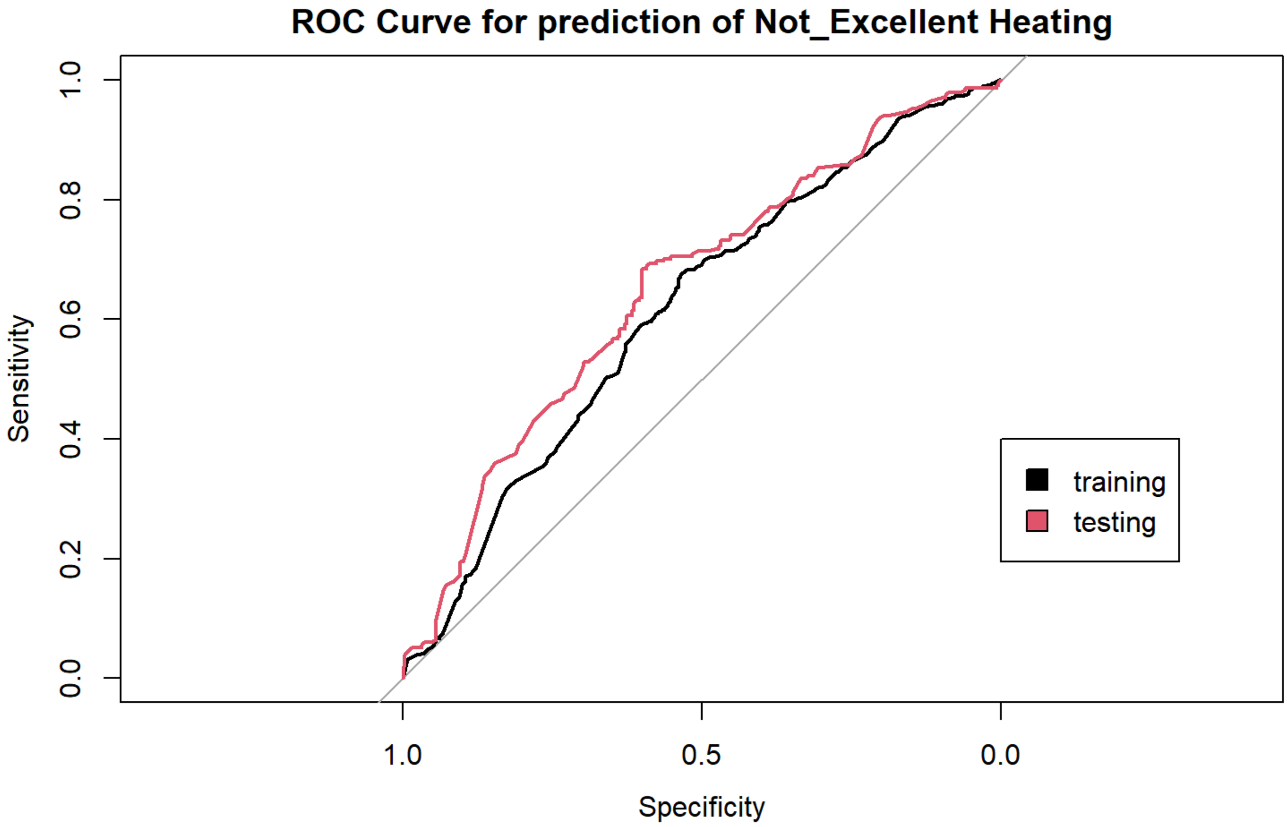
```
ilink <-family(glmod)$linkinv
newEH<-with(Ames3iii,data.frame(Fireplaces,Lot_Frontage=seq(min(Ames3iii$Lot_Frontage),
                                                max(Ames3iii$Lot_Frontage),
                                                length=2400)))

newEH<- cbind(newEH,predict(glmod,newEH,type="link",se.fit=TRUE)[1:2])

newEH<-transform(newEH,Fitted=ilink(fit),Upper=ilink(fit+(1.96*se.fit)),
                 Lower=ilink(fit-(1.96*se.fit)))

ggplot(Ames3iii, aes(x=Lot_Frontage,
                     y=as.numeric(as.factor(excellent_heating))-1,
                     colour = Fireplaces)) +
  geom_ribbon(data = newEH, aes(ymin = Lower, ymax = Upper,
                                x = Lot_Frontage, fill = Fireplaces),
              alpha = 0.2, inherit.aes = FALSE) +
  geom_line(data = newEH, aes(y = Fitted, x = Lot_Frontage)) +
  geom_point()+
  labs(y = "Probability of Not_Excellent Heating", x = "Lot_Frontage",
       title = "P(Not_Excellent Heating) with confidence intervals, depending on Lot Frontage
& No. of Fireplaces") +
  theme_classic()
```



3c)

Split the data using set.seed(120) and rebuild the model on 80% of the data. Cross validate on the remaining 20%. Plot the ROCs for both data and comment on your findings. (6 points)

```
set.seed(120)
training.samples <- c(Ames3iii$excellent_heating) %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data <- Ames3iii[training.samples, ]
test.data <- Ames3iii[-training.samples, ]

# Now create the model:
train.model <- glm(excellent_heating~Lot_Frontage + Fireplaces,
                   data = train.data, family = "binomial")

# Predict training and testing data values:
predtrain <- predict(train.model, type = "response")
predtest <- predict(train.model, newdata = test.data, type = "response")

roctrain <- roc(response = train.data$excellent_heating, predictor = predtrain,
                plot = TRUE,
                main = "ROC Curve for prediction of Not_Excellent Heating",
                auc = TRUE)
roc(response = test.data$excellent_heating, predictor = predtest, plot = TRUE,
    auc = TRUE, add = TRUE, col = 2)
legend(0, 0.4, legend = c("training", "testing"), fill = 1:2)
```



```
# Setting levels: control (base) = Excellent, case = Not_Excellent
```

```


Call:
roc.default(response = test.data$excellent_heating, predictor = predtest,     auc = TRUE, plo
t = TRUE, add = TRUE, col = 2)

Data: predtest in 248 controls (test.data$excellent_heating Excellent) < 231 cases (test.data
$excellent_heating Not_Excellent).
Area under the curve: 0.6517
```

The ROC curves above shows that the two curves for both training and testing data are very similar in shape. This suggests that the data is not overfitted to the training data, instead it correctly represents the entire data set.

4. Multinomial Regression

4a)

For the dataset “Ames”, create a model multregmod to predict BsmtFin_Type_1 from Total_Bsmt_SF and Year_Remod_Add. (3 points)

```
multiAmes <- multinom(BsmtFin_Type_1~Total_Bsmt_SF + Year_Remod_Add, data = Ames)

multiAmes
```

```
# weights:  28 (18 variable)
initial  value 5701.516737
iter  10 value 4611.614897
iter  20 value 4159.256251
iter  30 value 4153.561922
iter  40 value 4150.324235
iter  50 value 4146.549269
iter  60 value 4144.509436
iter  70 value 4144.474970
final  value 4144.474825
converged
Call:
multinom(formula = BsmtFin_Type_1 ~ Total_Bsmt_SF + Year_Remod_Add,
    data = Ames)

Coefficients:
            (Intercept) Total_Bsmt_SF Year_Remod_Add
BLQ           34.465254  6.282504e-05   -0.017706708
GLQ         -105.324418  1.030040e-03    0.052676145
LwQ           39.566891  1.243787e-05   -0.020550529
No_Basement    4.876103 -1.729079e-01    0.004007989
Rec           56.710979  1.596801e-06   -0.028929851
Unf          -29.377212 -6.987213e-04    0.015514051

Residual Deviance: 8288.95
AIC: 8324.95
```

4b)

Write out the formulas for this model in terms of P(No_Basement), P(Unf) P(Rec),P(BLQ), P(GLQ), P(LwQ), You may round coefficients to 3 dp. (4 points)

$$\begin{aligned}\text{logit}(\text{P}(\text{BLQ})) = 34.465 + 0.00006 \times \text{Total\_Bsmt\_SF} \\ -0.018 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{logit}(\text{P}(\text{GLQ})) = -105.324 + 0.001 \times \text{Total\_Bsmt\_SF} \\ +0.053 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{logit}(\text{P}(\text{LwQ})) = 39.567 + 0.00001 \times \text{Total\_Bsmt\_SF} \\ -0.021 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{logit}(\text{P}(\text{No\_Basement})) = 4.876 - 0.173 \times \text{Total\_Bsmt\_SF} \\ +0.004 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{logit}(\text{P}(\text{Rec})) = 56.711 + 0.000002 \times \text{Total\_Bsmt\_SF} \\ -0.029 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{logit}(\text{P}(\text{Unf})) = -29.377 - 0.0007 \times \text{Total\_Bsmt\_SF} \\ +0.016 \times \text{Year\_Remod\_Add}\end{aligned}$$

$$\begin{aligned}\text{P}(\text{ALQ}) = 1 - P(BLQ) - P(GLQ) - P(LwQ) \\ -P(Rec) - P(Unf) - P(\text{No\_Basement})\end{aligned}$$

4c)

Evaluate the performance of this model using a confusion matrix and by calculating the sum of sensitivities for the model. Comment on your findings. (4 points)

```
multitable <- table(Ames$BsmtFin_Type_1, predict(multiAmes, type = "class"))

names(dimnames(multitable)) <- list("Actual", "Predicted")

multitable
```

```
             Predicted
Actual        ALQ BLQ GLQ LwQ No_Basement Rec Unf
  ALQ           1   0 117   0           0  18 293
  BLQ           0   0  50   0           0  30 189
  GLQ           1   0 579   0           0   2 277
  LwQ           1   0  38   0           0  30  85
  No_Basement   0   0   0   0          80   0   0
  Rec           3   0  31   0           0  46 208
  Unf           6   0 291   0           0  76 478
```

```
SSens <-
multitable[1,1]/sum(Ames$BsmtFin_Type_1=="ALQ") +
multitable[2,2]/sum(Ames$BsmtFin_Type_1=="BLQ") +
multitable[3,3]/sum(Ames$BsmtFin_Type_1=="GLQ") +
multitable[4,4]/sum(Ames$BsmtFin_Type_1=="LwQ") +
multitable[5,5]/sum(Ames$BsmtFin_Type_1=="No_Basement") +
multitable[6,6]/sum(Ames$BsmtFin_Type_1=="Rec") +
multitable[7,7]/sum(Ames$BsmtFin_Type_1=="Unf")

SSens
```

```
[1] 2.397785
```

For perfect classification, the sensitivity of each category should be equal to 1. This means that the Sum of sensitivities for an optimum model with 7 categories (shown above) should be equal to 7. Therefore, the model above has a very low sensitivity (2.398) and is not good at representing the data set. This is because a lot of the categories have very few data points in comparison to other categories which have majority of the data points. An example of this is that the sensitivity of BLQ and LwQ is 0, whereas No_Basement has a sensitivity of 1 showing that it has been perfectly represented in the model.

The model could be improved to better represent all categories by making use of weighted probabilities.

5. Poisson/quasipoisson Regression

5a)

For the “footballer_data” dataset, create a model appearances_mod to predict the total number of overall appearances a player had based on position and age. (2 points)

```
appearances_mod <-
  glm(appearances_overall~position + age, data = footballer_data3, family = "poisson")

summary(appearances_mod)
```

```

Call:
glm(formula = appearances_overall ~ position + age, family = "poisson", 
    data = footballer_data3)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-7.5377  -3.5215   0.0351   2.1892   6.1853  

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)         1.575316   0.074884  21.037  < 2e-16 ***
positionForward     0.110606   0.027448   4.030 5.59e-05 ***
positionGoalkeeper -0.364605   0.040780  -8.941  < 2e-16 ***
positionMidfielder  0.118259   0.023309   5.074 3.90e-07 ***
age                 0.043704   0.002392  18.275  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 6539.7  on 564  degrees of freedom
Residual deviance: 6114.4  on 560  degrees of freedom
AIC: 8417.1

Number of Fisher Scoring iterations: 5
```

The summary function above provides a list of coefficients that are used to make an equation that calculates overall appearances as a function of position and age.

$$\begin{aligned}\log(\mathrm{E}(\text{Overall Appearances})) = 1.575 + 0.111 \times isForward \\ -0.365 \times isGoalkeeper + 0.118 \times isMidfielder \\ +0.044 \times Age\end{aligned}$$

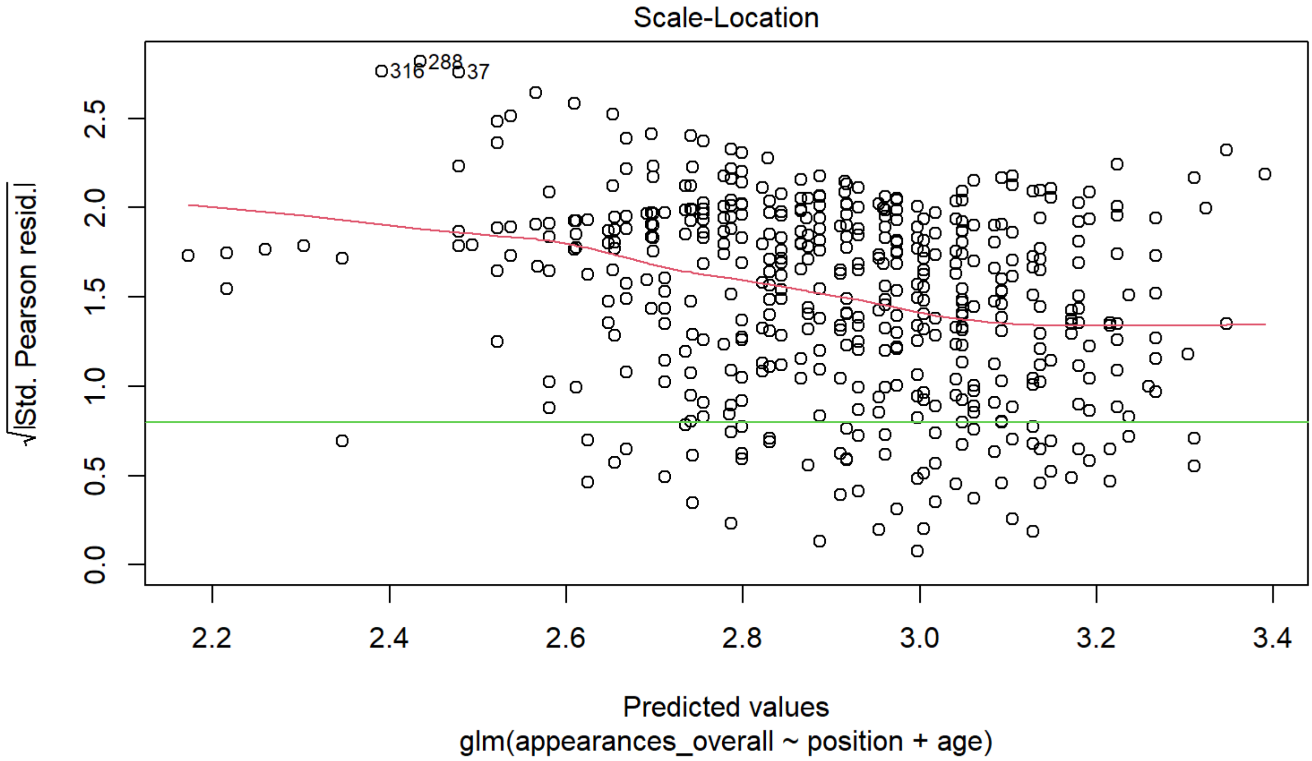
Alternatively, using the predict() function on appearances_mod, while specifying a particular age and position, will also provide a value for the response variable (Overall Appearances). (Note: This method follows a similar structure to that used in Q2k)

5b)

Check the assumption of the model using a diagnostic plot and comment on your findings. (3 points)

Using a poisson model assumes that mean = variance, hence the dispersion parameter is assumed to be 1. This can be tested using a plot of ‘Absolute value of residuals’ versus ‘Predicted Means’. which should look flat and hover around 0.8 (green line).

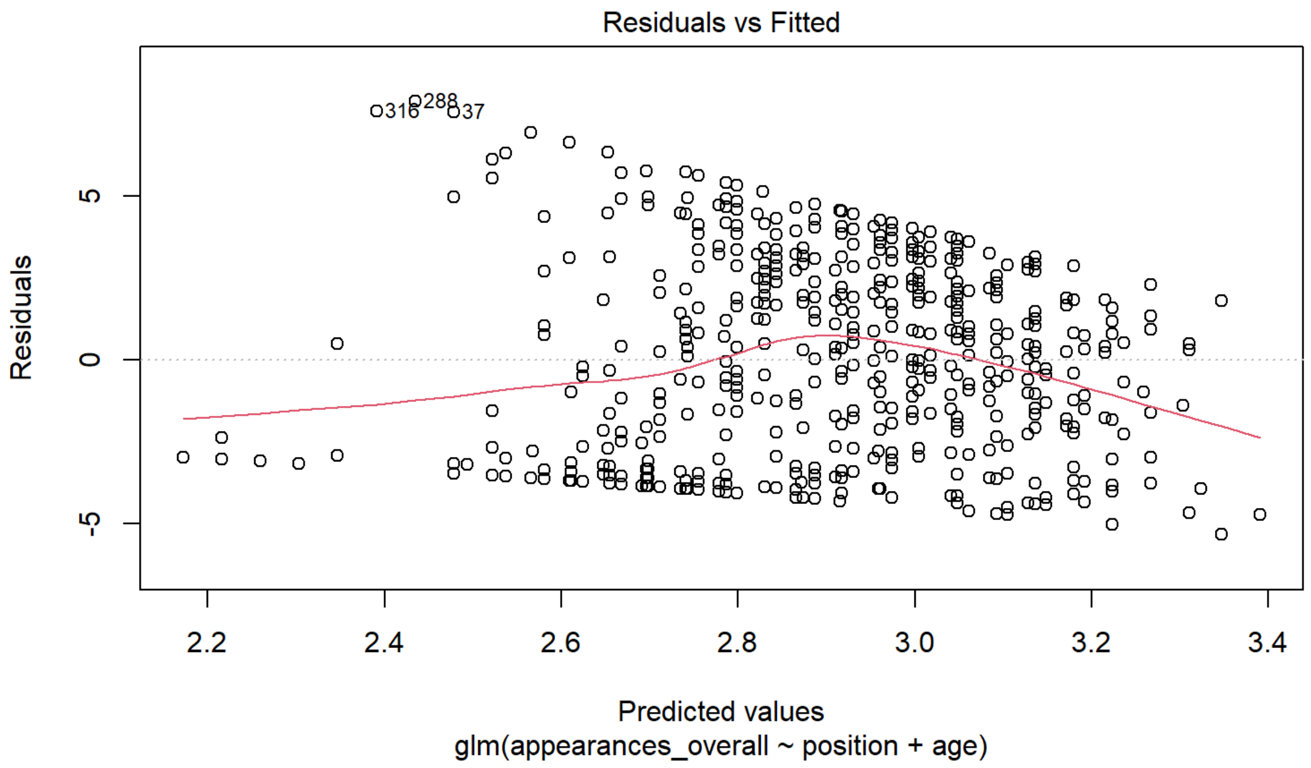
```
plot(appearances_mod,which=3)
abline(h=0.8,col=3)
```



The figure above shows that the red line (data set result) is not perfectly flat and does not hover around the green line (0.8), hence the model is not the most reasonable to use for this question. The model shows the data is heavily overdispersed, which could also suggest that we have not accounted for all of the important predictors in our model.

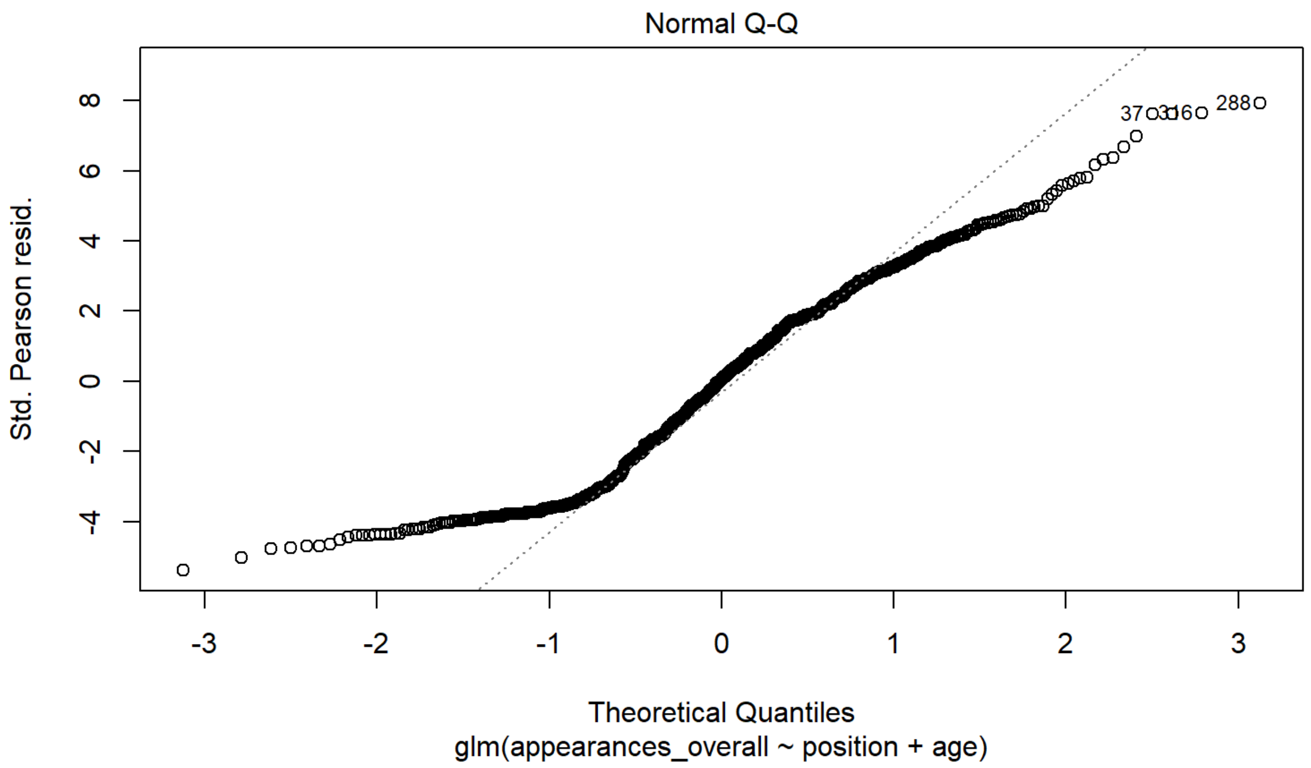
Another assumption of the Poisson model is linearity. The figure below shows the residuals vs fitted data, to see if the model data (red line) fits in with the assumption of linearity. It is evident that this is not the case as the red line slowly increases from approximately -2 until 1 (estimate), after which it decreases linearly to -3. This completely goes against the assumption of linearity as the plot is not flat at all.

```
plot(appearances_mod, which = 1)
```



The next assumption made for a Poisson Model is the assumption of Distribution. For deviance residuals, we investigate a qqplot to see if the data increases linearly. The figure below shows that the model is not perfect as the values deviate from the qqplot line towards the start and end of the x-axis (Theoretical Quantities). Between x-values of -1 to 1 however, the data satisfies the qqplot, suggesting that the Poisson Model is correct in satisfying the distribution assumption.

```
plot(appearances_mod, which = 2)
```



Thirdly, a Poisson Model assumes Independence. This investigates residuals as a function of order of data points for evidence of “snaking”. However, since the dataset is not of natural order, this cannot be investigated further.

Note: While the data above has been sufficiently analysed, the assumption validity can be further investigated by carrying out a Shapiro-Wilks test for normality, and a Breush-Pagan & NCV Test for Homoscedasticity.

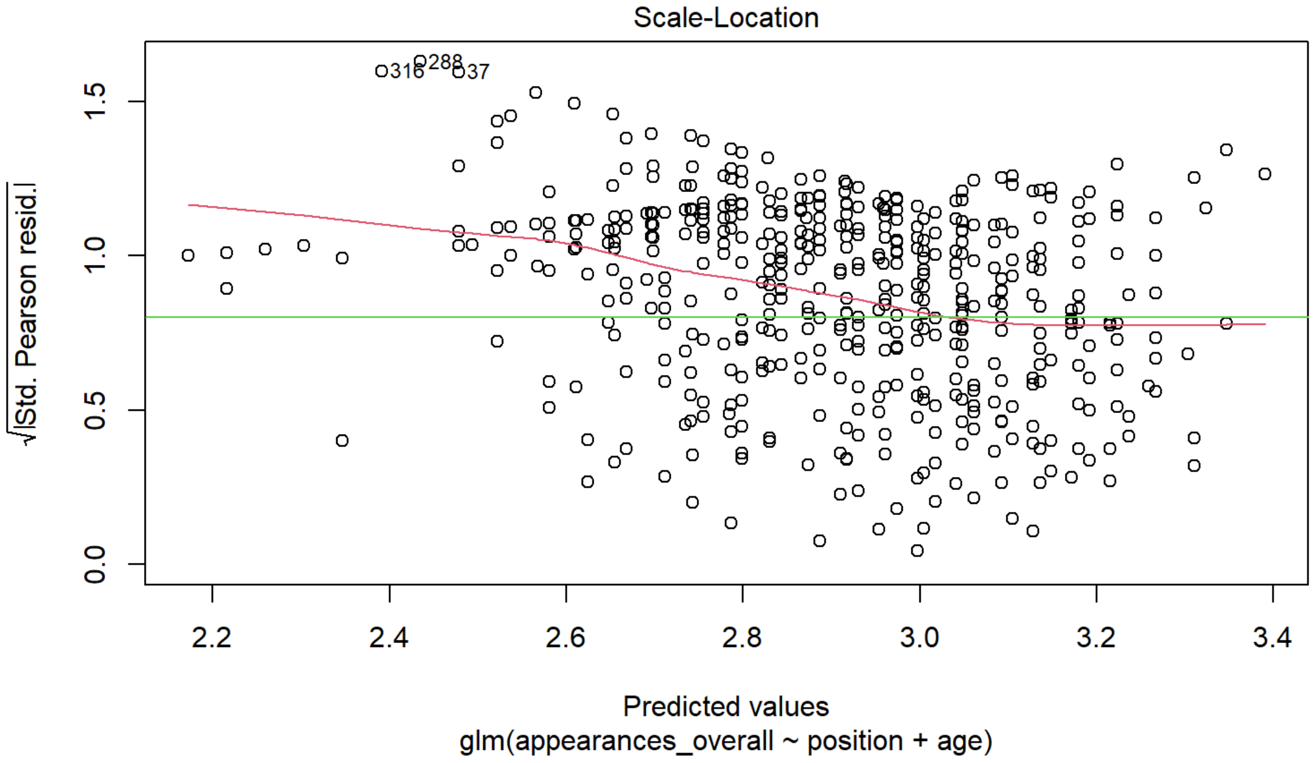
As the Poisson Model above does not satisfy most assumptions such as ‘the linearity assumption’ and ‘the variance = mean assumption’, it is worth creating a model where variance is not equal to mean, hence the dispersion parameter is not equal to 1. This can be done using a quasi-poisson model which assumes that dispersion is a linear function of mean. The summary below shows that the dispersion parameter is not equal to 1, instead it is equal to 8.95. As the dispersion parameter is greater than 1, it still suggests that there is over-dispersion in the data.

```
quasiappearances_mod <-
  glm(appearances_overall~position + age, data = footballer_data3, family = "quasipoisson")

summary(quasiappearances_mod)
```

```


Call:
glm(formula = appearances_overall ~ position + age, family = "quasipoisson",
    data = footballer_data3)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.5377  -3.5215   0.0351   2.1892   6.1853

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.575316   0.223980   7.033 5.90e-12 ***
positionForward     0.110606   0.082097   1.347  0.17844
positionGoalkeeper -0.364605   0.121975  -2.989  0.00292 **
positionMidfielder  0.118259   0.069717   1.696  0.09039 .
age                 0.043704   0.007153   6.110 1.87e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 8.946343)

    Null deviance: 6539.7  on 564  degrees of freedom
Residual deviance: 6114.4  on 560  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Using a plot of ‘Absolute value of residuals’ versus ‘Predicted Means’, the quasi-poisson model can be evaluated too, assuming the model line (red line) should be fairly linear and relatively close to the 0.8 (green line). While the figure below is not perfect, it is definitely an improvement from the Poisson Model in which the Residual value was not even close to 0.8, nor was it as linear.

```
plot(quasiappearances_mod,which=3)
abline(h=0.8,col=3)
```



5c)

What do the coefficients of the model tell us about? which position has the most appearances? How many times more appearances do forwards get on average than goalkeepers? (3 points)

The LaTex equation shows the model equation with all the coefficients for each variable.

$$\begin{aligned}\log(\mathrm{E}(\text{Overall Appearances})) = 1.575 + 0.111 \times isForward \\ -0.365 \times isGoalkeeper + 0.118 \times isMidfielder \\ +0.044 \times Age\end{aligned}$$

The model shows how likely a footballer is to play depending on their position (forward, midfielder, goalkeeper) and age, relative to the base case (likelihood of defenders playing). The coefficients in the equation above are formed from the quasi-poisson model (although coefficients are identical for the poisson model above too). The coefficients show how the response variable (‘Overall Appearances’) is affected in relation to the each of the predictor variables such as position and age. For example, a player playing in a forward position is [exp(0.111)]=1.117 times more likely to play than a player in defense (as defense is the base case). It is a log model due to it being a quasipoisson model.

The position that had the most appearances is a midfielder, as the coefficient for midfielders is 0.118, which is the highest of all the coefficients.

To find how many times more a forward plays than a defender, the above coefficients from the above equation are used in the following equation:

(Note: c is a constant being used to represent the rest of the equation, as it cancels out so is not of importance)

$$\begin{aligned}\log(\mathrm{P}(\mathrm{positionForward})) &= 0.111 + c\\ \log(\mathrm{P}(\mathrm{positionGoalkeeper})) &= -0.365 + c\end{aligned}$$

$$\begin{aligned}\log(\mathrm{P}(\mathrm{positionForward}/\mathrm{positionGoalkeeper})) &= 0.111 - -0.365(+c - c)\\ &= 0.476\\ \mathrm{P}(\mathrm{positionForward}/\mathrm{positionGoalkeeper}) &= exp(0.476)\\ &= 1.610\end{aligned}$$

Therefore, the model shows that a forward is likely to have 1.61 times more overall appearances than a goalkeeper.