Coursework MAP501 2022

Preamble

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

```
Ames<-make_ames()
```

1. Data Preparation

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

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

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

```
footballer_data_clean <- footballer_data %>%
  mutate_if(is.character, as.factor) %>%
  clean_names()
```

c. Remove the columns birthday and birthday_GMT. (2 points)

```
footballer_data_clean <- footballer_data_clean %>%
  dplyr::select(-c("birthday", "birthday_gmt"))
```

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

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

2. Linear Regression

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

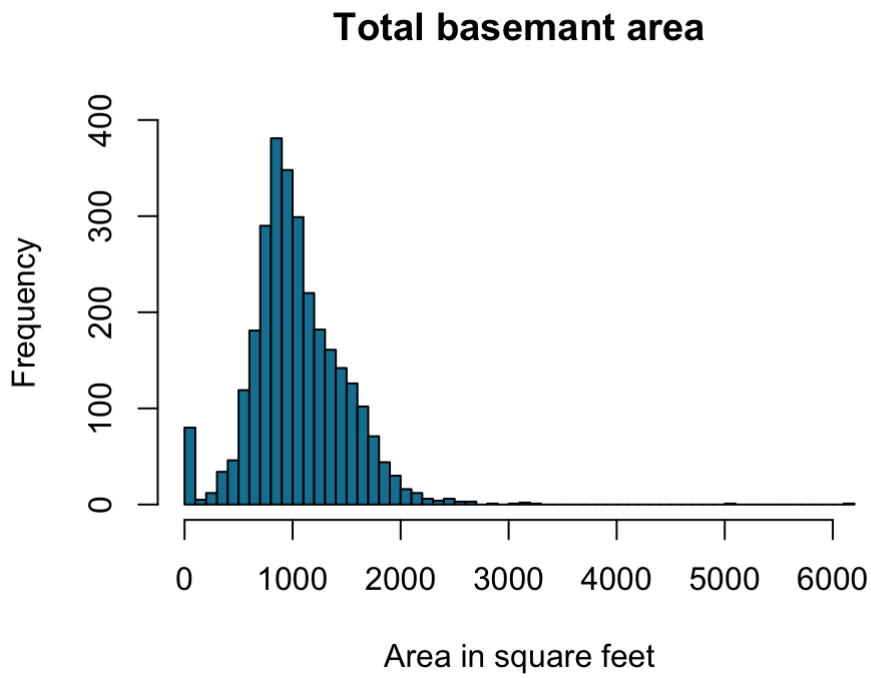
a. By adjusting x axis range and number of bars, create a useful histogram of Total_Bsmt_SF on the full dataset. Ensure that plot titles and axis labels are clear. (4 points)

From the histogram “Total basement area” can be seen, that the majority of the Basements in the dataset have an area between 0 and 2000 square feet. Further it can be seen that most of data is normally distributed.

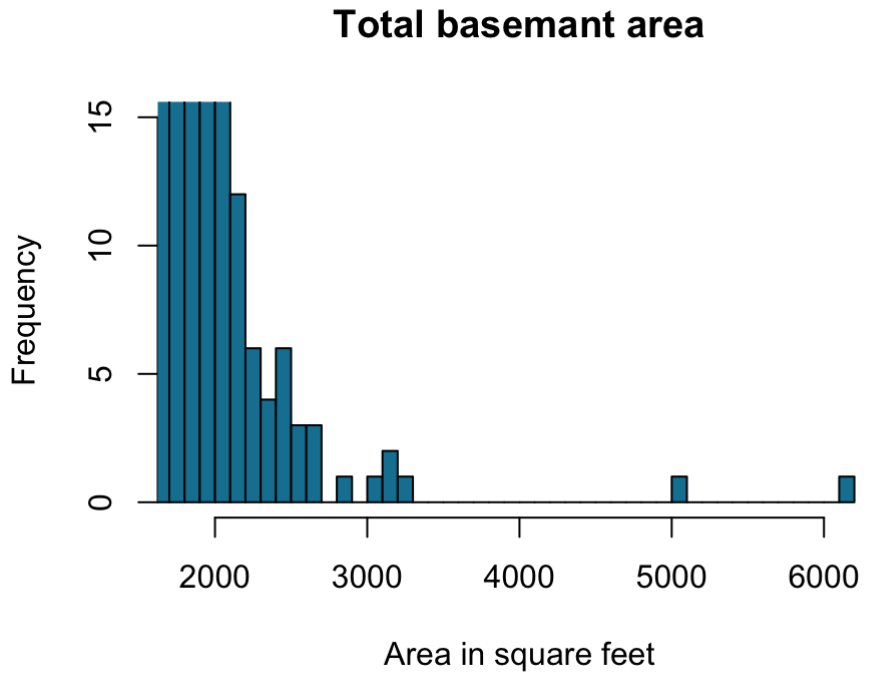
The second plot has been zoomed in on the right hand side. This has been done in order to get a better understanding of why the x range is so extended to the right on the first plot. As it can be seen, there are quite a few outliers. Those can be excluded alongside the 0 square foot outlier, as done in the third graph, in order to get a better feel of the distribution.

This has been done on the third graph, on which it can be clearly seen that the data has a normal distribution. If we have to be specific, this normal distribution is present for Basement areas ranging from above 0 and less than 2076 square feet.

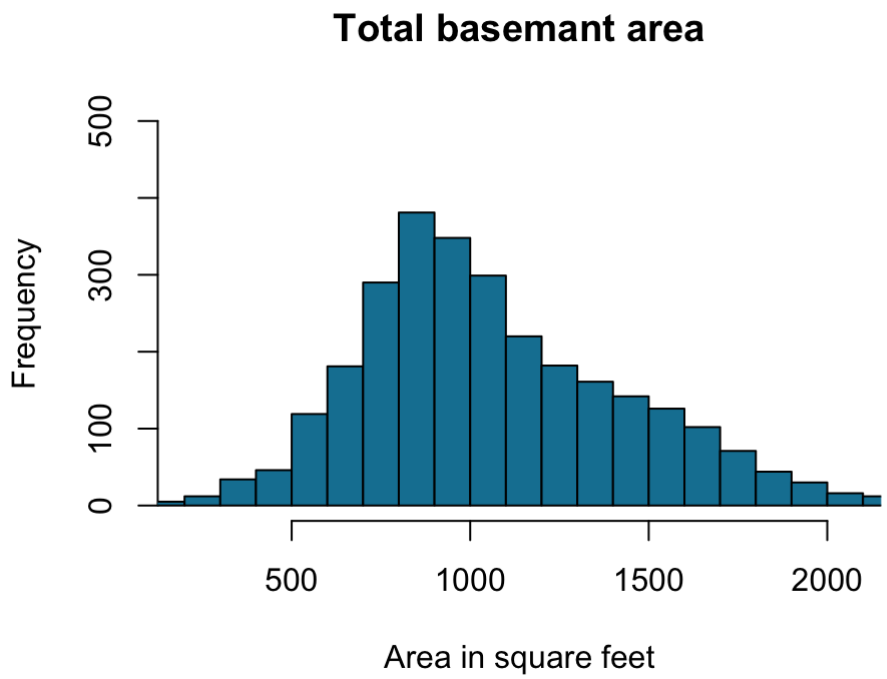
```
hist(Ames$Total_Bsmt_SF,
     main = "Total basemant area",
     xlab = "Area in square feet",
     ylim = c(0, 400),
     col = "#1380A1",
     breaks = 58)
```




```
# gain further insights from histogram, as can see some outliers
hist(Ames$Total_Bsmt_SF,
     main = "Total basemant area",
     xlab = "Area in square feet",
     xlim = c(1800, 6200),
     ylim = c(0, 15),
     col = "#1380A1",
     breaks = 58)
```



```
hist(Ames$Total_Bsmt_SF,
     main = "Total basemant area",
     xlab = "Area in square feet",
     xlim = c(200, 2076),
     ylim = c(0, 500),
     col = "#1380A1",
     breaks = 58)
```



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

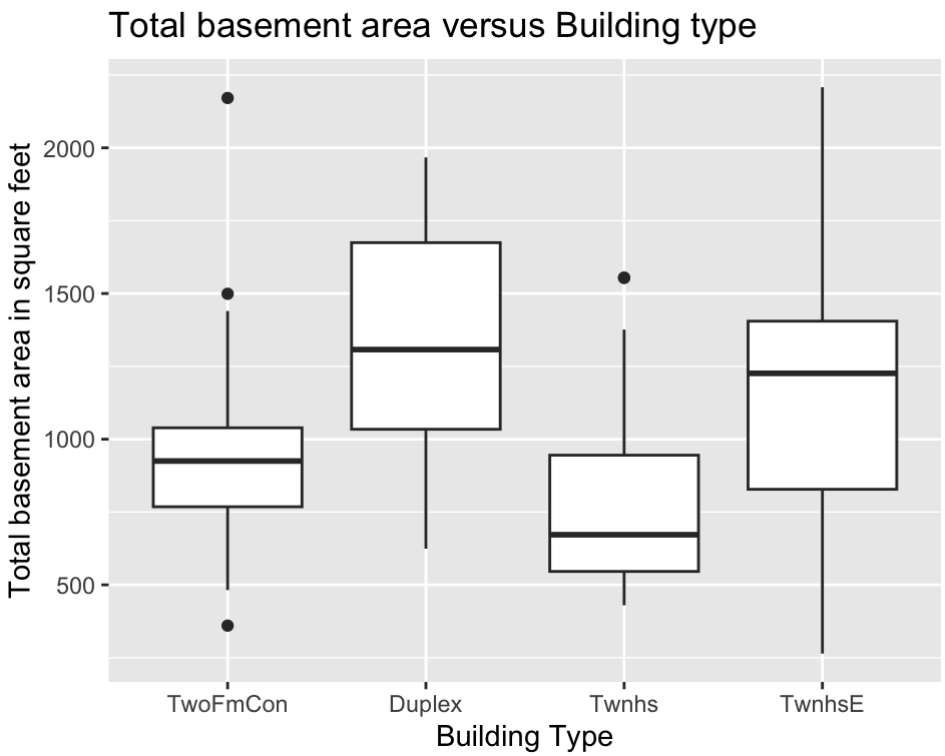
i. MS_Zoning categories of A_agr (agricultural), C_all (commercial) and I_all (industrial),
ii. BsmtFin_Type_1 category of “No_Basement”.
iii. Bldg_Type category of “OneFam” and drop the unused levels from the dataset “Ames2”. (4 points)

```
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")
Ames2 <- droplevels(Ames2)
```

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

On the figure below, it can be noticed, that Duplex and TwnhsE buildings tend to have similar median size and IQR of Basement area. This is also the case for TwoFmCon and Twnhs dwellings. Further it can be noticed that TwnhsE have the greatest range and thus variability in Basement area. While this is not the case for TwoFmCon and Twnhs which have much more conservative ranges, they include the only 4 outliers from this sample. Implying that there are not a lot of one-off exceptions in Basement areas, based on Building type.

```
Ames2 %>%
  ggplot(aes(x = Bldg_Type, y = Total_Bsmt_SF)) +
  geom_boxplot() +
  labs(title = "Total basement area versus Building type",
       x = "Building Type",
       y = "Total basement area in square feet")
```



d. 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)

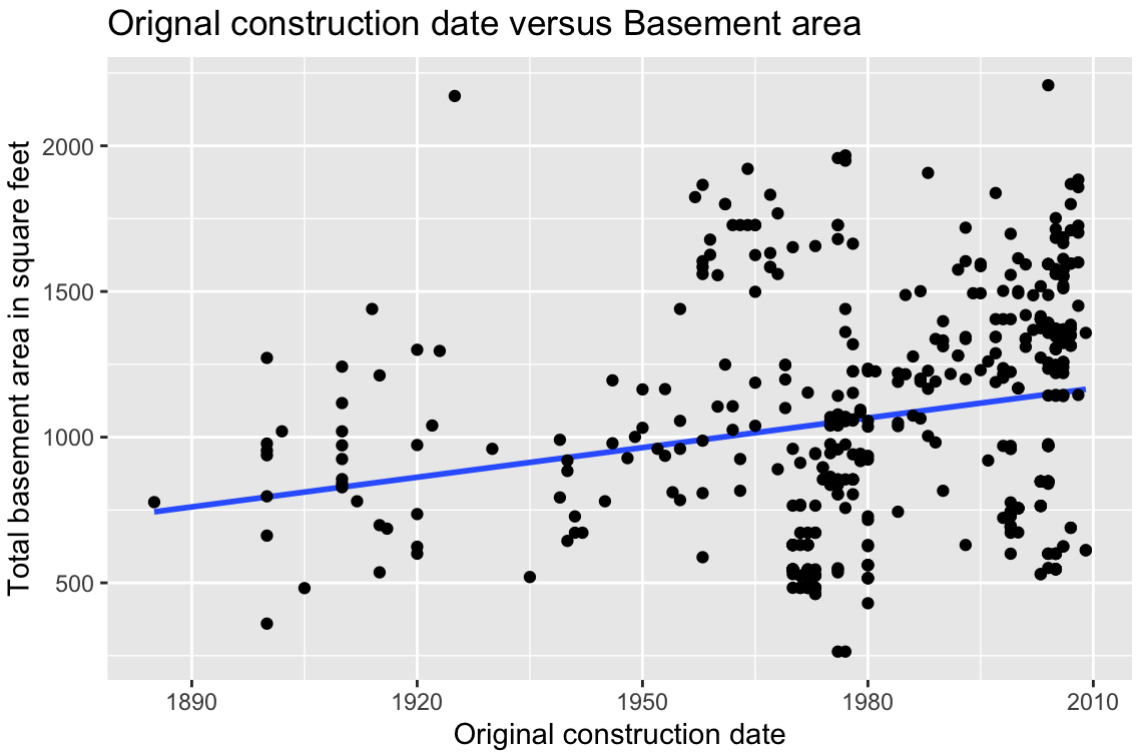
The first figure plots the original construction date against basement area on a scatter plot, regardless of dwelling type. From the provided line of best fit, it can be inferred a positive relationship between the two variables in mind. Implying that the later on a construction was build, the larger basement area it is likely to have.

Modifying this figure, in order to take into account the effect of dwelling type, gives us the second figure. From it it can be noticed that the positive relationship talked about in the paragraph above holds true for each construction category.

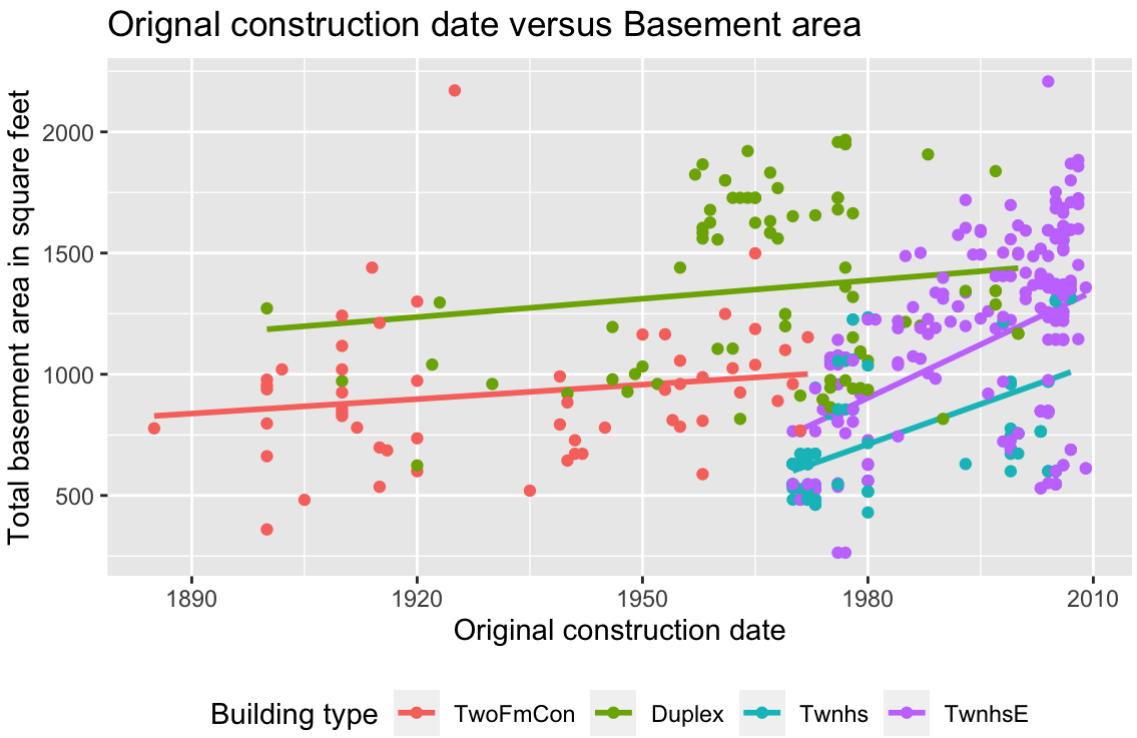
Another interesting observation is that some of the construction types have been phased out as time progresses. For instance it can be seen that around the 1970s is when the last TwoFmCon appears on the plot. Similar is the case with the Duplex category, which last appears in the early 2000s.

Further on the plots, it can be noticed that a lot of the points are clustered in dates either before 1920 or after 1950. This may be indicative of a slowdown in construction between those years. I suspect two reasons for this observation. Those are the Great Depression and both World Wars, which took place during the 30 year period in mind.

```
Ames2 %>%
  ggplot(aes(x = Year_Built, y = Total_Bsmt_SF)) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_point() +
  labs(title = "Orignal construction date versus Basement area",
       x = "Original construction date",
       y = "Total basement area in square feet")
```




```
Ames2 %>%
  ggplot(aes(x = Year_Built, y = Total_Bsmt_SF, color = Bldg_Type)) +
  geom_smooth(method = "lm", se = FALSE) +
  geom_point() +
  labs(title = "Orignal construction date versus Basement area",
       x = "Original construction date",
       y = "Total basement area in square feet",
       colour = "Building type") +
  theme(legend.position = "bottom")
```



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

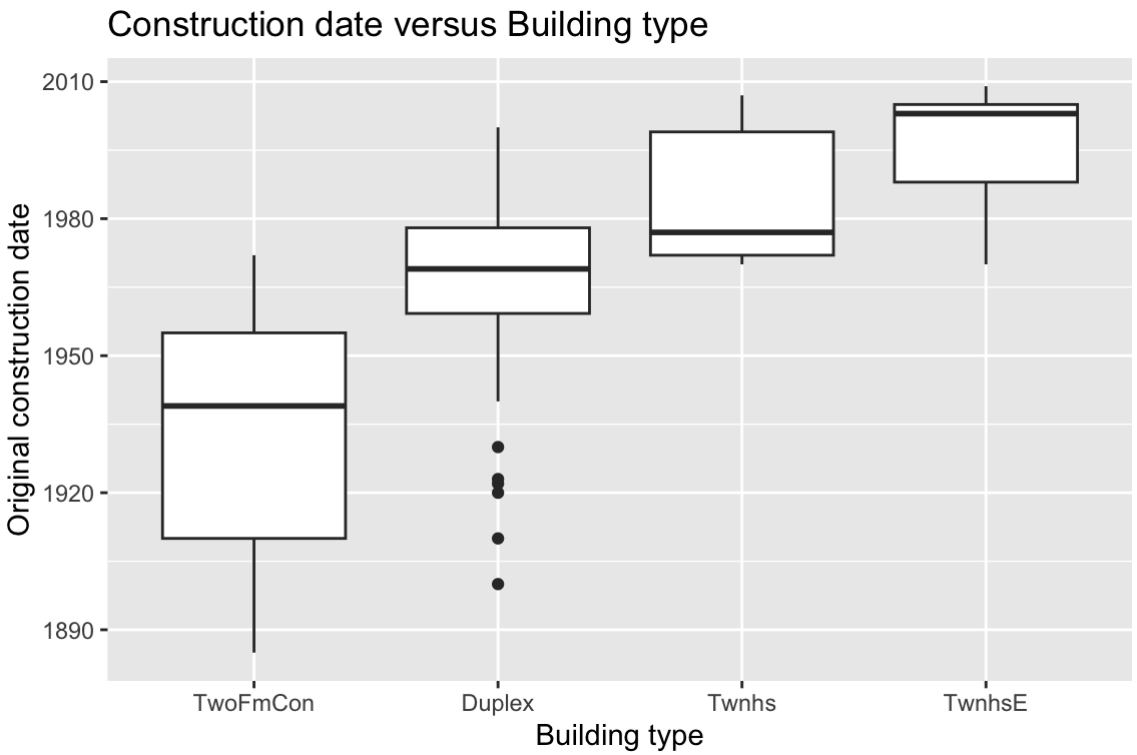
So far the plots constructed always included Total_Bsmt_SF. Initially the histogram looked over how the basement area might be distributed. The boxplot, gave us more insight on how that area might differ based on dwelling type. And the scatter plot included an additional variable, in the face of construction date that examined how basement area might differ based on it.

Therefore those plots were constructed in order to get an understanding of whether the variables Bldg_Type and Year_Built can be used to predict Basement size. From what the plots above have showed, it can be inferred that those two variables can act as predictors for basement area.

f. 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)

One of the assumptions of linear regression in that observations must be independent of each other. From the boxplot below it can be noticed that each dwelling type is associated to a certain construction date period. For instance half of the TwoFmCon have been constructed anywhere from the 1910s to 1950s. Where as Duplex dwellings from the 1960s to late 1970s. This suggests that the construction types are dependent on the original construction date. But this is to be expected when one of the variables is a time scale. Thus violating the independence assumption.

```
Ames2 %>%
  ggplot(aes(x = Bldg_Type, y = Year_Built)) +
  geom_boxplot() +
  labs(title = "Construction date versus Building type",
       x = "Building type",
       y = "Original construction date")
```

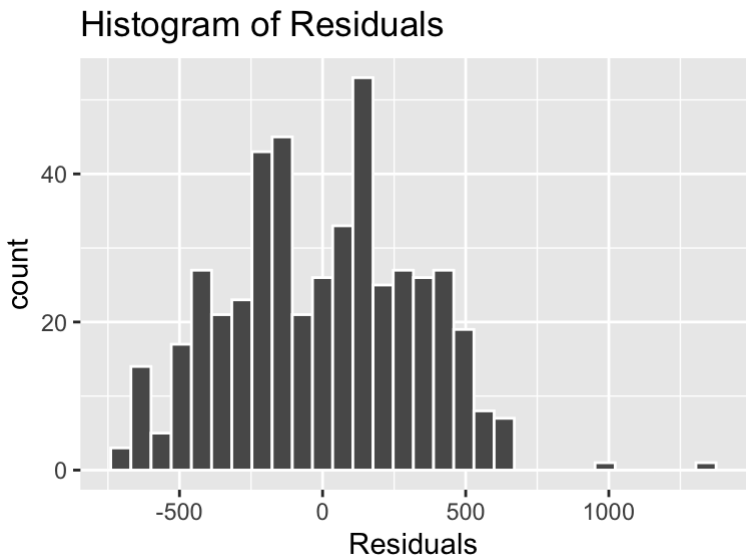


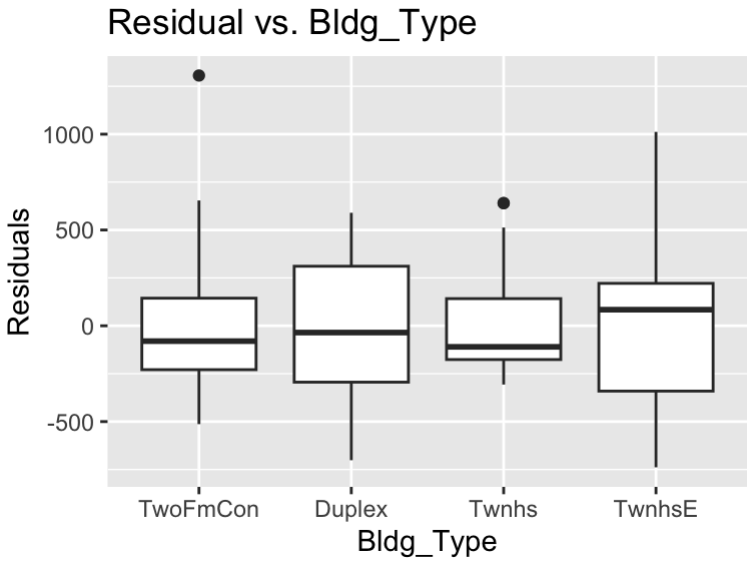
g. Use the lm command to build a linear model, linmod1, of Total_Bsmt_SF as a function of the predictors Bldg_Type and Year_Built for the “Ames2” dataset. (2 points)

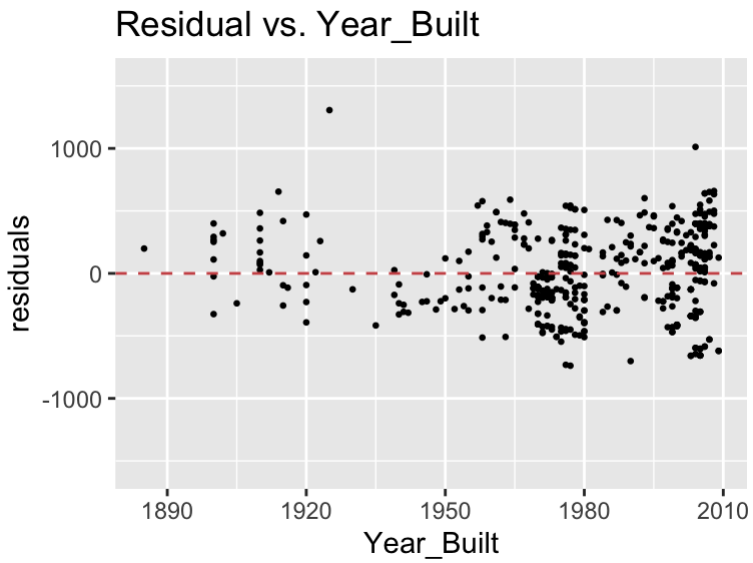
```
linmod1 <- lm(Total_Bsmt_SF ~ Bldg_Type + Year_Built, data = Ames2)
```

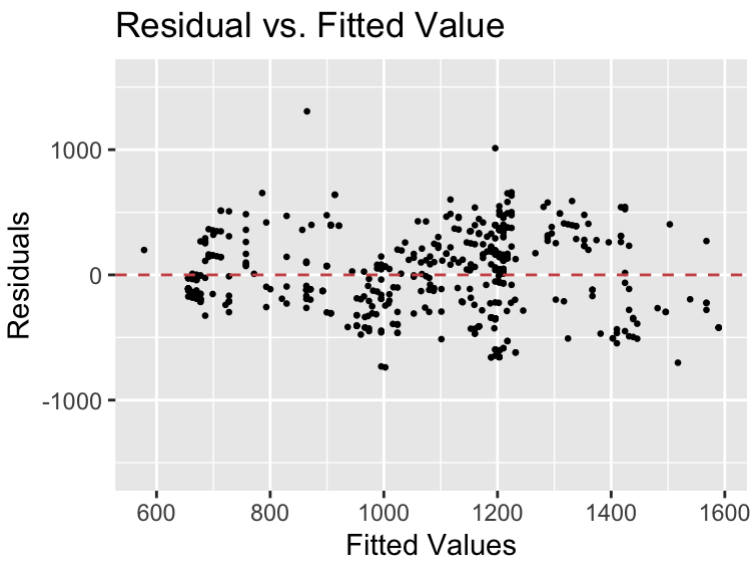
h. State and evaluate the assumptions of the model. (6 points)

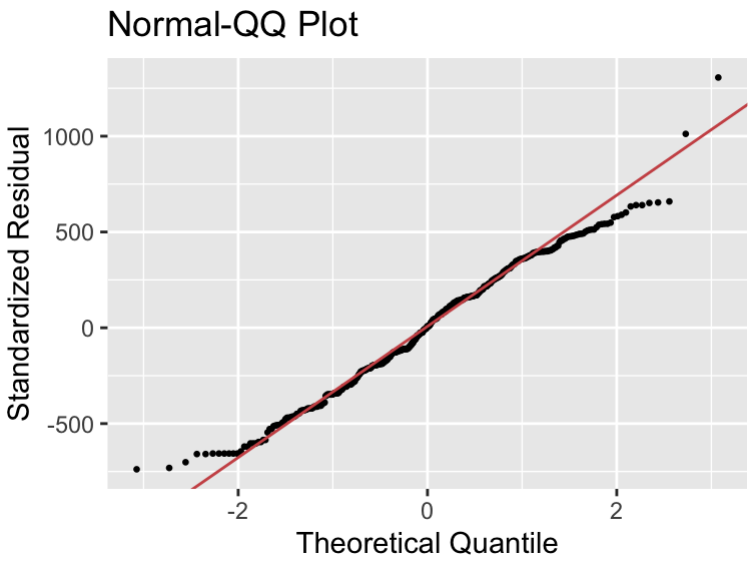
```
linmod1 %>%
  gg_diagnose(max.per.page = 1)
```

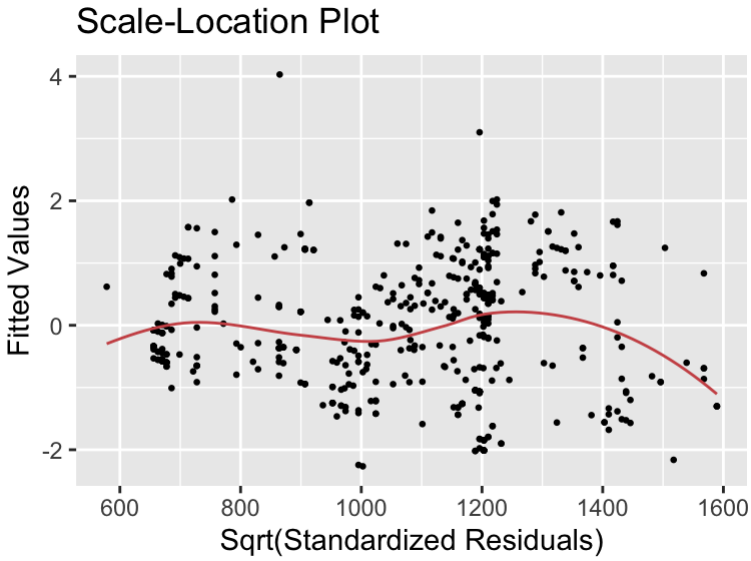


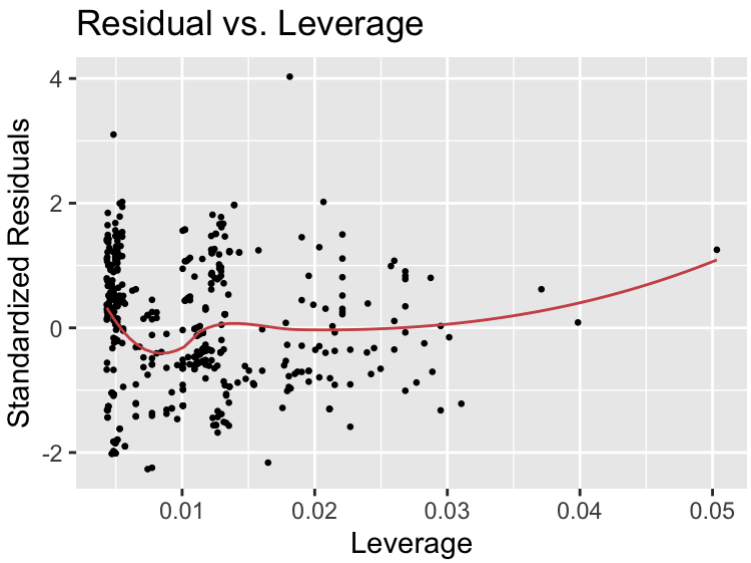

Residual vs. Bldg_Type
Residuals
1000
500
0
-500
TwoFmCon
Duplex
Twnhs
TwnhsE
Bldg_Type

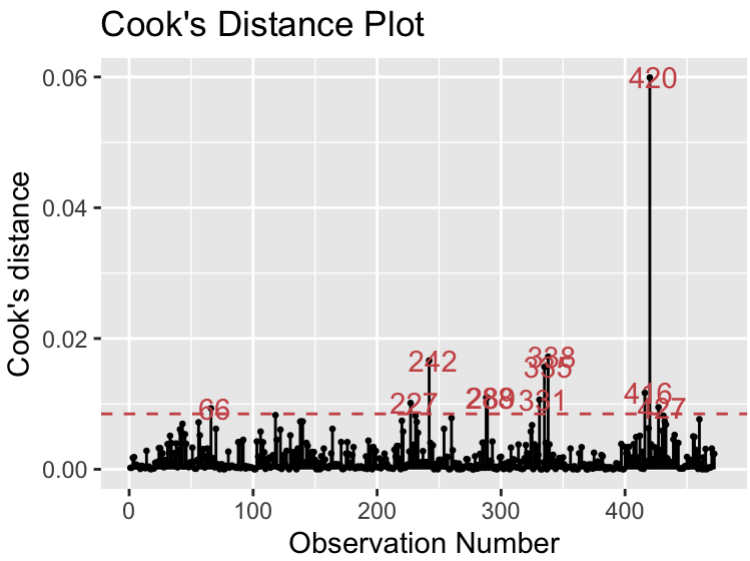

Residual vs. Year_Built
residuals
1000
0
-1000
1890
1920
1950
1980
2010
Year_Built


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)


Residual vs. Leverage
Standardized Residuals
4
2
0
-2
0.01
0.02
0.03
0.04
0.05
Leverage



- *Linearity*

This assumption of normal distribution requires there to be a linear relationship between the predictor and the response variable.

This assumption has been checked in part d) of this question, via plotting a line of best fit on the scatterplot. From it can be clearly seen that there is a positive relationship between the variables Year_Built and Total_Bsmt_SF. Further by examining the second plot from part d) it can be seen that this relationship holds true for each category of the factor Bldg_Type.

- *Homoscedasticity*

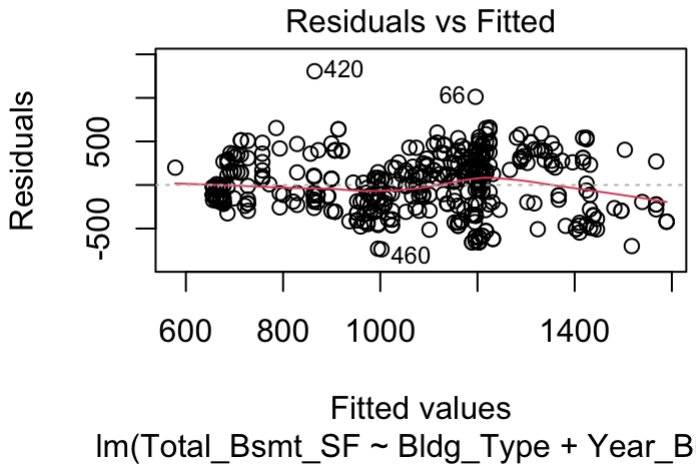
This assumption requires for our hidden parameter, to be independent of the predictors of the model. This has been examined by looking over 2 plots from gg_diagnose and 1 from plot function.

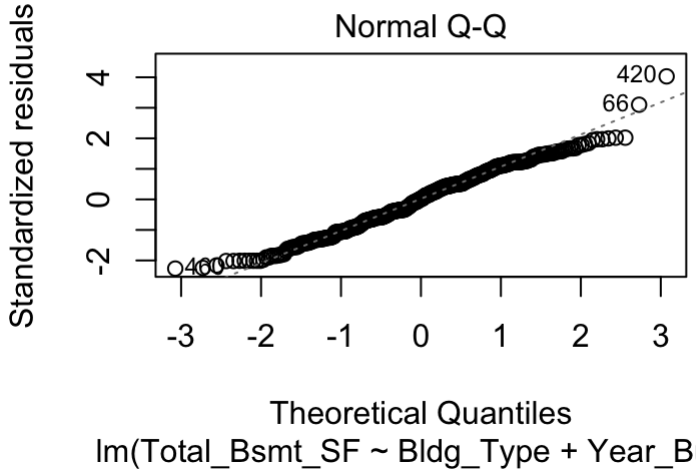
First lets examine the scatter plot against the Year_Built covariate and residuals. It can be seen that residuals tend to increase with Year_Built, seeing a larger and larger cluster of points toward later years. This would indicate the presence of heteroscedacitisity, implying that the Year_Built is in fact influencing our hidden parameter.

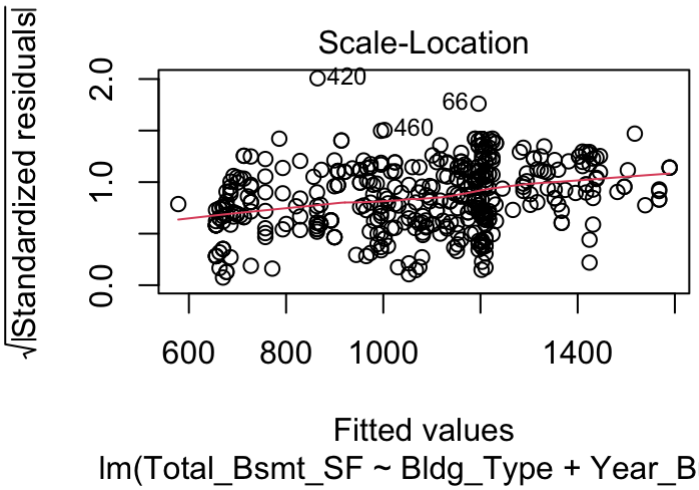
Second lets examine the categorical predictor in the face of Bldg_Type. To examine homoscedasticity, the box plot of residuals versus Bldg_Type should be looked over. From the boxplot, it can be seen that the median for all types, is somewhat around the 0 and the IQRs for each category tend to be similar in size. Looking as satisfactory output for the homoscedasticity assumption for the Bldg_Type predictor.

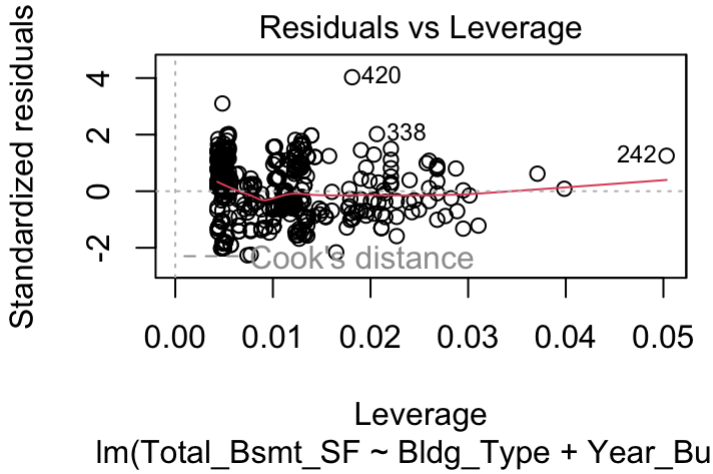
Overall it can be concluded that the model is somewhat good at meeting the homoscedacticity assumption. This can be concluded from the residuals vs fitted curve (from plot() function). As it can be seen that the relationship between residuals and fitted tends to fluctuate around the 0 vertical line, shown by the red curve. Indicating that the fitted value of predictors might be independent of the hidden parameter.

```
plot(linmod1)
```


Residuals vs Fitted
Residuals
-500
500
420
66
460
600
800
1000
1400
Fitted values
lm(Total_Bsmt_SF ~ Bldg_Type + Year_Built)


Normal Q-Q
Standardized residuals
-2
0
2
4
420
66
460
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Total_Bsmt_SF ~ Bldg_Type + Year_Built)


Scale-Location
√|Standardized residuals|
0.0
1.0
2.0
420
66
460
600
800
1000
1400
Fitted values
lm(Total_Bsmt_SF ~ Bldg_Type + Year_Built)



- *Normality*

This assumption requires that model residuals to be normally distributed.

This can be examined in 2 ways. First by looking over the histogram of residuals, which overall seems to have a typical normal/bell shaped curve. Further this can be confirmed by a qqplot, which shows us that the residuals tend to not deviate from the standard deviation of residuals line, besides by a small amount at the 2 tails. Suggesting that the model meets that assumption well.

i. 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)
```

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

Anova analysis, tests the alternative hypothesis in the face of linmod2 (fuller model), against the null in the face of linmod1 (reduced model). The p-value of 0.000081 that is produced from the Anova test, indicates that the alternative hypothesis is not rejected. Thus suggesting that linmod2, performs better that linmod1.

Adjust R Squared is suggestive of how much percent of the variation in the response is explained by the predictors. In the case for linmod1, it can be noticed that its predictors, explain 32.82% of the variation in Basement size. Whereas linnmod2's predictors explain 34.89% of the same variation. The only diffrence between linmod1 and linmod2 is the addition of Lot_Area as a predictor. Thus the addition of Lot_Area as a predictor has allowed for the additional 2.07% explanation of Basement size variability.

Those 2 tests, suggest that overall linmod2 is a better than linmod1.

```
anova(linmod1, linmod2) # h1 accept -> improvment
summary(linmod1)$adj.r.squared
summary(linmod2)$adj.r.squared # predictors better by 2% in explaining variability
```

```
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
[1] 0.3282376
[1] 0.348892
```

k. 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 confidence interval gives information about how “sure” the model is at calculating true mean values for the entire population, based on data provided from the sample. It can be seen that for Twnhs built in 1980, with a lot Area of 7300, the model predicts a true mean value for Basement size of 768,76 square feet. However the 95 % confidence interval suggests that within the whole population of dwelling with such characteristics, the model can produce a mean value of Basement size anywhere in the range 702,14 and 835,38 square feet.

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

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

The prediction interval gives information about the basement size range for a future sampled dwelling. In our case, that extra dwelling has the characteristcs of Twnhs built in 1980, with a lot Area of 7300. The prediction interval suggests that if such extra dwelling is sampled from the entire population, the model is 95% sure, that it might have a Basement area anywhere between 132.36 and 1405.16 square feet.

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

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

l. 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 | Neighborhoo
d), data = Ames2) #187.4
```

The Ames2 dataset has 27 categories under the Neighborhood factor. If this factor was to be included in the model, the whole equation would become too clustered with individual parameters and coefficients regarding each single Neighborhood. Also the effect of each Neighborhood, may not something that the model wants to

observe. Thus this predictor might be introduced into the model as a random effect. The result of it would give an idea of what is the overall “mixed” effect of Neighborhood onto the response variable.

This is explained by the hidden parameter of the random effect. In this case it is 187.4, suggesting that in general the Neighborhood variable, tends to affect basement size by 187.4 square feet.

```
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
```

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

In order to examine the significance of the random effect from the 95% confidence interval below, the value of sig01 should be looked over. It can be interpreted as the model is 95% sure that the random effect will influence basement size anywhere the range between 114.92 and 253.19 square feet. As those values do not cross the 0 point, it can be concluded that the random effect is having a significant effect onto our model.

```
confint(linmod3)
```

```
                                        2.5 %     97.5 %
.sig01                               114.916972  253.19244
.sigma                               244.221572  278.77404
(Intercept)                        -9699.022206 -595.99552
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
```

n. 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)

$$\begin{aligned}\text{Total\_Bsmt\_SF} \sim N(&-11763 + 238 \times \text{isDuplex} \\ &- 412 \times \text{isTwnhs} - 127 \times \text{isTwnhsE} \\ &+ 6.509 \times \text{YearBuild} \\ &+ 0.008 \times \text{LotArea}; 322, 1)\end{aligned}$$

$$\begin{aligned}\text{Total\_Bsmt\_SF} \sim N(&-4891 + 2.876 \times \text{YearBuild} \\ &+ 149 \times \text{isHighDensity} + 288 \times \text{isLowDensity} \\ &+ 109 \times \text{isMediumDensty} + 265 \times \text{isDuplex} \\ &- 63 \times \text{isTwnhs} + 105 \times \text{isTwnhsE} + \text{U}; 261.8)\end{aligned}$$

$$\text{Where U} \sim N(0; 187.4)$$

3. Logistic Regression

a. Do the following:

i. 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)

```
Ames3 <- Ames
Ames3$excellent_heating <- Ames3$Heating_QC %>%
  fct_collapse(not_excellent = c("Fair", "Good", "Poor", "Typical")) %>%
  fct_collapse(excellent = c("Excellent"))
```

ii. 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 dataset. (2 points)

```
Ames3 <- Ames3 %>%
  filter(Fireplaces != "3" & Fireplaces != "4") %>%
  filter(Lot_Frontage <= 130 & Lot_Frontage >= 20)

Ames3 <- droplevels(Ames3)
```

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

```
Ames3 <- Ames3 %>%
  mutate_at(vars(Fireplaces), list(factor))
```

iv. 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, data = Ames3, family = "b
inomial")
```

b. 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)

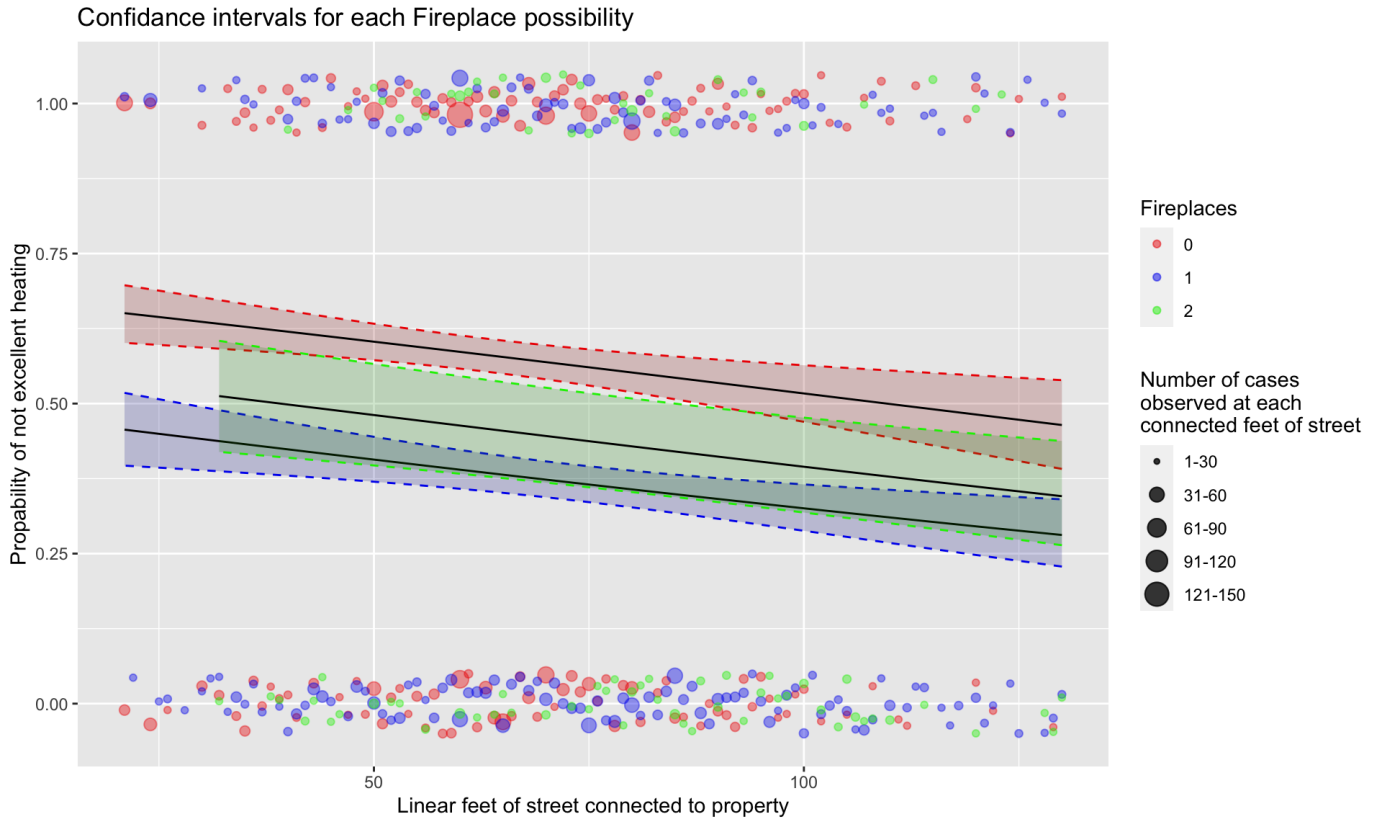
From the plot below it can be seen that all of the 3 lines for each Fireplace have a negative slope, moving from a high chance to experience not excellent heating to a lower one, as Lot_Frontage increases. Therefore regardless of how many Fireplaces a dwelling might have, as its Linear feet of street connected to the property increase, the higher chance it has to be excellently heated.

Further it can be examined how confident the model is in the chance of experiencing not excellent heating, based on the number of fireplaces that a dwelling has. That can be done by looking over the confidence bands plotted for each Fireplace category. From those it can be seen that the model is much more confident in its predictions for dwellings with 0 or 1 fireplaces, relative to those with 2. As evident from the size of the relevant confidence bands. This might be the case due to the large discrepancy of sample size that there is for the various fireplaces. In fact in the sample data there are roughly 3 times less dwellings that have 2 fireplaces, relative to those with 0 or 1.

Lastly it can be also noticed that all 3 confidence bands, tend to become thinner around values of 60 Lot_Frontage. This means that the model is much more confident about the chance of success for properties that have around 60 linear feet of street connected to the property. This might be the case, as it can be noticed the majority of dwellings are clustered in that area, regardless of fireplaces and heating outcome.

```
# set up
ilink <- family(glmod)$linkinv

Ames3_F0 <- Ames3 %>%
  filter(Fireplaces == 0)

fireplaces_0 <- with(Ames3_F0, data.frame(Lot_Frontage = seq(min(Ames3_F0$Lot_Frontag
e), max(Ames3_F0$Lot_Frontage), length = 100), Fireplaces = "0"))
fireplaces_0 <- cbind(fireplaces_0, predict(glmod, fireplaces_0, type = "link", se.fi
t = TRUE) [1:2])
fireplaces_0 <- transform(fireplaces_0, Fitted_0 = ilink(fit), Lower_0 = ilink(fit -
 (1.96 * se.fit)), Upper_0 = ilink(fit + (1.96 * se.fit)))

Ames3_F1 <- Ames3 %>%
  filter(Fireplaces == 1)

fireplaces_1 <- with(Ames3_F1, data.frame(Lot_Frontage = seq(min(Ames3_F1$Lot_Frontag
e), max(Ames3_F1$Lot_Frontage), length = 100), Fireplaces = "1"))
fireplaces_1 <- cbind(fireplaces_1, predict(glmod, fireplaces_1, type = "link", se.fi
t = TRUE) [1:2])
fireplaces_1 <- transform(fireplaces_1, Fitted_1 = ilink(fit), Lower_1 = ilink(fit -
 (1.96 * se.fit)), Upper_1 = ilink(fit + (1.96 * se.fit)))

Ames3_F2 <- Ames3 %>%
  filter(Fireplaces == 2)

fireplaces_2 <- with(Ames3_F2, data.frame(Lot_Frontage = seq(min(Ames3_F2$Lot_Frontag
e), max(Ames3_F2$Lot_Frontage), length = 100), Fireplaces = "2"))
fireplaces_2 <- cbind(fireplaces_2, predict(glmod, fireplaces_2, type = "link", se.fi
t = TRUE) [1:2])
fireplaces_2 <- transform(fireplaces_2, Fitted_2 = ilink(fit), Lower_2 = ilink(fit -
 (1.96 * se.fit)), Upper_2 = ilink(fit + (1.96 * se.fit)))

Ames3_F0_pointplot <- Ames3_F0 %>%
  group_by(Lot_Frontage, Fireplaces, excellent_heating) %>%
  summarise(
    number_of_cases = n()
  )

Ames3_F1_pointplot <- Ames3_F1 %>%
  group_by(Lot_Frontage, Fireplaces, excellent_heating) %>%
  summarise(
    number_of_cases = n()
  )

Ames3_F2_pointplot <- Ames3_F2 %>%
  group_by(Lot_Frontage, Fireplaces, excellent_heating) %>%
  summarise(
    number_of_cases = n()
  )
```

```
# plotting
ggplot(data = Ames3, aes(x = Lot_Frontage, y = as.numeric(as.factor(excellent_heatin
g)) - 1)) +
  # Fireplaces = 0
  geom_ribbon(data = fireplaces_0, aes(ymin = Lower_0, ymax = Upper_0, x = Lot_Fronta
ge), colour = "red2", fill = "red4", alpha = 0.2, lty = 2, inherit.aes = FALSE) +
  geom_line(data = fireplaces_0, aes(y = Fitted_0, x = Lot_Frontage)) +
  geom_point(data = Ames3_F0_pointplot, aes(size = number_of_cases, colour = Fireplac
es), alpha = 0.4, position = position_jitter(0, 0.05)) +
  # Fireplaces = 1
  geom_ribbon(data = fireplaces_1, aes(ymin = Lower_1, ymax = Upper_1, x = Lot_Fronta
ge), colour = "blue2", fill = "blue4", alpha = 0.2, lty = 2, inherit.aes = FALSE) +
  geom_line(data = fireplaces_1, aes(y = Fitted_1, x = Lot_Frontage)) +
  geom_point(data = Ames3_F1_pointplot, aes(size = number_of_cases, colour = Fireplac
es), alpha = 0.4, position = position_jitter(0, 0.05)) +
  #Fireplaces = 2
  geom_ribbon(data = fireplaces_2, aes(ymin = Lower_2, ymax = Upper_2, x = Lot_Fronta
ge), colour = "green2", fill = "green4", alpha = 0.2, lty = 2, inherit.aes = FALSE) +
  geom_line(data = fireplaces_2, aes(y = Fitted_2, x = Lot_Frontage)) +
  geom_point(data = Ames3_F2_pointplot, aes(size = number_of_cases, colour = Fireplac
es), alpha = 0.4, position = position_jitter(0, 0.05))+
  labs(title = "Confidance intervals for each Fireplace possibility",
       x = "Linear feet of street connected to property",
       y = "Propability of not excellent heating",
       size = "Number of cases \nobserved at each \nconnected feet of street") +
  scale_size_continuous(limits = c(0, 150), breaks = c(0, 30, 60, 90, 120, 200),
                        labels = c("1-30", "31-60", "61-90", "91-120", "121-150", ">1
50")) +
  scale_colour_manual(values = c("0" = "red2", "1" = "blue2", "2" = "green2")) +
  guides(colour = guide_legend(override.aes = list(alpha = 0.2)))
```



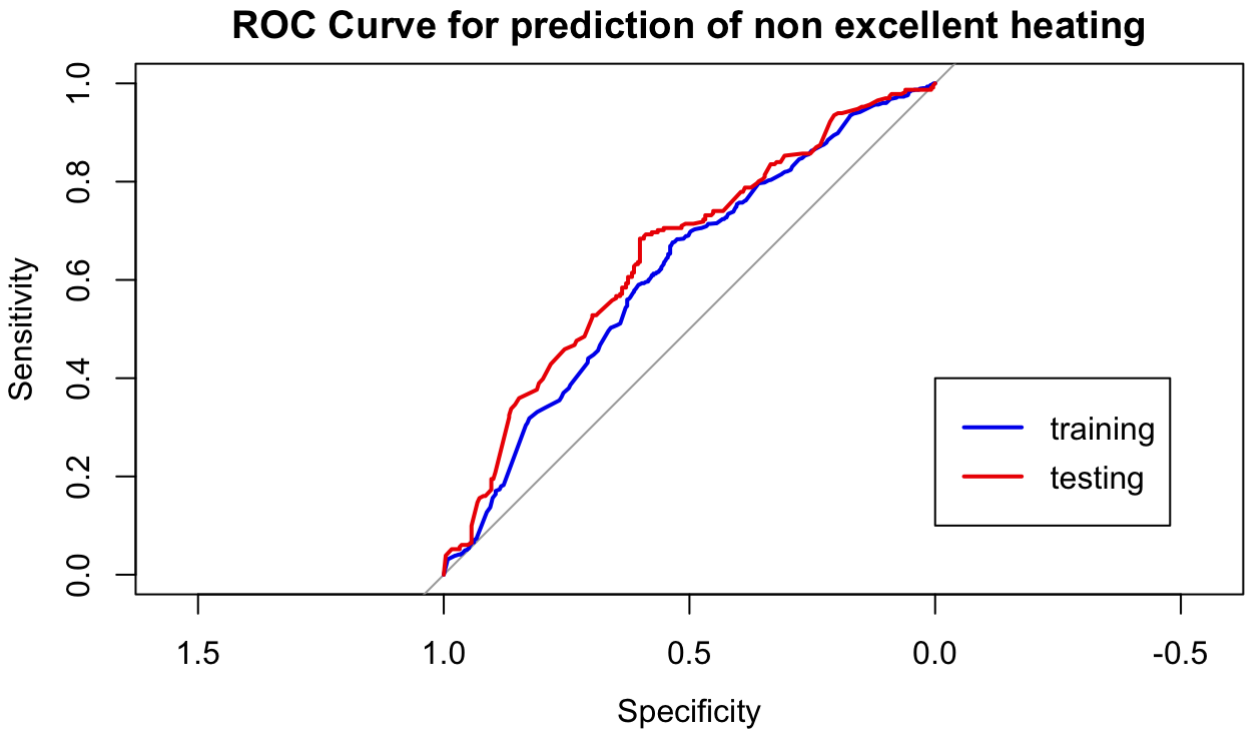
c. Split the data using set.seed(120) and rebuild the model on 80% of the data. Cross validate on the remaining 20%. Plot the ROCs for both data and comment on your findings. (6 points)

As it can be seen the ROC curves for both training and testing, seem to produce really similarly shaped curves, with very small gap across the sensitivity spectrum. It can be seen at certain sensitivity values that gap tends to get larger between the 2 curves, however even this gap can be considered small. Therefore suggesting that there is no overfitting.

Further it can be seen that the two ROC curves are not overlapping throughout, up to a point that there is essentially no gap. This suggests that the model is not underfitting either.

Those two observations suggest that the model in mind (glmod) is able to to pick up the general trend from the train data and utilize it effectively in the provided train data.

```
# split data
set.seed(120)
training.samples <- Ames3$excellent_heating %>%
  createDataPartition(p = 0.8, list = FALSE)
train.data <- Ames3[training.samples,]
test.data <- Ames3[-training.samples,]

# create training model based on train data
train.model <- glm(excellent_heating ~ Lot_Frontage + Fireplaces, data = train.data,
 family = "binomial")

# predict values on training and testing data
train_prediction <- predict(train.model, type = "response")
test_prediction <-  predict(train.model, newdata = test.data, type = "response")

# building ROC curves
roctrain <- roc(response = train.data$excellent_heating, predictor = train_predictio
n, plot = TRUE, main = "ROC Curve for prediction of non excellent heating", auc = TRU
E, col = "blue2")
roc(response = test.data$excellent_heating, predictor = test_prediction, plot = TRUE,
auc = TRUE, add = TRUE, col = "red2")
legend(0, 0.4, legend = c("training", "testing"), col = c("blue2", "red2"), lwd = 2)
```




```
Call:
roc.default(response = test.data$excellent_heating, predictor = test_prediction,
auc = TRUE, plot = TRUE, add = TRUE, col = "red2")

Data: test_prediction in 248 controls (test.data$excellent_heating excellent) < 231 c
ases (test.data$excellent_heating not_excellent).
Area under the curve: 0.6517
```

4. Multinomial Regression

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

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

```
# weights:  28 (18 variable)
initial  value 5701.516737
iter  10 value 4611.614897
iter  20 value 4159.256252
iter  30 value 4153.561922
iter  40 value 4150.324235
iter  50 value 4146.549270
iter  60 value 4144.509436
iter  70 value 4144.474970
final  value 4144.474825
converged
```

b. 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)

$$
\begin{aligned}
\text{P(BLQ)} &= \text{inverselogit}(34.465 + 0.000063 \times \text{TotalBsmtSF} - 0.0177 \times \text{YearRemodAdd}) \\
\text{P(GLQ} &= \text{inverselogit}(-105.324 + 0.0010 \times \text{TotalBsmtSF} + 0.053 \times \text{YearRemodAdd}) \\
\text{P(LwQ)} &= \text{inverselogit}(39.567 + 0.000012 \times \text{TotalBsmtSF} - 0.0206 \times \text{YearRemodAdd}) \\
\text{P(Rec)} &= \text{inverselogit}(56.711 + 0.0000016 \times \text{TotalBsmtSF} - 0.029 \times \text{YearRemodAdd}) \\
\text{P(Unf)} &= \text{inverselogit}(-29.377 - 0.00070 \times \text{TotalBsmtSF} \ + 0.016 \times \text{YearRemodAdd}) \\
\text{P(No Basement)} &= \text{inverselogit}(4.876 - 0.173 \times \text{TotalBsmtSF} + 0.004 \times \text{YearRemodAdd}) \\
\text{P(ALQ)} &= 1 - \text{P(NoBasement)} - \text{P(Unf)} - \text{P(Rec)} - \text{P(BLQ)} - \text{P(GLQ)} - \text{P(Lwq)}
\end{aligned}
$$

c. 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)

The confusion matrix computed below givess some insight on how good the model is at its predictions for each category. For instance, multregmod, did not make any correct predictions for the BLQ and LwQ categories. On the other hand, it guessed correctly all 80 cases of No_basement (perfect sensitivity). However those 3 categories in mind, represent only 17,2% of the total data points. Hinting that the rest of the predictions might be more important for assessing the model performance.

For the ALQ and Rec categories, multregmod was somewhat able to correctly guess only a small fraction of the cases. However the model was able to correctly guess more than half of the cases, in the most data dense categories which account for 58,4 of the total cases - GLQ and Unf.

Given the explanation above, summing the total sensitivities for each category of BsmtFin_Type_1, gives a results of 2,398. Ideally, the sum of sensitivities, has to be maximized, which in this case should yield a result of 7. As there are 7 categories, for which each category can obtain a perfect sensitivity of 1. Therefore this implies that there is more to be desired from the model in terms of predictive power. In order to improve the model, perhaps different weights should be attached to different categories that will allow for maximizing the sum of sensitivities.

However, a correct classification rate can also be calculated in order to examine the model. In this case it yields a value of 40,41%. Suggesting that this is the total amount of cases that were correctly classified, regardless of their category is not that bad afterall.

```
confussion_matrix <- table(Ames$BsmtFin_Type_1, predict(multregmod, type = "class"))
names(dimnames(confussion_matrix)) <- list("Actual", "Predicted")
confussion_matrix


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

sum_sens
```

```
             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
[1] 2.397785
```

```
CCR <- ((confussion_matrix[1,1] + confussion_matrix[2,2] + confussion_matrix[3,3]
        + confussion_matrix[4,4] + confussion_matrix[5,5] +
          confussion_matrix[6,6] + confussion_matrix[7,7])
        / length(Ames$BsmtFin_Type_1))

CCR
```

```
[1] 0.4040956
```

5. Poisson/quasipoisson Regression

a. 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 ~ age + position, data = footballer_data_c
lean, family = "poisson")
```

$$\text{appearances\_overall} \sim \text{Pois}(\exp(1.575 + 0.044 \times \text{age} + 0.106 \times \text{isForward} - 0.365 \times \text{isGoalkeeper} + 0.118 \times \text{isMidfielder}))$$

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

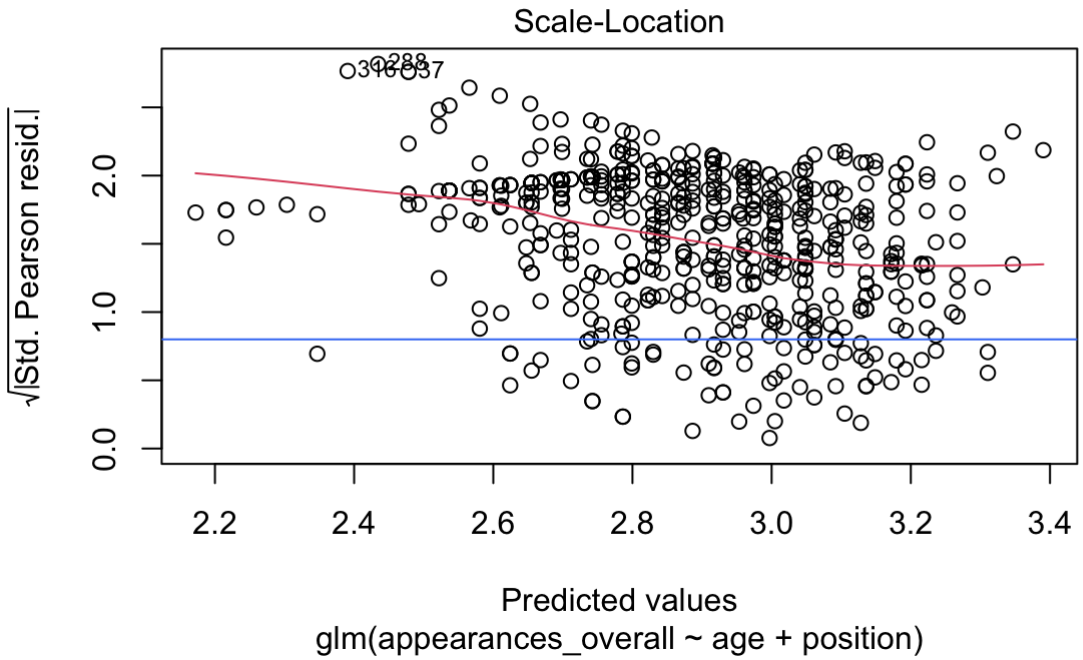
- *Dispersion*

Poisson models have an assumption that requires their variance to be equal to their mean.

To test the assumption, the scatter plot of absolute value of residuals versus predicted means is examined. Ideally the relationship between the 2 variables, should be centered and hoovering around the value 0.8, which is the blue line on the plot below. However as it can be seen by the red line, it is constantly above 0,8 even after dropping down by a bit. This suggests a case of overdispersion, which might be arising due to the model, not taking account of all relevant predictors. This may lead to throwing away our estimates for model’s coefficients.

To combat this, a distribution with more relaxed assumption about the mean and variance might be utilized, such as the quasipossion one. As it assumes dispersion as a linear function of mean.

```
plot(appearances_mod, which = 3)
abline(h = 0.8, col = "#427ef5")
```

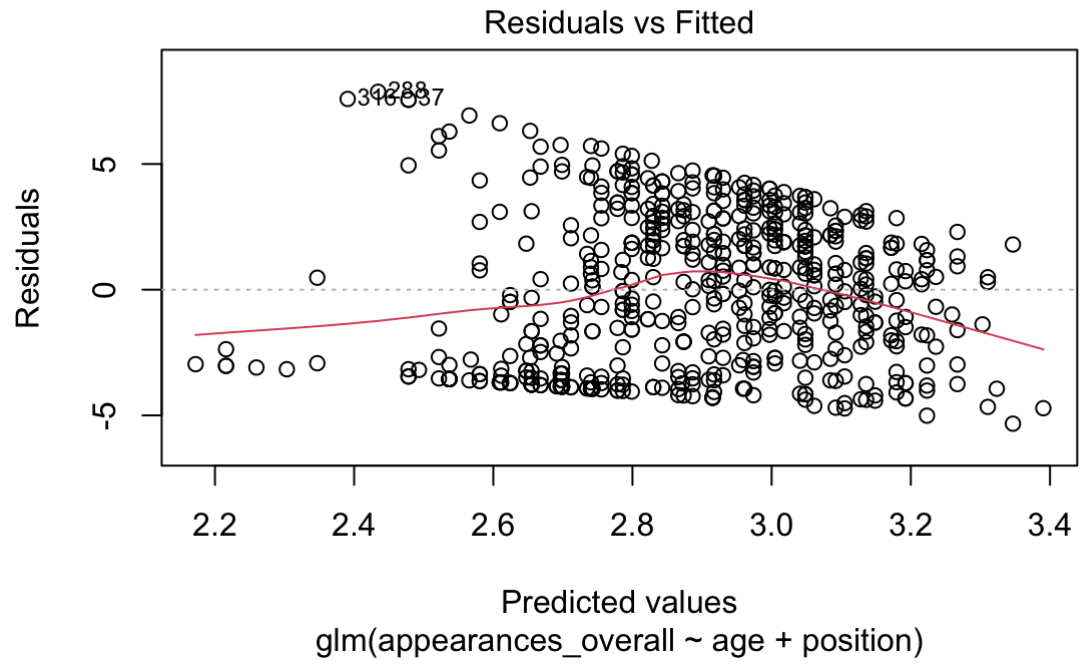


- *Linearity*

This assumptions requires there to be a linear relationship between the predictor and the response variable.

Linearity can be examined, by looking at the residuals against fitted values scatterplot. Ideally there should be a linear relationship, along the vertical 0 line, where the black dashed line is. However in this case, it can be seen that the red line tends to initially incline and then decline. Implying that linearity assumptions is more than likely to be not met.

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

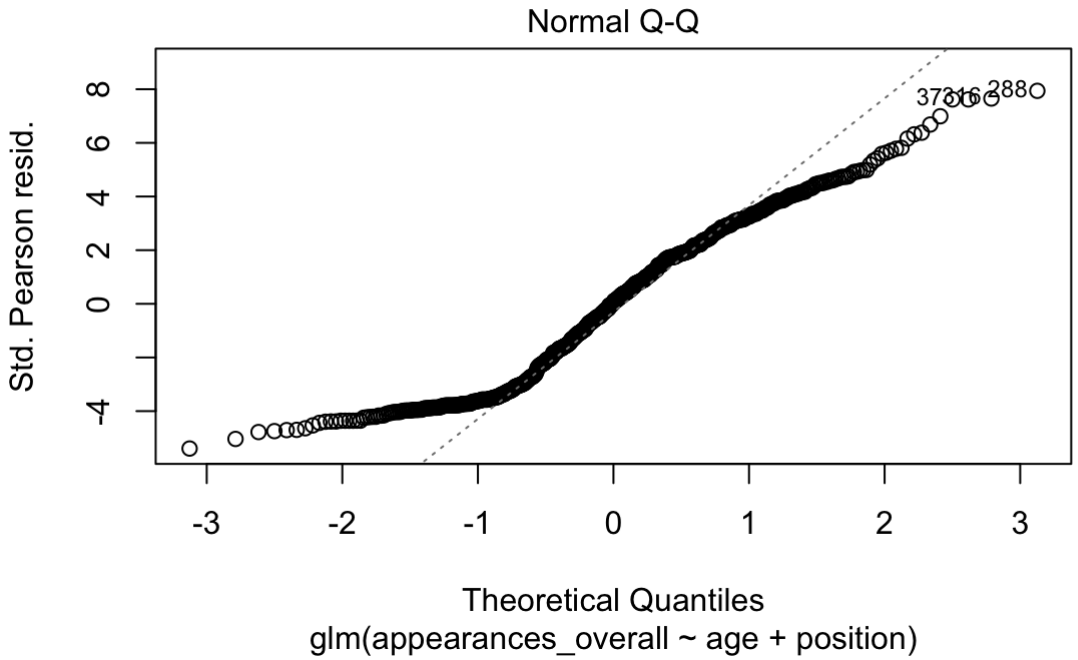


- *Distribtion*

In this assumption, the deviance residuals should be poissonaly distributed.

This can be checked via a qqplot, where we would want the residuals on the scatterplot to be following the drawn dashed line. However as it can be seen, in this case there seems to be a pretty severe deviance at the two tails. Suggesting that the model does not meet that assumption very well.

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



- *Independance*

This assumption requires residuals to be independent of each other. However as in this case, the data does not have any natural order. Implying that this assumption can not be tested.

c. 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)

In the equation appearance overall, all of the categorical parameters related to position are relative to the base level for the position Defender. As it can be spotted, the coefficient of the isMidfielder parameter is the highest. Indicating that this position is likely to have the most appearances according to the model.

Further Forwards have 1.61 times more appearances than Goalkeepers. Such estimates are gotten from the appearances_overall_altered model. It alters appearances_overall, by switching the base level for position to be Goalkeeper. Thus the coefficient for the isForward parameter is indicative of how many times more appearances Forwards have relative to Goalkeepers. However the value there is 0.475 not 1.61. This is the case, as the Poisson distribution requires bringing the value to an exponential, in order to obtain the actual value for appearances overall.

By conducting a 95% confidence interval, it can be seen that the model is sure that the value discussed above can fluctuate anywhere between 1.48 and 1.75.

However as discussed in section b) perhaps a poisson distribution is not ideal for this model, due to failing the dispersion assumption. To combat this a quasipoisson distribution may be implemented. Doing so, it again is yielded again that Forwards are 1.61 times more likely to appear on average than Goalkeepers.

By conducting a 95% confidence interval, it can be noticed that this model now has more conservative ranges. Those now fluctuate between 1.26 and 2.08.

```
footballer_data_clean$position_altered <- factor(footballer_data_clean$position, c("G
oalkeeper", "Defender", "Forward", "Midfielder"))

appearances_mod_altered <- glm(appearances_overall ~ position_altered + age, data = f
ootballer_data_clean, family = "poisson")

appearances_mod_altered_quasi <- glm(appearances_overall ~ position_altered + age, da
ta = footballer_data_clean, family = "quasipoisson")
```

$$\begin{aligned}\text{appearances\_overall\_altered} \sim \text{Pois}(\exp(&1.211 + 0.044 \times \text{age} + 0.475 \times \text{isForward} \\ &+ 0.365 \times \text{isDefender} + 0.482 \times \text{isMidfielder}))\end{aligned}$$

```
appearances_mod_altered$coefficients
appearances_mod_altered_quasi$coefficients
exp(confint(appearances_mod_altered))
exp(confint(appearances_mod_altered_quasi))

exp(0.47521046)
```

```
               (Intercept)   position_alteredDefender
                1.21071154                 0.36460478
   position_alteredForward position_alteredMidfielder
                0.47521046                 0.48286355
                       age
                0.04370446
               (Intercept)   position_alteredDefender
                1.21071154                 0.36460478
   position_alteredForward position_alteredMidfielder
                0.47521046                 0.48286355
                       age
                0.04370446
                              2.5 %   97.5 %
(Intercept)                2.828957 3.977649
position_alteredDefender   1.330174 1.560805
position_alteredForward    1.479320 1.750553
position_alteredMidfielder 1.497969 1.755865
age                        1.039791 1.049585
                              2.5 %   97.5 %
(Intercept)                2.008376 5.566260
position_alteredDefender   1.140009 1.840121
position_alteredForward    1.256136 2.079593
position_alteredMidfielder 1.285376 2.068268
age                        1.030153 1.059451
[1] 1.608353
```