Exploratory Data Analysis

Introduction to Dataset

Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients. These ingredients include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate. The actual concrete compressive strength (MPa) for a given mixture under a specific age (days) was determined from laboratory. Data is in raw form (not scaled).

Given is the variable name, variable type, the measurement unit and a brief description. The concrete compressive strength is the regression problem. The order of this listing corresponds to the order of numerals along the rows of the database.

Name	Data Type	Measurement	Description
Cement	quantitative	kg in a m3 mixture	Input Variable
Blast Furnace Slag	quantitative	kg in a m3 mixture	Input Variable
Fly Ash	quantitative	kg in a m3 mixture	Input Variable
Water	quantitative	kg in a m3 mixture	Input Variable
Superplasticizer	quantitative	kg in a m3 mixture	Input Variable
Coarse Aggregate	quantitative	kg in a m3 mixture	Input Variable
Fine Aggregate	quantitative	kg in a m3 mixture	Input Variable
Age	quantitative	Day (1~365)	Input Variable
Concrete compressive strength	quantitative	MPa	Output Variable

In [3]:
```
install.packages("readxl")
library(readxl)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [5]:
```
concrete = read_excel("/content/Concrete_Data.xls")
head(concrete)
```

A tibble: 6 × 9

Cement (component 1)(kg in a m^3 mixture)	Blast Furnace Slag (component 2)(kg in a m^3 mixture)	Fly Ash (component 3)(kg in a m^3 mixture)	Water (component 4)(kg in a m^3 mixture)	Superplasticizer (component 5)(kg in a m^3 mixture)	Coarse Aggregate (component 6)(kg in a m^3 mixture)	Fine Aggregate (component 7)(kg in a m^3 mixture)
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
540.0	0.0	0	162	2.5	1040.0	676.0
540.0	0.0	0	162	2.5	1055.0	676.0
332.5	142.5	0	228	0.0	932.0	594.0
332.5	142.5	0	228	0.0	932.0	594.0
198.6	132.4	0	192	0.0	978.4	825.5
266.0	114.0	0	228	0.0	932.0	670.0

In [6]:
```
colnames(concrete) <- c("cement", "blast_furnace", "fly_ash", "water", "plast", "coarse", "fine", "age", "strength")
```

In [7]:
```
# Checking shape of dataset

dim(concrete)
```

```
1030 · 9
```

In [8]:
```
# Checking for nulls presence

colSums(is.na(concrete))
```

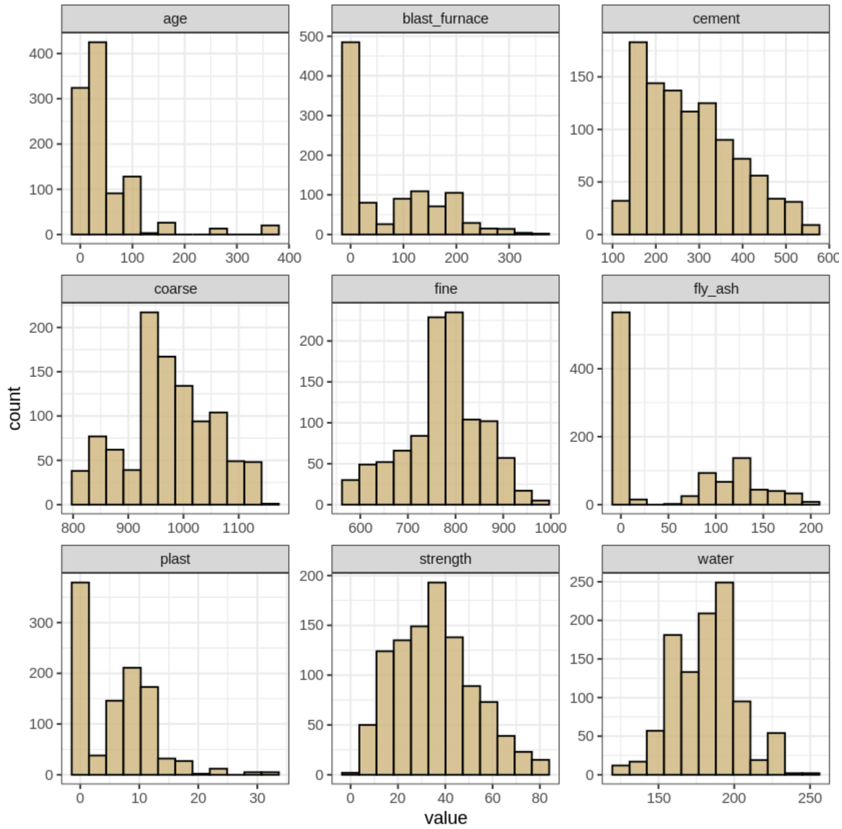
cement: 0 **blast_furnace:** 0 **fly_ash:** 0 **water:** 0 **plast:** 0 **coarse:** 0 **fine:** 0 **age:** 0 **strength:** 0

There are no missing values in our dataset. So, let us see distribution of these features.

In [9]:

```
library(purrr)
library(tidyr)
library(ggplot2)

concrete %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram(bins = 12, color="black", fill ="#CFB87C", alpha = 0.8)
+
    theme_bw()
```

```
Attaching package: ‘purrr’


The following object is masked from ‘package:base’:

    %||%
```



The histograms of each variable provide insight into the distributions of the variables. None look normal; however, there is no requirement that the predictors come from a normal distribution. The response variables, `strength` does not look all that normal, but can be seen as an approximate normal distribution. Also, we can notice that, feature variables, `age` , `blast_furnace` , `fly_ash` and `plast` seems to have few outliers. Let's do some regression analyses can be fairly robust to deviations in the normality assumption.

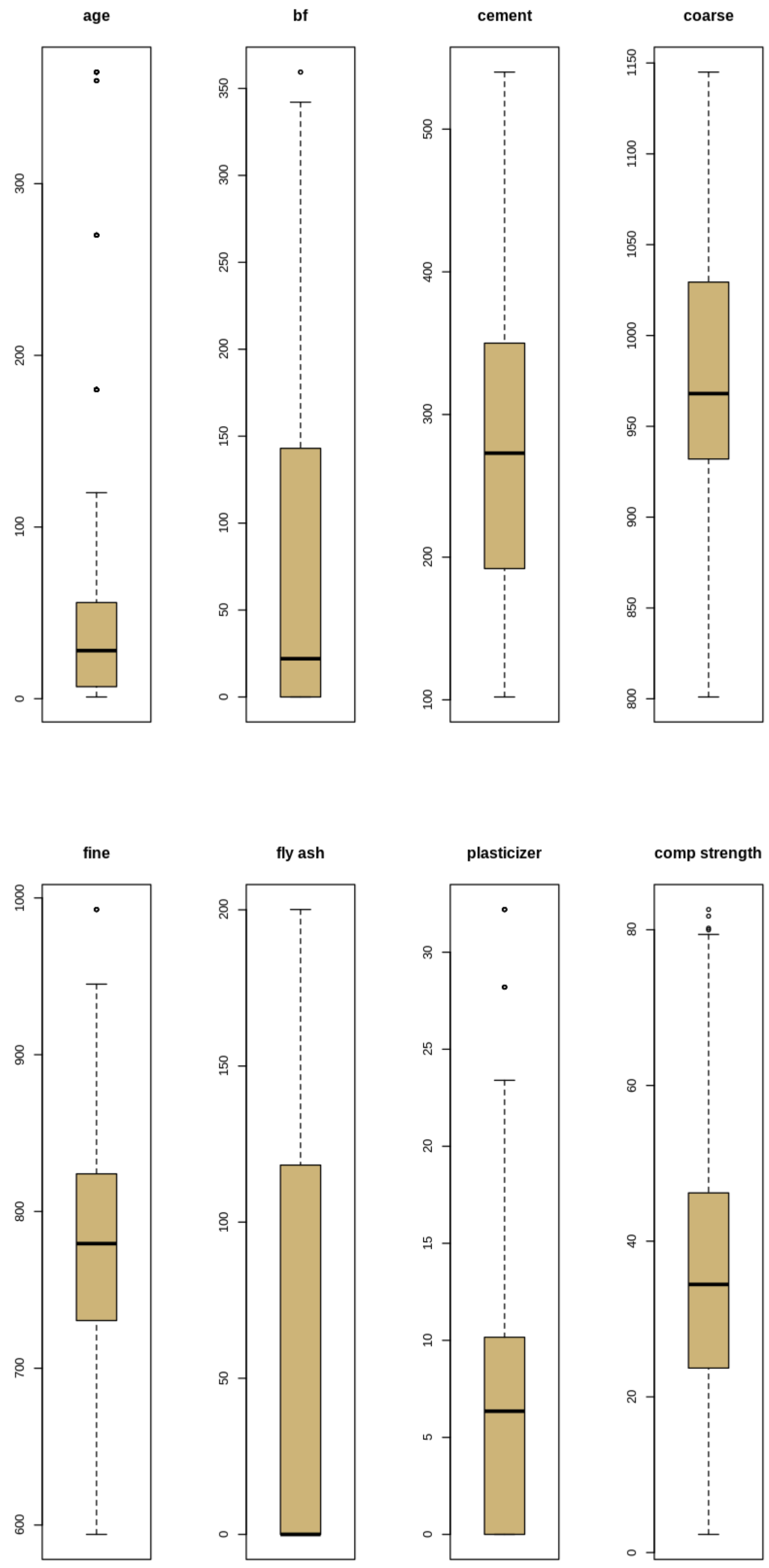
As we may have noticed from the histogram of `fly_ash` , there may be a few outliers. Let's look at some boxplots to see in further detail. R classifies potential outliers by the "IQR criterion". This criterion means that all observations above $q_{0.75} + 1.5 \times IQR$ or below $q_{0.25} - 1.5 \times IQR$ are classified as outliers, where

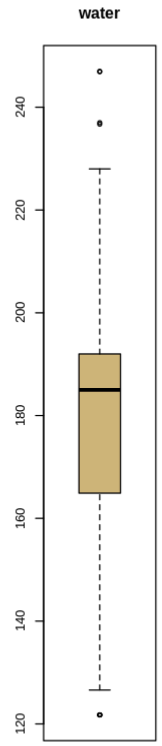
- $q_{0.25}$ is the first quartile;
- $q_{0.75}$ is the third quartile.
- IQR is the interquartile range, defined as the difference between the third and first quartile.

A boxplot will "flag" the outliers. Let's construct a boxplot for each variable and comment on the existence of potential outliers.

In [10]:

```
par(mfrow = c(1, 4))
boxplot(concrete$age, main='age',col="#CFB87C")
boxplot(concrete$blast_furnace, main='bf',col="#CFB87C")
boxplot(concrete$cement, main='cement',col="#CFB87C")
boxplot(concrete$coarse, main='coarse',col="#CFB87C")
boxplot(concrete$fine, main='fine',col="#CFB87C")
boxplot(concrete$fly_ash, main='fly ash',col="#CFB87C")
boxplot(concrete$plast, main='plasticizer',col="#CFB87C")
boxplot(concrete$strength, main='comp strength',col="#CFB87C")
boxplot(concrete$water, main='water',col="#CFB87C")
```


age
0
100
200
300
bf
0
50
100
150
200
250
300
350
cement
100
200
300
400
500
coarse
800
850
900
950
1000
1050
1100
1150
fine
600
700
800
900
1000
fly ash
0
50
100
150
200
plasticizer
0
5
10
15
20
25
30
comp strength
0
20
40
60
80



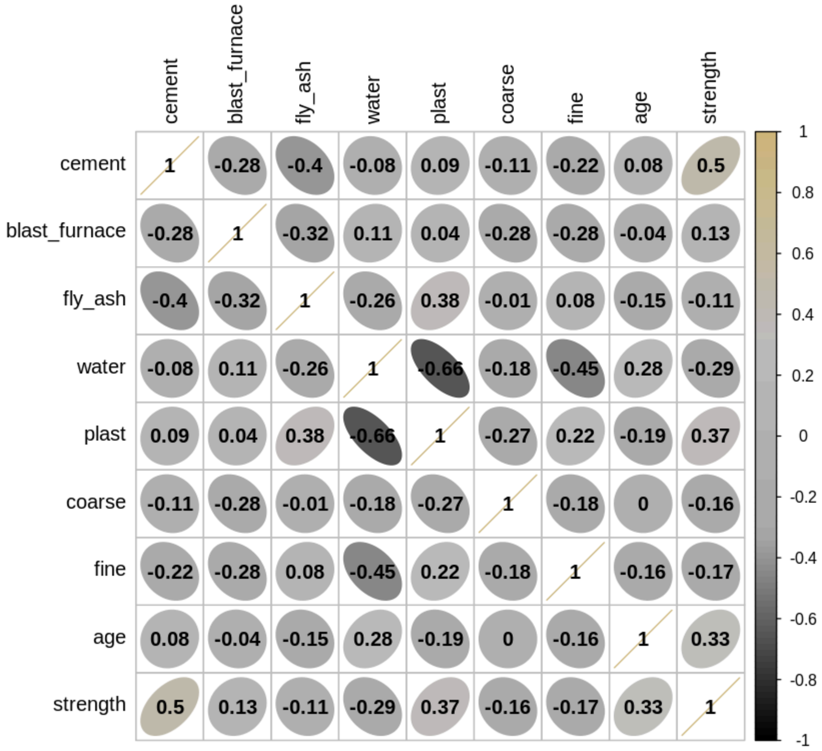
It appears from above box plots that there are few outliers in almost all of our feature variables. Let's proceed further to see any correlations existing between variables in order to avoid the issue of multi-collinearity. Also, exploring how the variables may or may not relate to each other helps us in feature selection. First, I am proceeding with calculation of correlations between variables. Correlations can help us meaasure the strength of the linear relationship between variables. For the time being, let's keep these outliers as it for all features and we simply note this for now, and note that outliers can impact the fit of a regression.

In [11]:
```
install.packages("corrplot")
library(corrplot)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

corrplot 0.92 loaded
```

In [12]:

```
col4 = colorRampPalette(c("black", "darkgrey", "grey","#CFB87C"))
corrplot(cor(concrete), method = "ellipse",
         col = col4(100),  addCoef.col = "black", tl.col = "black")
```



However, knowing correlations alone isn't enough; the correlation coefficient can be misleading if there are nonlinear relationships, and so we should explore the relationships further.

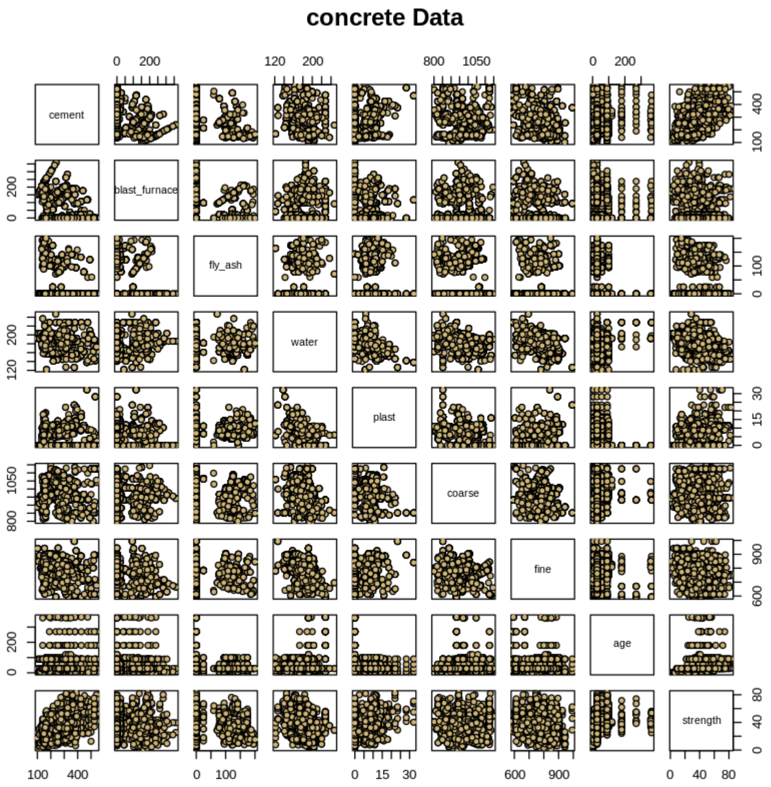
To do so, we'll look at pairwise scatter plots, i.e., a scatter plot of each variable with each other variable. We should be looking for:

- Relationships between the response and each predictor.
- Relationships between predictor variables. Such relationships are undesirable.

So, I am creating all possible pairwise scatter plots of the data.

In [13]:

```
pairs(concrete, main = "concrete Data", pch = 21,
      bg = c("#CFB87C"))
```



Here are some notes on the relationships:

- The relationship between `cement` and `strength` appears linear, with an increase in strength for an increase in `cement` . However, this relationship seems to vary as we go into higher ranges of cement though the increase in strngth is prevalent.
- There doesn't appear to be any dicernable relationship between `strength` and any of other features.
- There aren't any other strong trends in the data.

T-Test

First Hypothesis:

There is a standard practice in construction industry to keep concrete pouring hydrated for around 28-31 days. The reason behind this is: It takes about 28 days to reach 90% of its full strength i.e., ~34.4 MPa as long as its kept moist. The remaining 10% of strength is developed slowly after long period of time. If its not kept moist for 28 days it probably wont reach its design compressive strength. Thats why 28 days of curing period is recommended as a standard. Also, it is expected to develop only 40% of strength within first 3 days. Since this dataset is collected from testing various compositions of concrete at Lab, let's check whether there is statistically significant difference between means of strength of concrete whose age is <= 3 days and concrete with age >= 28 days using t-test. Below is the table for a standard concrete block to develop its compressive strength.

Null Hypothesis H_0: There is no significant difference in means of compressive strength of concrete with age $<= 3$ days and concrete with age $>= 28$ days. $\mu_{<=3d} = \mu_{>=28d}$

Alternate Hypothesis H_1: The true difference in means of compressive strength of concrete with age $<= 3$ days and concrete with age $>= 28$ days is statistically significant. $\mu_{<=3d} \neq \mu_{>=28d}$

```
In [14]: head(concrete, n=2)
```

A tibble: 2 × 9

cement	blast_furnace	fly_ash	water	plast	coarse	fine	age	strength
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
540	0	0	162	2.5	1040	676	28	79.98611
540	0	0	162	2.5	1055	676	28	61.88737

```
In [15]: age_28_data <- subset(concrete, age >= 28)

         mean_strength_28 <- mean(age_28_data$strength)

         print(mean_strength_28)
```

```
[1] 41.45192
```

```
In [16]: age_5_data <- subset(concrete, age <= 3)

         mean_strength_5 <- mean(age_5_data$strength)

         print(mean_strength_5)
```

```
[1] 18.84096
```

```
In [70]: t_test_result <- t.test(age_28_data$strength, age_5_data$strength, alternat
         ive = "two.sided")
```

```
In [71]: t_test_result
```

```
        Welch Two Sample t-test

data:  age_28_data$strength and age_5_data$strength
t = 22.073, df = 278.76, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 20.59446 24.62747
sample estimates:
mean of x mean of y 
 41.45192  18.84096 
```

Inference: From the result of t-test, we can infer that p-value $(= 2.26 \times 10^-16)$ < $\alpha (= 0.05)$. This indicates that we reject null hypothesis and there is no evidence for the claim made in null hypothesis statement i.e., means are not significantly different. Thus, we can say that true difference in means is statistically significant.

Second Hypothesis:

In general, the purpose of adding blast furnace slag is to enhance strength and durability of concrete. So, keeping all others factors constant, the concrete block is ideally expected to reach more strengths at age of 28 days than those without any slag at same age. Let's check this with oor t-test. Now, let's check this using T-test. Before moving on further, lets also look at distributions of subsetted data to identify any significant anamoly.

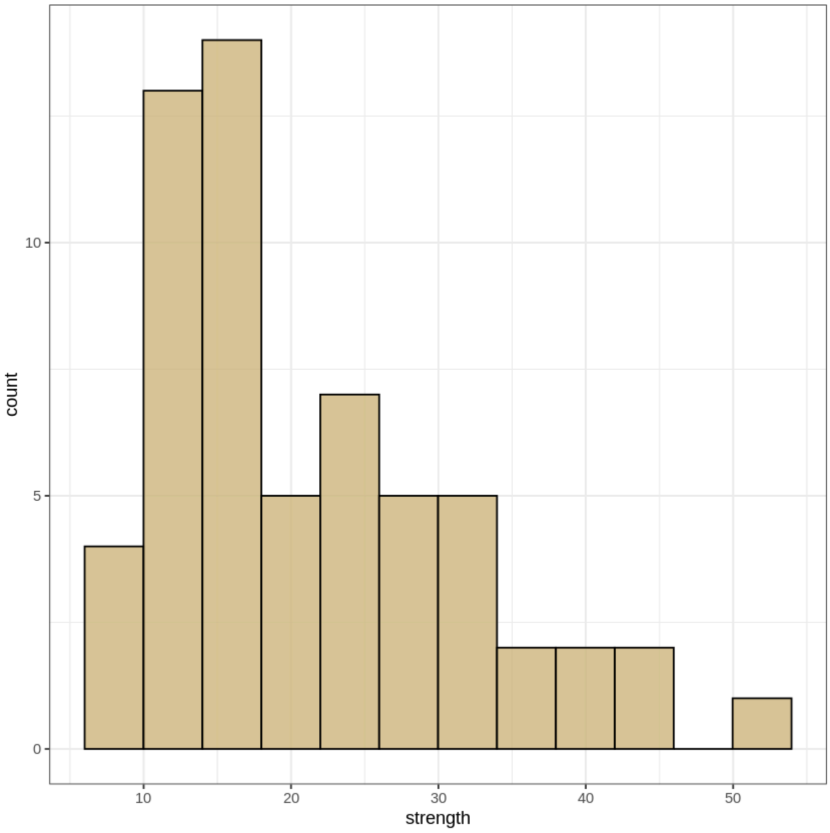
```
In [20]: bf_0 <- subset(concrete, blast_furnace==0 & age==28 & cement<= 250)
```

```
In [21]: bf_not_0 <- subset(concrete, blast_furnace!=0 & age==28 & cement <= 250)
```

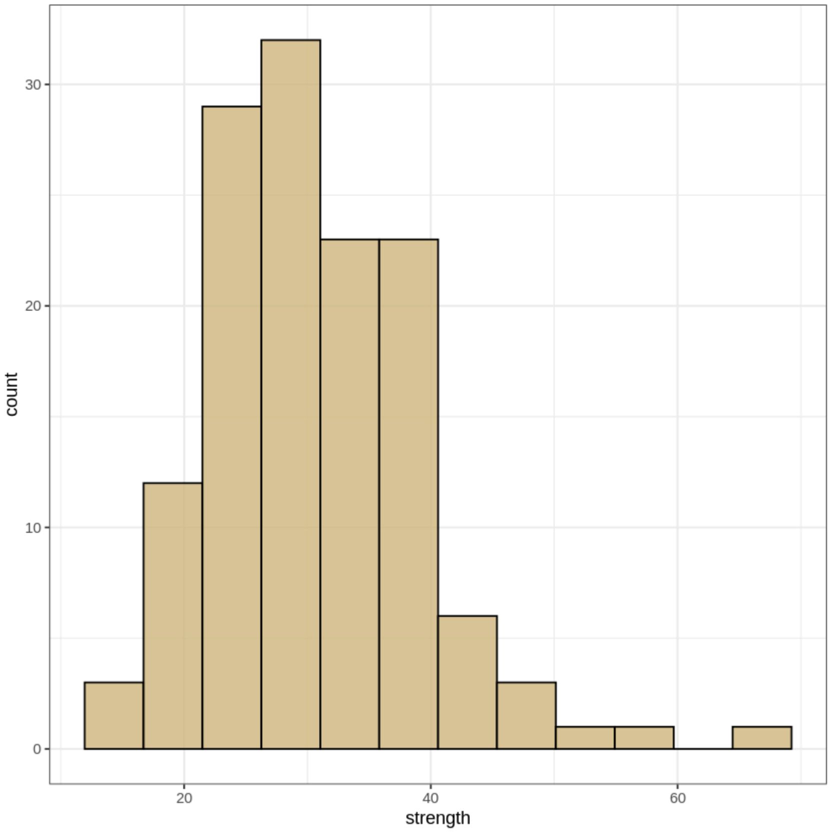
```
In [22]: cat("Number of datapoints:", dim(bf_0)[1],",",dim(bf_not_0)[1])
```

```
Number of datapoints: 60 , 134
```

```
In [23]: ggplot(data = bf_0) +
           geom_histogram(bins = 12, color="black", fill ="#CFB87C", alpha = 0.8, ae
         s(strength)) +
           theme_bw()
```



In [24]:
```
ggplot(data = bf_not_0) +
  geom_histogram(bins = 12, color="black", fill ="#CFB87C", alpha = 0.8, ae
s(strength)) +
  theme_bw()
```



Both the distributions are not normal, we will carry out hypothesis testing using t-test.

Null Hypothesis H_0: There is no significant difference in means of compressive strength of concrete with addition of `blast furnace slagage` and those without any `blast furnace slag` at same age and same range of cement proportions. $\mu_{noslag} = \mu_{slag}$.

Alternate Hypothesis H_1: The true difference in means of compressive strength of concrete with `slag` and without `slag` is statistically significant. $\mu_{noslag} < \mu_{slag}$ with all factors constant.

In [72]:
```
t.test(bf_0$strength, bf_not_0$strength, alternative = "less")
```

```
	Welch Two Sample t-test

data:  bf_0$strength and bf_not_0$strength
t = -6.1519, df = 96.506, p-value = 8.682e-09
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -6.734656
sample estimates:
mean of x mean of y 
 21.49030  30.71543 
```

Inference: From the result of t-test, we can infer that p-value $(= 8.68 \times 10^-9)$ < $\alpha(= 0.05)$. This indicates that we reject null hypothesis. Thus, there is statistically significant difference in means of strengths of concrete with and without blast furnace slag addition. We can colcude that mean of strengths of concrete without slag is less than mean of strength of concrete with slag addition.

F-Test

In order to determine whether atleast one of the predictors is necessary in predicting response `strength` , a full F-test first is carried out. Null and alternate hypothesis statements for F-test are as below.

Null Hypothesis: $Y_i = \beta_0 + \epsilon_i$. Reduced model is sufficient i.e., None of these features can help in predicting response variable.

Alternate Hypothesis: $H_a =$ Atleast one among the predictors i.e., $\beta_j \neq 0$. Atleast one of these predictors is significant in predicting response variable.

In [26]:

```
null_model <- lm(strength ~ 1, data = concrete)
full_model <- lm(strength ~ cement + blast_furnace + fly_ash + water + plas
t + coarse + fine + age, data = concrete)
```

In [27]:

```
anova(null_model, full_model)
```

A anova: 2 × 6

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1029	287173.0	NA	NA	NA	NA
2	1021	110428.2	8	176744.9	204.2691	6.761578e-206

Note that the p-value associated with this full F-test is very small ($6.76 \times 10^{-206} < \alpha = 0.05$). Consequently, we infer that there is compelling evidence suggesting the inadequacy of the reduced model and the necessity for at least one additional predictor. Given that this F-test helps in recognizing the significance of at least one predictor in forecasting our response variable, I am moving forward with regression modeling.

Regression Modelling

In [28]:

```
set.seed(11)
n = floor(0.8 * nrow(concrete)) #find the number corresponding to 80% of the data
index = sample(seq_len(nrow(concrete)), size = n) #randomly sample indicies to be included in the training set

train = concrete[index, ] #set the training set to be the randomly sampled rows of the dataframe
test = concrete[-index, ] #set the testing set to be the remaining rows
cat("There are", dim(train)[1], "rows and",dim(train)[2],"columns in the training set. ")  #check the dimensions
cat("There are", dim(test)[1], "rows and",dim(test)[2],"columns in the testing set.")  #check the dimensions
```

```
There are 824 rows and 9 columns in the training set. There are 206 rows and 9 columns in the testing set.
```

In [29]:

```
lm_concrete = lm(strength ~ cement + blast_furnace + fly_ash + water + plast + coarse + fine + age , data = train)
summary(lm_concrete)
```

```
Call:
lm(formula = strength ~ cement + blast_furnace + fly_ash + water + 
    plast + coarse + fine + age, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-28.308  -6.497   0.791   6.663  34.448 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -13.347031  29.475307  -0.453 0.650798    
cement          0.114098   0.009529  11.974  < 2e-16 ***
blast_furnace   0.100093   0.011419   8.765  < 2e-16 ***
fly_ash         0.084308   0.014121   5.970 3.53e-09 ***
water          -0.160331   0.044175  -3.629 0.000302 ***
plast           0.273957   0.106048   2.583 0.009958 ** 
coarse          0.015616   0.010434   1.497 0.134867    
fine            0.015847   0.011906   1.331 0.183558    
age             0.113417   0.005996  18.917  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.43 on 815 degrees of freedom
Multiple R-squared:  0.5959,	Adjusted R-squared:  0.592 
F-statistic: 150.3 on 8 and 815 DF,  p-value: < 2.2e-16
```

From above output summary, we can write our MLR model as follows:

$$\widehat{\beta}_0 = -13.34, \widehat{\beta}_1 = 0.11, \widehat{\beta}_2 = 0.1, \widehat{\beta}_3 = 0.08, \widehat{\beta}_4 = -0.16, \widehat{\beta}_5 = 0.27, \widehat{\beta}_6 = 0.015, \widehat{\beta}_7 = 0.015,$$

$$strength = -13.34 + 0.11 \times \text{cement} + 0.1 \times \text{blast\_furnace} + 0.080.11 \times \text{fly\_ash} - 0.16 \times \text{wate}$$
$$\times \text{ fine} + 0.11 \times \text{age}$$

Inference: As we know that p-value of features `coarse` and `fine` are higher than 0.05, we consider these as not significant in predicting our response variable `strength` . So, new MLR would be framed based on removing those predictors only and will continue till all predictors are significant.

In [30]:
```r
lm_concrete = update(lm_concrete, .~.-coarse)
summary(lm_concrete)
```

```
Call:
lm(formula = strength ~ cement + blast_furnace + fly_ash + water +
    plast + fine + age, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-29.308  -6.517   0.803   6.751  34.902

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   28.337628   9.654991   2.935  0.00343 **
cement         0.103011   0.005998  17.173  < 2e-16 ***
blast_furnace  0.086574   0.006992  12.382  < 2e-16 ***
fly_ash        0.069638   0.010173   6.845  1.5e-11 ***
water         -0.214398   0.025443  -8.426  < 2e-16 ***
plast          0.219947   0.099795   2.204  0.02780 *
fine           0.001121   0.006709   0.167  0.86735
age            0.112753   0.005984  18.843  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.44 on 816 degrees of freedom
Multiple R-squared:  0.5948,    Adjusted R-squared:  0.5914
F-statistic: 171.1 on 7 and 816 DF,  p-value: < 2.2e-16
```

We can see here that p-val of predictor `fine` < 0.05, it is not significant in predicting output and will take one more iteration to remove this and conduct regression.

In [31]:
```
lm_concrete = update(lm_concrete, .~.-fine)
summary(lm_concrete)
```

```
Call:
lm(formula = strength ~ cement + blast_furnace + fly_ash + water + 
    plast + age, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.396  -6.532   0.772   6.753  34.827 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   29.750200   4.659992   6.384 2.89e-10 ***
cement         0.102400   0.004752  21.548  < 2e-16 ***
blast_furnace  0.085881   0.005625  15.268  < 2e-16 ***
fly_ash        0.068761   0.008710   7.894 9.40e-15 ***
water         -0.216029   0.023484  -9.199  < 2e-16 ***
plast          0.224069   0.096638   2.319   0.0207 *  
age            0.112658   0.005953  18.924  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.43 on 817 degrees of freedom
Multiple R-squared:  0.5948,	Adjusted R-squared:  0.5918 
F-statistic: 199.9 on 6 and 817 DF,  p-value: < 2.2e-16
```

Notice that each of our remaining predictors have a p-value less than $\alpha_0 = 0.05$. Now, that all our predictors are significant in predicting response varaible and coefficients are also changed. This is process of backward selection. Now, our revised MLR model can be written as -

$$strength = 29.75 + 0.102 \times \text{cement} + 0.086 \times \text{blast\_furnace} + 0.069 \times \text{fly\_ash} - 0.216 \times \text{water}$$

Assuming linear model is correct, "adjusted R-squared", R^2: 0.592, Only 59.2\% proportion of observed variability in `strength` is explained by this linear regression model. It raises a suspicion that this is not a right fit and let's check this using diagnostic plots. Let's first plot this visually with one variable at a time for better appreciation. Also, I will be using AIC, BIC, R-sq to identify best models for each size of k.

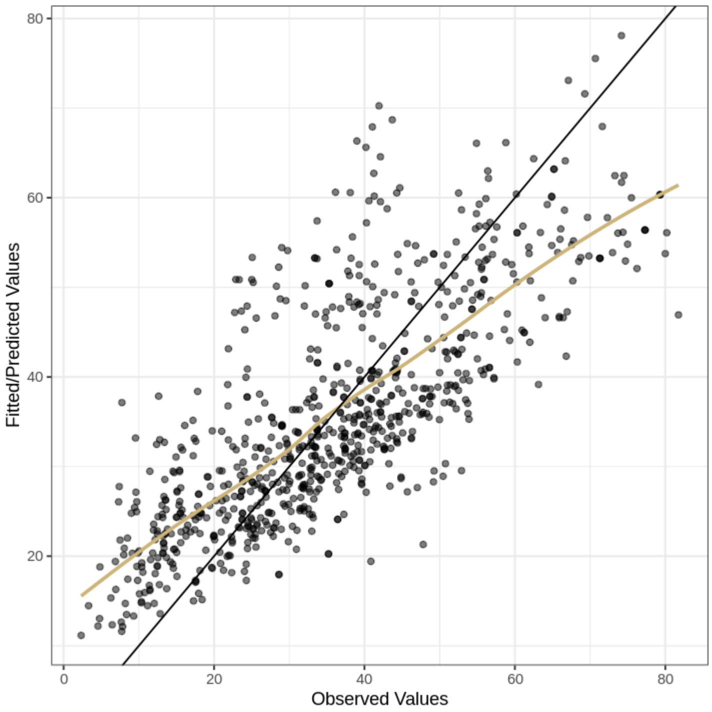
Diagnostics

In [32]:
```
df.diagnostics = data.frame(yhat = fitted(lm_concrete), r = resid(lm_concre
te), y = train$strength)
```

Observed Vs Fitted

```
In [33]: options(repr.plot.width = 6, repr.plot.height = 6)
         ggplot(df.diagnostics, aes(x = y, y = yhat)) +
             geom_point(alpha = 0.5) +
             geom_smooth(se = F, col = "#CFB87C") +
             geom_abline(intercept = 0, slope = 1)+
             xlab("Observed Values") +
             ylab("Fitted/Predicted Values") +
             theme_bw()
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



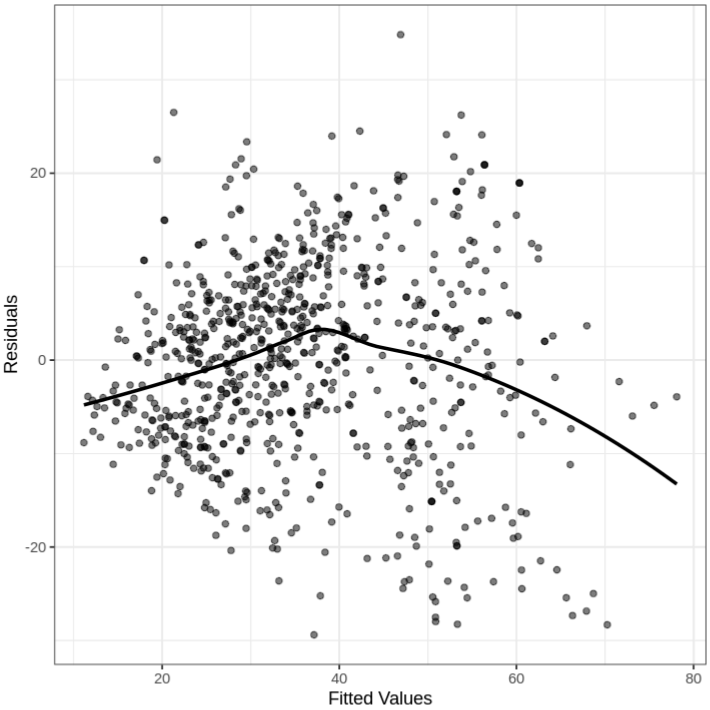
The observed vs predicted value plot should follow the black line, $y = x$, with only random deviations. Instead, the plot shows some deviation, as captured by the gold curve. This curvature shows that, for high values of the response, say, above about 50, the model is underpredicting (because the fitted values are lower than the observed values). For middle and lower values of the response, there is an overprediction, but with more variability. All of this suggests that the structural part of the model is misspecified.

Now let's look at the residual vs fitted values plot.

Fitted Vs Residuals

```
In [34]: ggplot(df.diagnostics, aes(x = yhat, y = r)) +
             geom_point(alpha = 0.5) +
             geom_smooth(se = F, col = "black") +
             xlab("Fitted Values") + ylab("Residuals")+
             theme_bw()
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

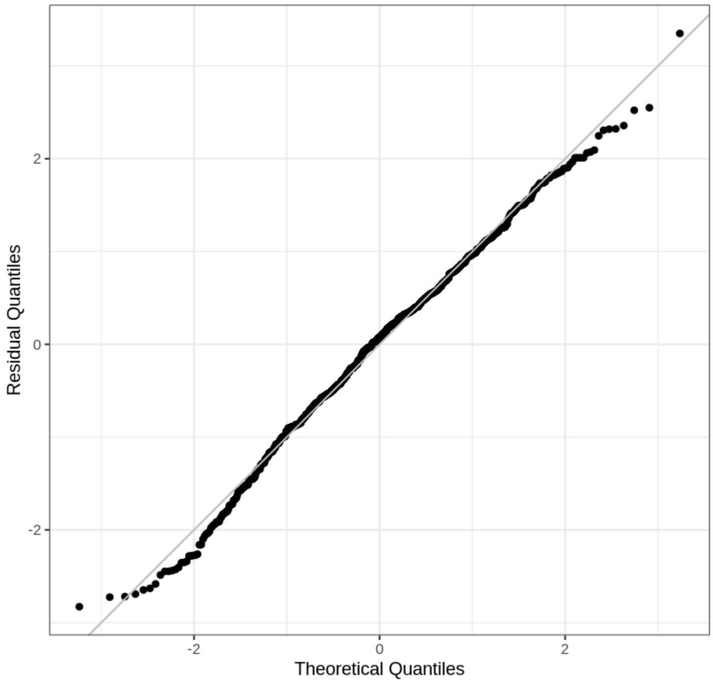


Here, we corroborate our result from the observed vs fitted plot: that there is some curvature in the data not being captured by our model. we don't seem to see a difference in variability of the residuals across the fitted values. So, we can see that there is slight funnel shape appearing which indicates evidence of a violation in the nonconstant variance assumption. The ideal condition of homoscedasticity is violated.

Given the structural issues with the model, it is difficult to use this plot to assess the normality assumption. But let's take a quick look at a qq plot:

QQ Plot

In [35]:
```
ggplot(df.diagnostics, aes(sample = (r - mean(r))/sd(r))) +
    stat_qq() + geom_abline(slope = 1, intercept = 0, col = "grey") +
    xlab("Theoretical Quantiles") +
    ylab("Residual Quantiles") +
    ggtitle("") +
    theme_bw()
```

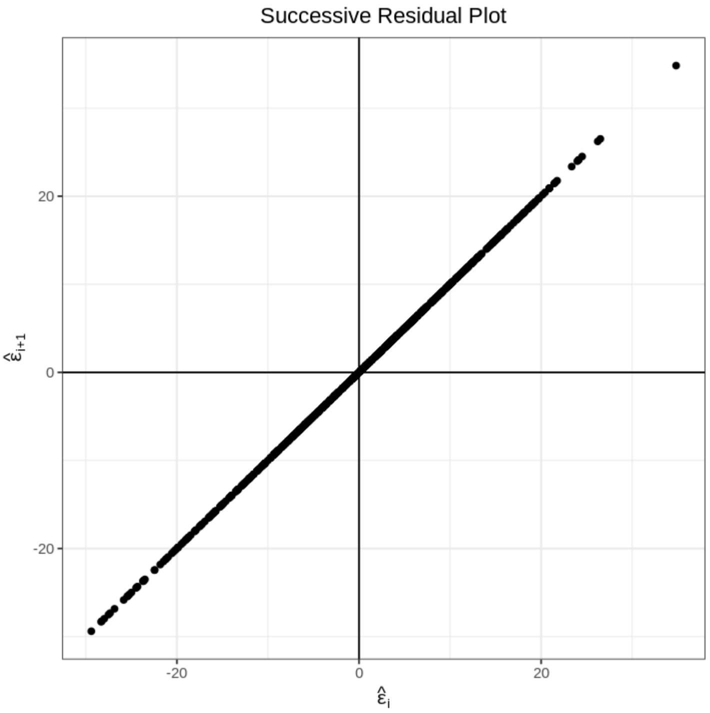


From above QQ plot, we an say that residuals seem to be normal with slighter devaitions in the ends. However, these consequences are not serious enough to reject normality. Hence, we can say that assumption of normality is satisfied.

Independence

In [36]:
```
n = dim(concrete)[1];
x = head(df.diagnostics$r, n-1)
y = tail(df.diagnostics$r, n-1)
cor(x,y)
srp = data.frame(x,y)
ggplot(srp, aes(x = x, y = y)) +
    geom_point() +
    geom_vline(xintercept = 0) +
    geom_hline(yintercept = 0) +
    xlab(expression(hat(epsilon)[i])) +
    ylab(expression(hat(epsilon)[i+1])) +
    ggtitle("Successive Residual Plot") +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5))
```

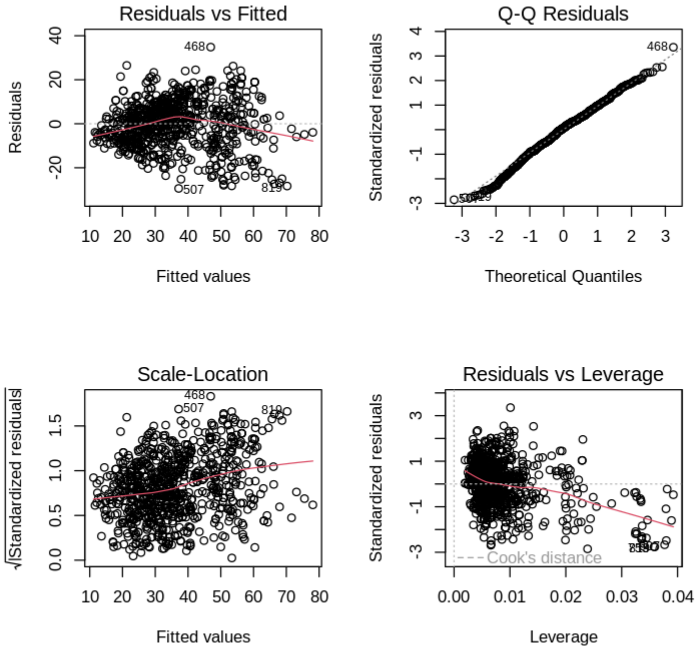
1



If the independence assumption is satisfied, the residuals will be randomly scattered around zero; there shouldn't be any correlation between successive errors. But there appears a clear cut evidence of linear association between successive residuals. This might be due to the nonlinearity in the data or structural problems of the fit.

In [37]:

```
par(mfrow=c(2,2))
plot(lm_concrete)
```



The levergae Vs residuals plot doesn't indicate presence of any potential influential points. However, there are some large leverage points, which indicates chances of these becoming potentail influential points.

Model selection using AIC, BIC, and adjusted R^2

In [38]:

```
install.packages("leaps")
library(leaps)
library(MASS)

n = dim(concrete)[1]
reg1 = regsubsets(strength ~ cement + blast_furnace + fly_ash + water + pla
st + coarse + fine + age, data=train)
rs = summary(reg1)
rs$which
```

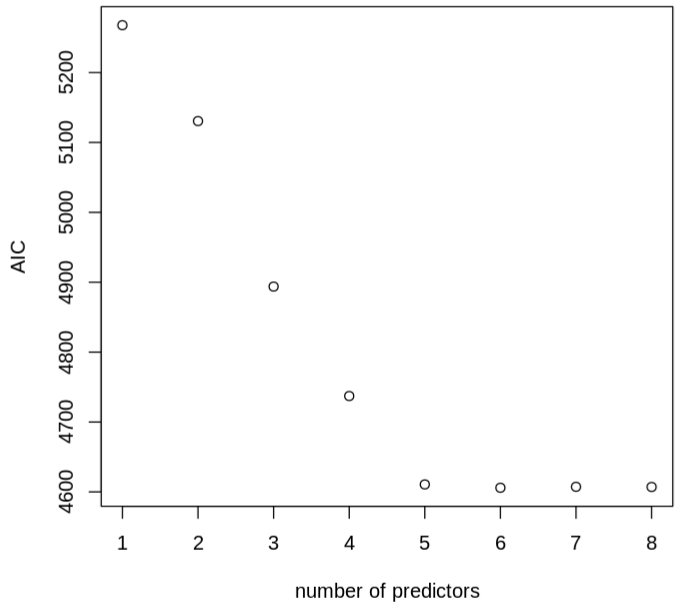
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

A matrix: 8 × 9 of type lgl

	(Intercept)	cement	blast_furnace	fly_ash	water	plast	coarse	fine	age
1	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
3	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
4	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
5	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE
6	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE
7	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
8	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

The table above provides the best model (in terms of RSS) of size k, for $k = 1, 2, \ldots, 6$. For example, the best simple linear regression model is the model `strength` = $\widehat{\beta}_0 + \widehat{\beta}_1 \times$ `cement` . Now, to compare these models with each other, we calculate AIC, and plot the AIC values as a function of model size.

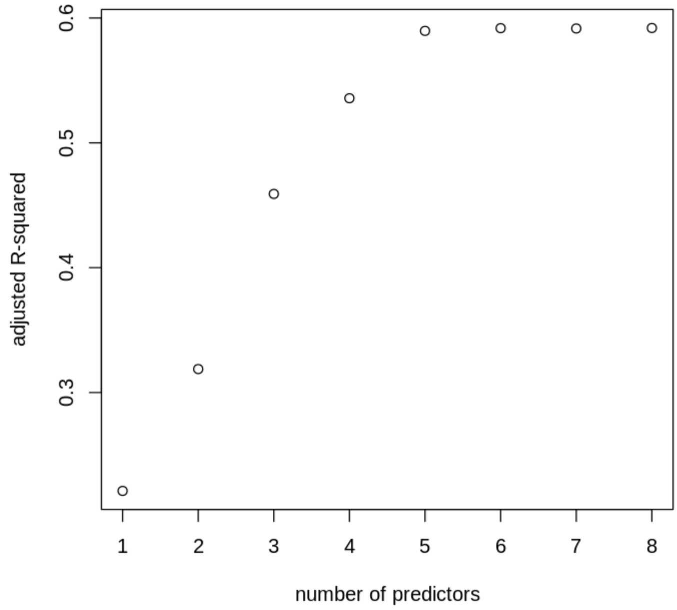
In [39]:
```
AIC = 2*(2:9) + n*log(rs$rss/n)
plot(AIC ~ I(1:8), xlab = "number of predictors", ylab = "AIC")
```



In this plot, we see that the model of size $k = 5, 6, 7, 8$ has the lowest AIC. Since the curve looks almost flat after $k = 5$, we will go with five predictors as this is less complicated. That means that our model selection procedure has chosen:

strength $= \widehat{\beta}_0 + \widehat{\beta}_1 \times$ cement $+ \widehat{\beta}_2 \times$ blast_furnace $+ \widehat{\beta}_3 \times$ fly_ash $+ \widehat{\beta}_4 \times$ water $+ \widehat{\beta}_5 \times$ age .

In [40]:
```
plot(1:8, rs$adjr2, xlab = "number of predictors", ylab = "adjusted R-squar
ed")
```

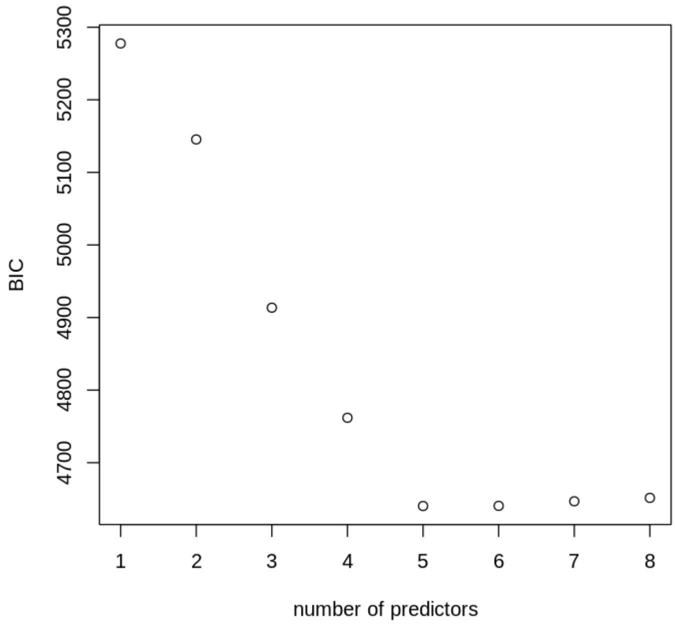


Interestingly, using

$$R_a^2 = 1 - \frac{RSS/\left(n-(p+1)\right)}{TSS/\left(n-1\right)},$$

we got same model with five predictors (`cement` , `balst_furnace` , `fly_ash` , `water` , `age`). Now, let's see what BIC would suggest

```
In [41]: BIC = log(n)*(2:9) + n*log(rs$rss/n)
         plot(BIC ~ I(1:8), xlab = "number of predictors", ylab = "BIC")
```



This is also the same model as suggested by both AIC and R-Sq. Here, in our case, all three models suggested by AIC, BIC and R^2 are same with five predictors.

`strength` = $\widehat{\beta}_0 + \widehat{\beta}_1\times$ `cement` $+\widehat{\beta}_2\times$ `blast_furnace` $+\widehat{\beta}_3\times$ `fly_ash` $+\widehat{\beta}_4\times$ `water` $+\widehat{\beta}_5\times$ `age` .

Now. let us check the evidence of collinearity between our predictors using VIF and condition number.

In [42]:
```
install.packages("car")
library(car)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘numDeriv’, ‘SparseM’, ‘MatrixModels’, ‘mi
nqa’, ‘nloptr’, ‘Rcpp’, ‘RcppEigen’, ‘carData’, ‘abind’, ‘pbkrtest’, ‘quant
reg’, ‘lme4’


Loading required package: carData


Attaching package: ‘car’


The following object is masked from ‘package:purrr’:

    some
```

In [43]:
```
lm_concrete_aic = lm(strength ~ cement + blast_furnace + fly_ash + water +
age, data=train)
vif(lm_concrete_aic)
kappa(lm_concrete_aic)
cor(model.matrix(lm_concrete_aic)[,-1])
```

cement: 1.58056636599228 **blast_furnace:** 1.45566619002869 **fly_ash:** 1.75880850360963 **water:** 1.19627733822943 **age:** 1.10531581566171

2393.83584817497

A matrix: 5 × 5 of type dbl

	cement	blast_furnace	fly_ash	water	age
cement	1.00000000	-0.2785267	-0.3970881	-0.0477334	0.06949158
blast_furnace	-0.27852674	1.0000000	-0.3266007	0.1137684	-0.03599440
fly_ash	-0.39708814	-0.3266007	1.0000000	-0.2750419	-0.15849398
water	-0.04773340	0.1137684	-0.2750419	1.0000000	0.27961000
age	0.06949158	-0.0359944	-0.1584940	0.2796100	1.00000000

In [44]:
```
summary(lm_concrete_aic)
```

```
Call:
lm(formula = strength ~ cement + blast_furnace + fly_ash + water + 
    age, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-31.934  -6.475   0.721   6.388  33.608 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   35.048513   4.072112   8.607   <2e-16 ***
cement         0.106651   0.004396  24.261   <2e-16 ***
blast_furnace  0.091332   0.005124  17.825   <2e-16 ***
fly_ash        0.078858   0.007564  10.425   <2e-16 ***
water         -0.249513   0.018568 -13.438   <2e-16 ***
age            0.113081   0.005966  18.953   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.46 on 818 degrees of freedom
Multiple R-squared:  0.5922,	Adjusted R-squared:  0.5897 
F-statistic: 237.5 on 5 and 818 DF,  p-value: < 2.2e-16
```

Model chosen by AIC, BIC, adj R-Sq : `strength` = $35.04 + 0.106\times$ `cement` $+0.091\times$ `blast_furnace` $+0.078\times$ `fly_ash` -0.249 `water` $+0.113$ `age` .

For this model, we see:

1. The VIF for the estimators are below 5, which is not evidence of collinearity.
2. The condition number is still very high ($>> 30$), suggesting collinearity is an issue.
3. The correlation matrix for the predictors doesn't show any high pairwise correlations.

Now, let's calculate MSPE for predicted values for both models suggested by backward selection and ACI/BIC/R^2.

In [45]:
```
back_pred = predict(lm_concrete, newdata=test)
```

In [46]:
```
squared_residuals <- (test$strength - back_pred)^2
MSPE_back <- mean(squared_residuals)

MSPE_back
```

107.13772481423

In [47]:
```
aic_pred = predict(lm_concrete_aic, newdata=test)
```

```
In [48]: squared_residuals <- (test$strength - aic_pred)^2
         MSPE_aic <- mean(squared_residuals)

         MSPE_aic
```

```
108.515143239183
```

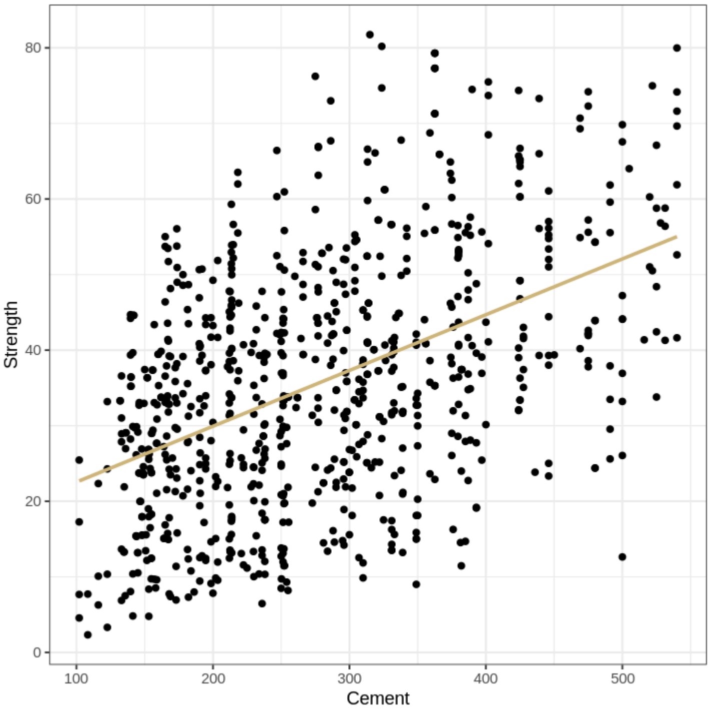
There is no much difference in MSPEs obtained from test datasets. However, upon reviewing diagnostic plots, there is a slight suspicion regarding the linearity of the model. To investigate this further, let's examine Generalized Additive Models (GAMs). MSPE vlues for models suggested

Digging deep: Non-Linearity

Plotting with cement

```
In [49]: ggplot(train, aes(x = cement, y = strength)) +
             geom_point() +
             geom_smooth(method = "lm", col = "#CFB87C", se = F) +
             theme_bw() +
             xlab("Cement") +
             ylab("Strength")
```

```
`geom_smooth()` using formula = 'y ~ x'
```

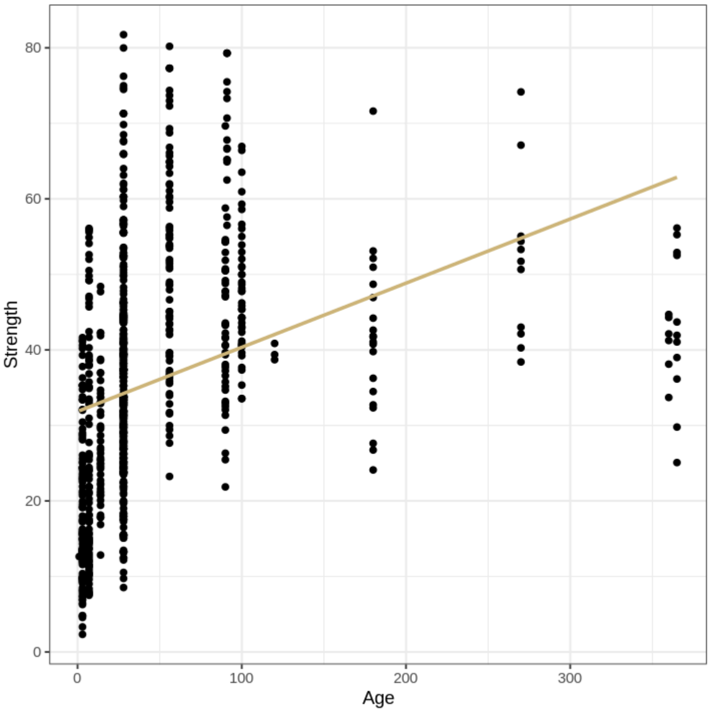


Though the plot looks reasonably linear with predictor `cement`, when we bring other predictors into regression, it appears to be not performing well, no more linear. Let's plot with other variables too.

Plotting with age

In [50]:
```
ggplot(train, aes(x = age, y = strength)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C", se = F) +
    theme_bw() +
    xlab("Age") +
    ylab("Strength")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Here is some interesting thing to make note of. At some ranges of age, there seems to be some pattern existing between `cement` and `strength`. Let's plot `cement` Vs `strength` at different values of ages and observe how it is varying at each age bracket.

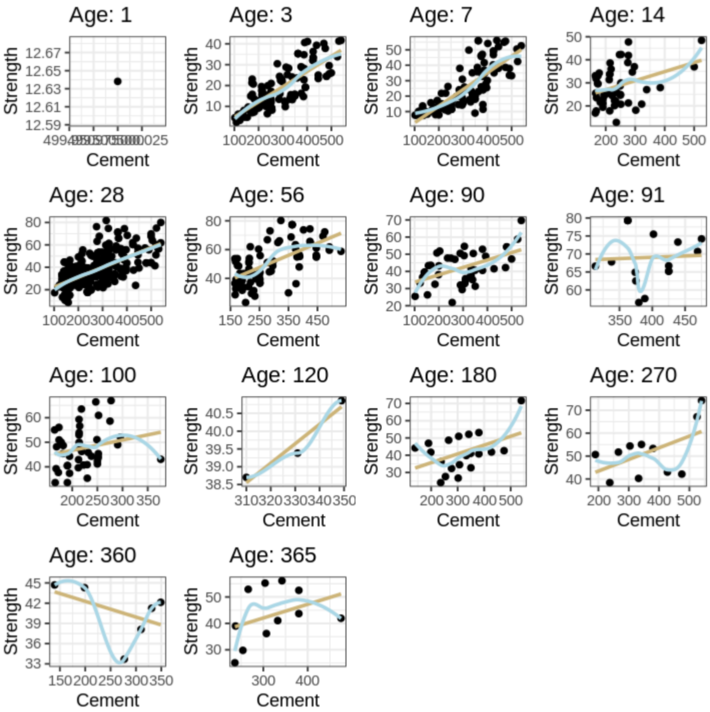
In [51]:
```
# finding unique value so that we can plot at all ages

unique_age <- sort(unique(concrete$age))
unique_age
```

```
1 · 3 · 7 · 14 · 28 · 56 · 90 · 91 · 100 · 120 · 180 · 270 · 360 · 365
```

In [52]:

```
library(ggplot2)
install.packages("gridExtra")
library(gridExtra)  # Load gridExtra package

plot_list <- list()

for (age_ in unique_age) {
  subset_data <- subset(train, age == age_)

  plot <- ggplot(subset_data, aes(x = cement, y = strength)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C", se = FALSE) +
    geom_smooth(method = "loess", col = "lightblue", se = FALSE) +
    theme_bw() +
    xlab("Cement") +
    ylab("Strength") +
    ggtitle(paste("Age:", age_))

  plot_list[[as.character(age_)]] <- plot
}

ncol <- ceiling(sqrt(length(plot_list)))

# Combine all plots into a single plot grid
grid.arrange(grobs = plot_list, ncol = ncol)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
“span too small.   fewer data values than degrees of freedom.”
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
“pseudoinverse used at 309.81”
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
“neighborhood radius 21.195”
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
“reciprocal condition number  0”
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
“There are other near singularities as well. 331.06”
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



I found an interesting pattern in data that at every age, there is some clearcut relationship emerging out which we couldnot see when we plotted `cement` Vs `strength` for all ages in single plot. ***There is noticeable trend that at early ages, relation seems to be in increasing trend linearly, and at middle & higher ages it is gradually becoming non-linear but still in increasing trend***.

Let's see the anamolous behaviour of strength at $age = 360$, to understand why there is a decreasing trend. Below is the filtered dataset at $age = 360$. For cement=139.6 and 198.6, the component of blast furnace slag was mixed as an additive to concrete, which caused strength to increase tremendously even at lower values of cement. If we see the trend for **balance 4 datapoints**, it still **conforms** to our **interesting pattern** what we found above. These 4 points are shown separately with light blue line in below graph for better visualisation. So, we can infer that if we isolate effect of blast_furnace slag addition, compressive strength of concrete keeps on increasing with cement at specific age.

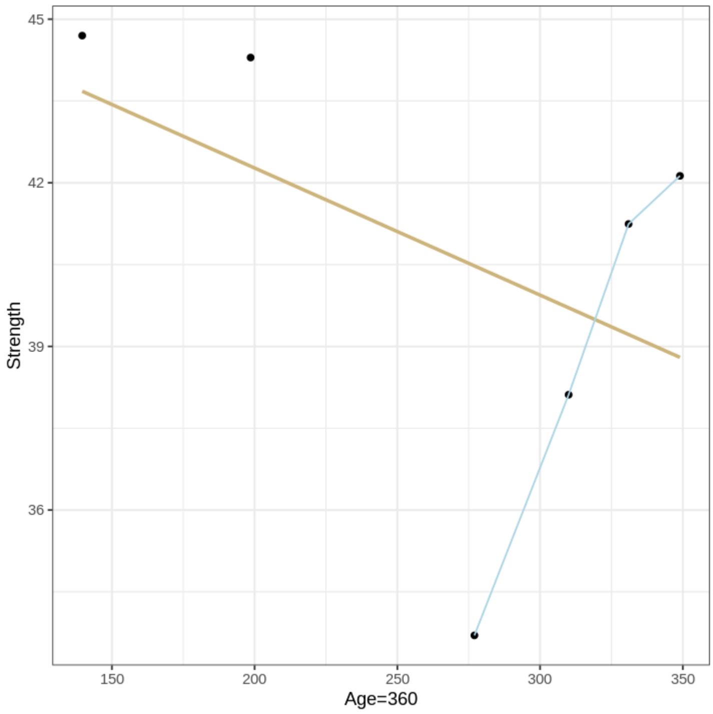
In [53]:
```
subset(train, age == 360)
```

A tibble: 6 × 9

cement	blast_furnace	fly_ash	water	plast	coarse	fine	age	strength
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
139.6	209.4	0	192	0	1047.0	806.9	360	44.69804
277.0	0.0	0	191	0	968.0	856.0	360	33.70159
331.0	0.0	0	192	0	978.0	825.0	360	41.24445
349.0	0.0	0	192	0	1047.0	806.0	360	42.12698
310.0	0.0	0	192	0	970.0	850.0	360	38.11423
198.6	132.4	0	192	0	978.4	825.5	360	44.29608

In [54]:
```
ggplot(subset(train, age == 360), aes(x = cement, y = strength)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C", se = F) +
    geom_line(data = subset(train, age == 360 & blast_furnace == 0), aes(x
= cement, y = strength), col = 'lightblue') +
    theme_bw() +
    xlab("Age=360") +
    ylab("Strength")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



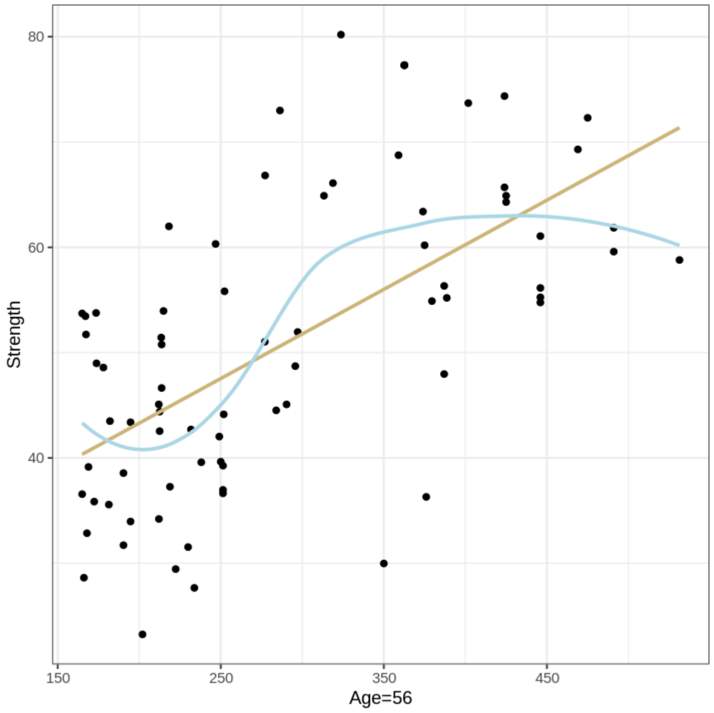
The above clearly shows the impact of addition of blast furnace slag to concrete. It drasticaaly improved strength even at lower proportions of cement. The blue curve indicates increasing trend conforming to our interesting pattern.

As we described in introduction to dataset section, dataset was collected by varying components proportion to see its effectiveness and understand relationship. Let's zoom in to see what's happening at middle and higher ages.

In [55]:

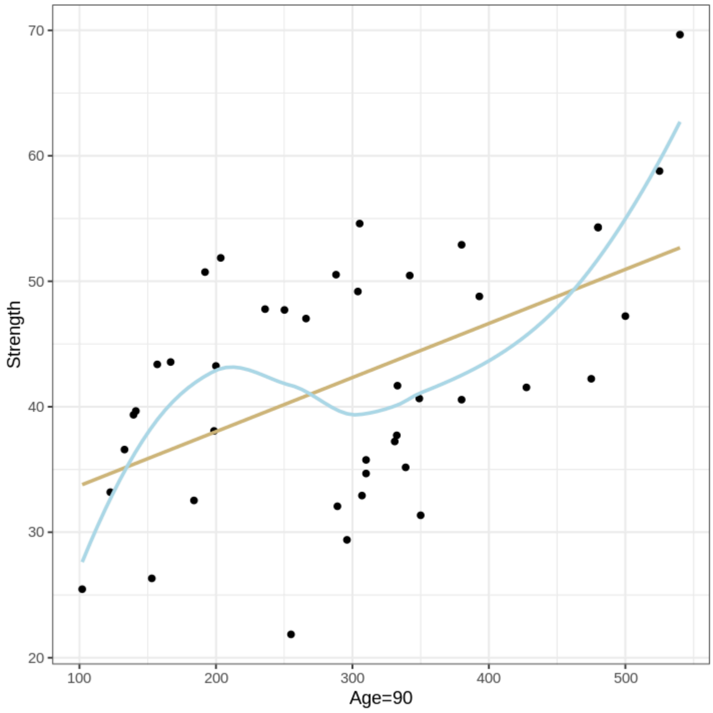
```
ggplot(subset(train, age == 56), aes(x = cement, y = strength)) +
    geom_point() +
    geom_smooth(method = "lm", col = "#CFB87C", se = F) +
    geom_smooth(method = "loess", col = "lightblue", se = F) +
    theme_bw() +
    xlab("Age=56") +
    ylab("Strength")
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



```
In [56]: ggplot(subset(train, age == 90), aes(x = cement, y = strength)) +
             geom_point() +
             geom_smooth(method = "lm", col = "#CFB87C", se = F) +
             geom_smooth(method = "loess", col = "lightblue", se = F) +
             theme_bw() +
             xlab("Age=90") +
             ylab("Strength")
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



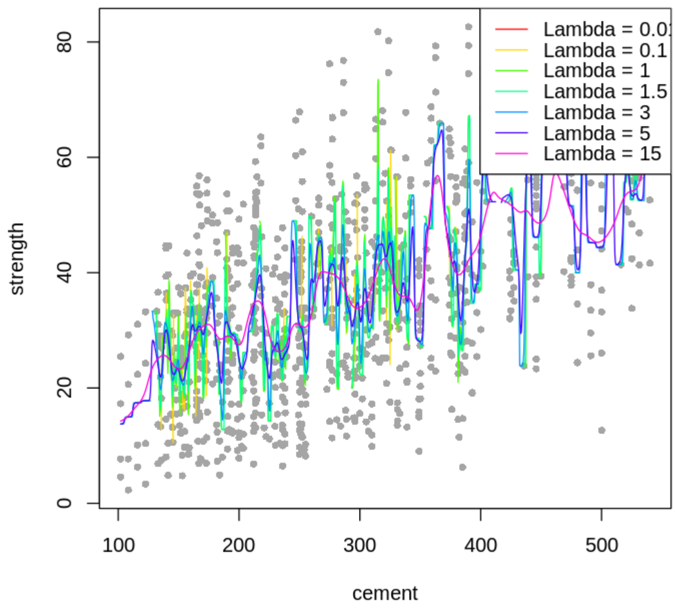
From above plots, we can clearly see that there exists a non-linear relationship between cement and strength. In order to cater for multiple functions between response and predictors, we need to use GAM, generative additive model as it allows for more complex relationships and capturing more patterns.

GAMs and diagnostic plots

Since best one-predictor model as suggested is with predictor `cement` , I'm initiating exploration by focusing on this predictor. If the curve fits suitably, we can then consider incorporating additional predictors as well.

In [57]:

```
with(concrete, plot(strength ~ cement, pch = 16, cex=0.8, col = "darkgrey"))

lambda_values <- c(0.01, 0.1, 1, 1.5, 3, 5, 15)

for (lambda in lambda_values) {
  fit <- ksmooth(concrete$cement, concrete$strength, "normal", bandwidth = lambda)

  lines(fit, col = rainbow(length(lambda_values))[which(lambda_values == lambda)])
}

# Add legend
legend("topright", legend = paste("Lambda =", lambda_values), col = rainbow(length(lambda_values)), lty = 1)
```

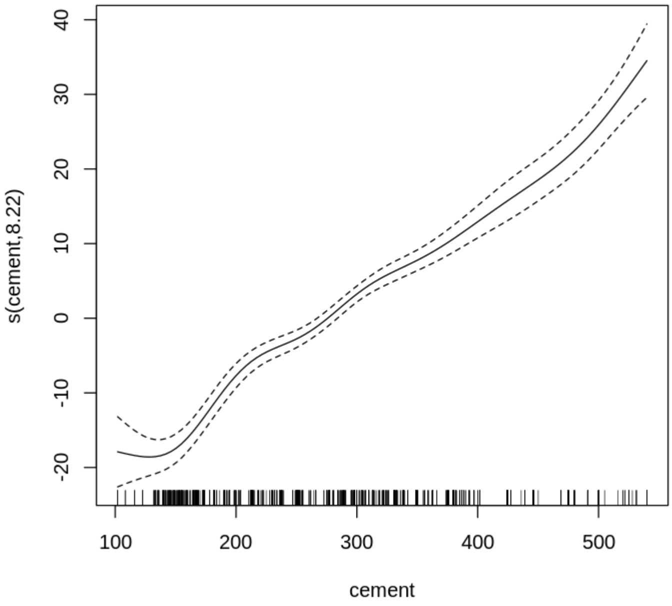


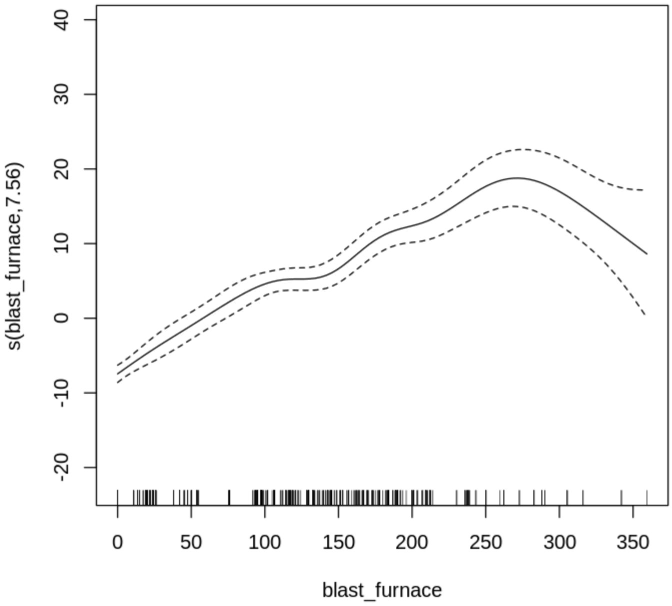
From the above plots, we can see that none of the curve is properly fitting the distribution. When λ is too small like 0.01, 0.1, cuvre has heavy wiggleness and is trying to pass through every point in dataset which is sign of overfitting. When we increase λ in steps, though curve is capturing the overall trend of distribution, but there is a risk of missing some important trend in thd data, which is a signal of underfitting. This happens when the relationship between the response variable and the predictor variable is highly non-linear or complex. In these type of situations, we may try alternative approaches for modeling non-linear relationships, such as using spline regression or generalized additive models (GAMs). So, either it is overfitting or underfitting at different values of λ especially in higher values of `youtube` . This incapacitance (inability to capture intricate fluctuations) to perform well in higher values of feature makes it not an ideal fit for the data.

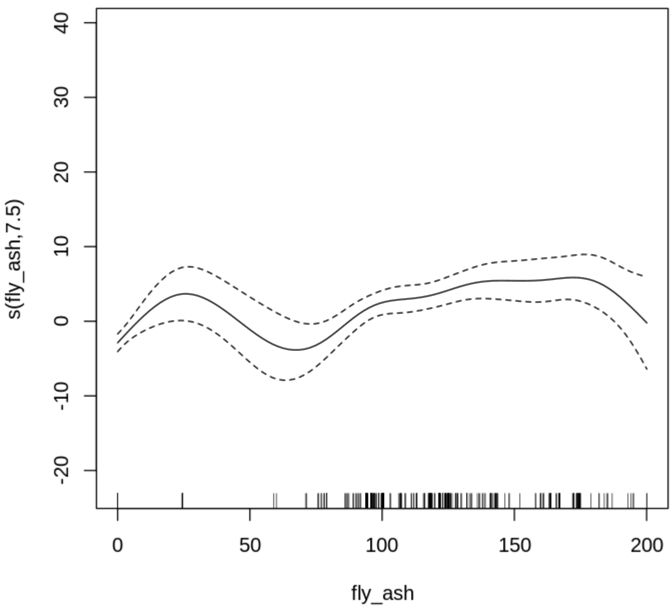
In [58]:

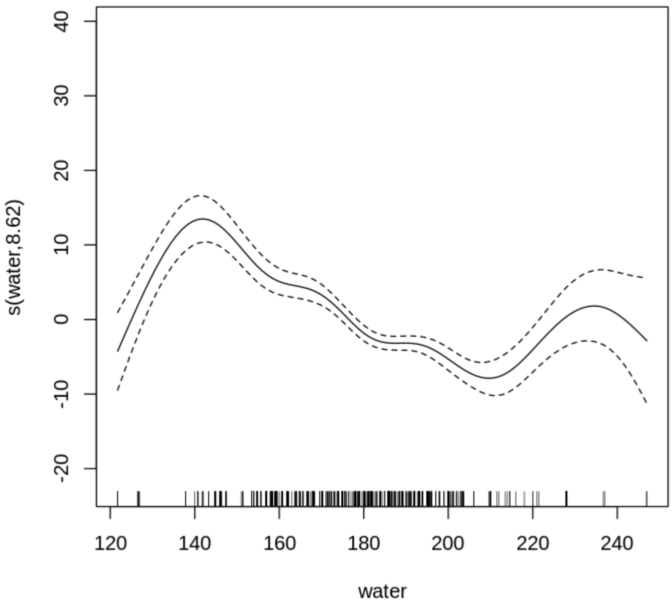
```
library(mgcv)

gammod = gam(strength ~ s(cement) + s(blast_furnace) + s(fly_ash) + s(water) + s(plast) + s(coarse) + s(fine) + s(age), data = train)

# cement + blast_furnace + fly_ash + water + plast + coarse + fine + age

res = residuals(gammod, type="deviance") #compute the deviance residuals
plot.gam(gammod)
```

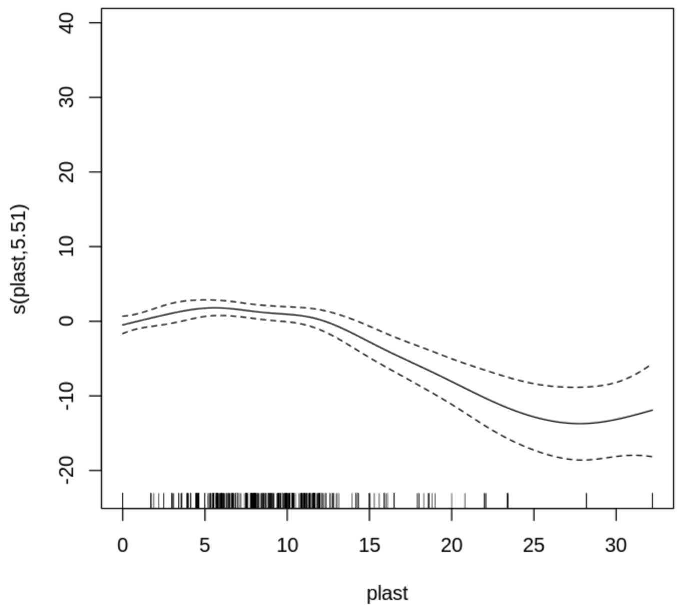
```
Loading required package: nlme

This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

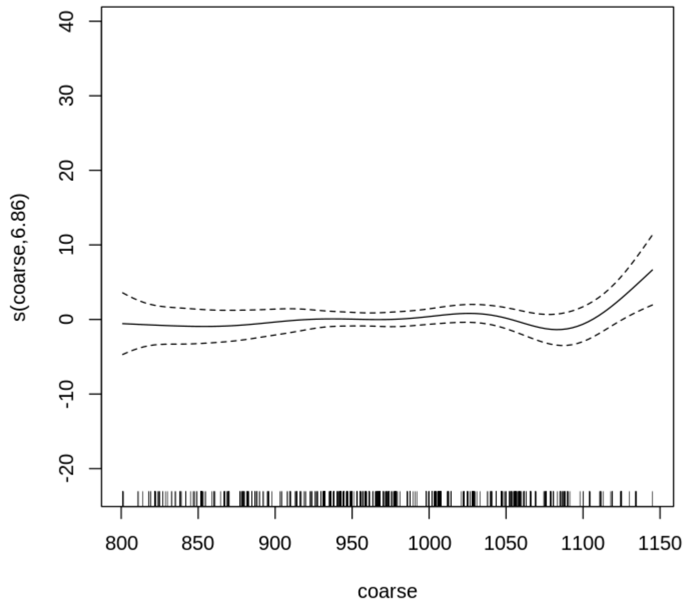


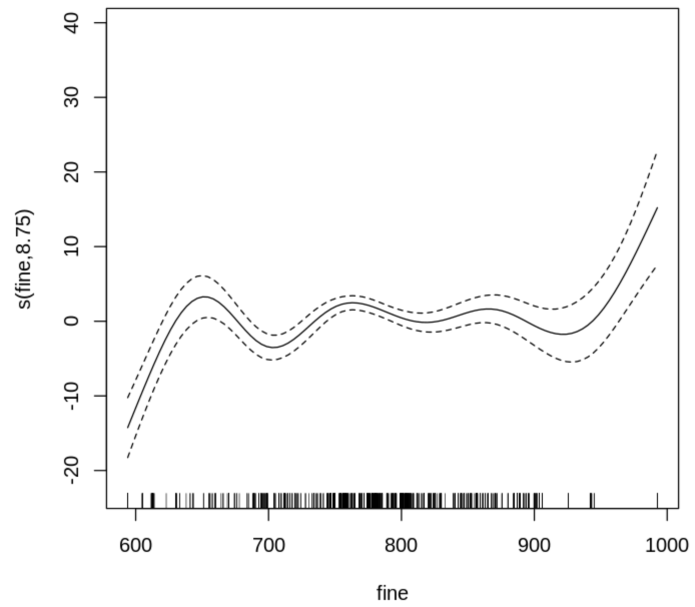


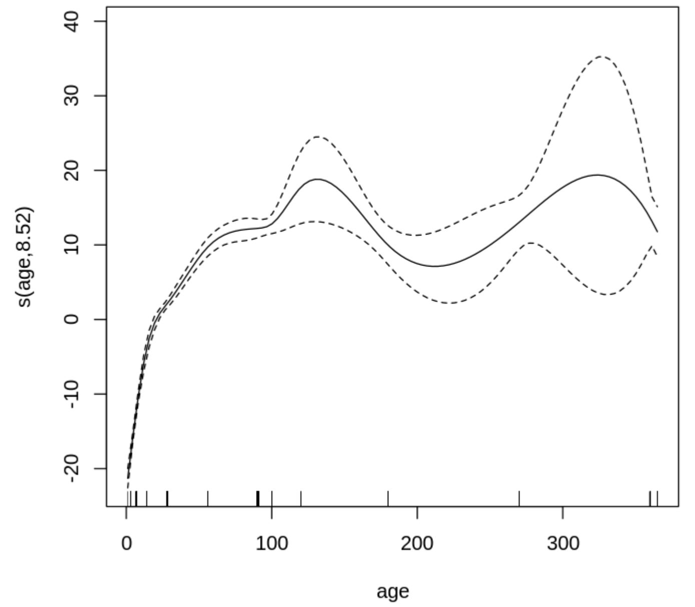

s(fly_ash,7.5)
40
30
20
10
0
-10
-20
0
50
100
150
200
fly_ash


s(water,8.62)
40
30
20
10
0
-10
-20
120
140
160
180
200
220
240
water


s(plast,5.51)
40
30
20
10
0
-10
-20
0
5
10
15
20
25
30
plast


s(coarse,6.86)
40
30
20
10
0
-10
-20
800
850
900
950
1000
1050
1100
1150
coarse





The above plots show the relationship between each predictor and response variable. If we observe dotted lines, which indicates band of confidence intervals, we are not able to draw a straight line between upper and lower curves for any of our predictors. This is an evidence to say that there exists a non-linear relationship. The performance of GAM will be obviously better compared to any of the parametric model because there is no limitation by any particular function, have more flexibility in determining associations. However, it also comes with some cons that it is a bit trickier to formulaically describe the relationship.

```
In [59]: summary(gammod)
```

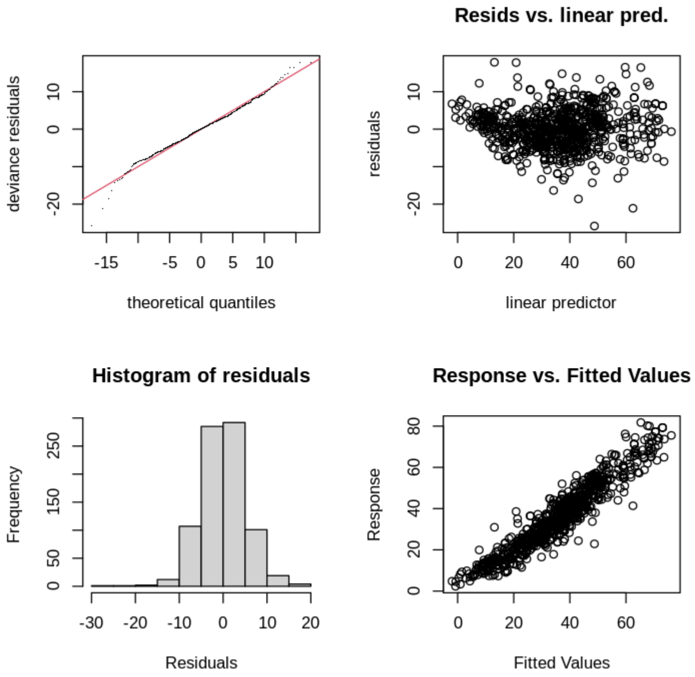
```
Family: gaussian
Link function: identity

Formula:
strength ~ s(cement) + s(blast_furnace) + s(fly_ash) + s(water) +
    s(plast) + s(coarse) + s(fine) + s(age)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   35.793      0.187   191.4   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                   edf Ref.df       F p-value
s(cement)        8.220  8.823  41.173  <2e-16 ***
s(blast_furnace) 7.558  8.429  22.320  <2e-16 ***
s(fly_ash)       7.498  8.359   6.307  <2e-16 ***
s(water)         8.621  8.945  19.313  <2e-16 ***
s(plast)         5.507  6.583  10.291  <2e-16 ***
s(coarse)        6.857  7.937   1.714     0.1
s(fine)          8.750  8.973  17.104  <2e-16 ***
s(age)           8.519  8.886 291.881  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.892   Deviance explained =   90%
GCV = 31.184  Scale est. = 28.818    n = 824
```

The adjusted R^2 has improved significantly with same number of predictors which we used in our linear regression. So, GAM is explaining 86.2\% proportion of variance in predicting output variable. The amount of deviance explained is quite good which further supports the model's goodness of fit

In [60]:

```
gam.check(gammod)
```

```
Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 22 iterations.
The RMS GCV score gradient at convergence was 7.761442e-07 .
The Hessian was positive definite.
Model rank =  73 / 73 

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

                   k'  edf k-index p-value    
s(cement)        9.00 8.22    0.90  <2e-16 ***
s(blast_furnace) 9.00 7.56    0.91   0.010 ** 
s(fly_ash)       9.00 7.50    0.90  <2e-16 ***
s(water)         9.00 8.62    0.86  <2e-16 ***
s(plast)         9.00 5.51    0.91   0.005 ** 
s(coarse)        9.00 6.86    0.84  <2e-16 ***
s(fine)          9.00 8.75    0.82  <2e-16 ***
s(age)           9.00 8.52    0.98   0.220    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



If we observe residuals Vs linear predictor plot, there is no funnel or trumpet or horn shape, it indcates that GAM has done a good fit of the data. This indicates that structure of the model doen't have issues. Also, upon examining QQ plot, residulas very well stick to the straight line, it is a good indicator of model performance. Finally, in fitted Vs true values plot, we can see a near straight line indicating that model's predictions are more or less fitting well.

Now, let us see how model will perform if we remove one feature. Let's try with a reduced GAM and perform ANOVA to see the comparison.

Null Hypothesis: H_0 $=$ Reduced model is sufficient
Alternative Hypothesis: H_1 $=$ Reduced model is not sufficient

In [61]:
```
gammod_red = gam(strength ~ s(fly_ash) + s(blast_furnace) + s(water) + s(pl
ast) + s(coarse) + s(fine) + s(age) , data = train)

res = residuals(gammod_red, type="deviance")
summary(gammod_red)
```

```
Family: gaussian
Link function: identity

Formula:
strength ~ s(fly_ash) + s(blast_furnace) + s(water) + s(plast) + 
    s(coarse) + s(fine) + s(age)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  35.7933     0.2216   161.5   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                   edf Ref.df      F p-value    
s(fly_ash)       8.148  8.765  54.38  <2e-16 ***
s(blast_furnace) 8.199  8.802  11.16  <2e-16 ***
s(water)         8.714  8.970  77.80  <2e-16 ***
s(plast)         7.195  8.192  15.35  <2e-16 ***
s(coarse)        8.128  8.784  51.54  <2e-16 ***
s(fine)          8.721  8.971  61.61  <2e-16 ***
s(age)           8.651  8.937 202.40  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.848   Deviance explained = 85.9%
GCV = 43.563  Scale est. = 40.457    n = 824
```

In [62]:
```
anova(gammod_red, gammod)
```

A anova: 2 × 6

	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	761.5800	30959.13	NA	NA	NA	NA
2	756.0635	21943.92	5.516431	9015.217	56.70953	3.502309e-54

Based on above ANOVA output on F-statistic and p-val, it suggests that we reject null hypothesis. There is no evidence to say that reduced model is sufficient. That means we can conclude that full model is necessary for predicting varaiability in response variable.

In [63]:

```
gammod_red1 = gam(strength ~ s(cement) + s(blast_furnace) + s(water) + s(pl
ast) + s(coarse) + s(fine) + s(fly_ash) , data = train)

res = residuals(gammod_red1, type="deviance")
summary(gammod_red1)
```

```
Family: gaussian
Link function: identity

Formula:
strength ~ s(cement) + s(blast_furnace) + s(water) + s(plast) +
    s(coarse) + s(fine) + s(fly_ash)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.7933     0.3914   91.45   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                   edf Ref.df      F  p-value
s(cement)        3.933  4.890 19.033  < 2e-16 ***
s(blast_furnace) 1.972  2.436 15.018  < 2e-16 ***
s(water)         8.564  8.931 12.439  < 2e-16 ***
s(plast)         3.156  3.911  7.222 1.61e-05 ***
s(coarse)        1.000  1.000  0.089    0.766
s(fine)          8.002  8.717  4.479 1.37e-05 ***
s(fly_ash)       1.003  1.006  2.615    0.106
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.527   Deviance explained = 54.3%
GCV = 130.77  Scale est. = 126.23    n = 824
```

In [64]:

```
anova(gammod_red1, gammod)
```

A anova: 2 × 6

	Resid. Df	Resid. Dev	Df	Deviance	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	792.1089	100398.94	NA	NA	NA	NA
2	756.0635	21943.92	36.04532	78455.02	75.52832	1.235588e-223

Based on second ANOVA output on F-statistic and p-val, it suggests that we reject null hypothesis. There is no evidence to say that reduced model is sufficient. That means we can conclude that full model is necessary for predicting varaiability in response variable.

Let's check MSPE values on test data to compare different models suggested by backward selection, AIC/BIC/R^2 and GAM.

In [65]:

```
pred_gam <- predict(gammod, newdata = test, type = "response")
MSPE_gam <- mean((test$strength - pred_gam)^2)

pred_gam_r <- predict(gammod_red, newdata = test, type = "response")
MSPE_gam_r <- mean((test$strength - pred_gam_r)^2)

pred_gam1 <- predict(gammod_red1, newdata = test, type = "response")
MSPE_gam1 <- mean((test$strength - pred_gam1)^2)


cat("MSPE for GAM full:", MSPE_gam, "\n")
cat("MSPE for GAM without cement:", MSPE_gam_r, "\n")
cat("MSPE for GAM without age:", MSPE_gam1, "\n")
```

```
MSPE for GAM full: 33.17473
MSPE for GAM without cement: 45.65184
MSPE for GAM without age: 137.0106
```

The Backward MLR model got a score of 107.13.

Another model selected by AIC, BIC, R_a^2 got a score of 108.5.

The GAM-Full model got the best score of 33.17, meaning it was the most accurate.

There were also two other models which I tried in GAM excluding some features.The one without cement got a score of 45.65, and the other got a score of 137.01.

These scores help us see which model does the best job. As we can see that MSPE is quite low for GAM-Full compared to GAM without cement as predictor and GAM without age as predictor, we can say that GAM-Full has outperformed remaining models.

Report

Background:

I am interested in this analysis because understanding concrete strength is relevant in the context of infrastructure resilience and disaster preparedness. I am originally from a city in India which is prone to natural disasters such as earthquakes. So, I believe any knowledge about durable concrete structures can contribute to community resilience and disaster mitigation efforts. Also, I am naturally curious about how everyday materials and structures work. Reading more on these topics like concrete compressive strength can satisfy this curiosity and can help in developing a deeper appreciation for the science and engineering behind common materials we encounter in our daily lives.

This dataset is about various composition materials which can be mixed up to form concrete. Concrete is a vastly used construction material composed of cement, aggregates (such as sand and gravel), water, and sometimes additional materials like admixtures or additives. It's popular because it's strong, lasts for long time. Infact, this dataset was the outcome of **laboratory experiment** where compositions of these components were varied across range of values and all such concrete bricks were tested for compressive strength as shown in image in below cell.

Experiment Methodology

Compressive strength refers to the ability of a certain material or structural element to withstand loads that reduce the size of that material, or structural element, when applied. A force is applied to the top and bottom of a test sample, until the sample fractures or is deformed. Compressive strength is typically measured using standardized tests such as the cylinder compression test or cube compression test. These tests involve subjecting concrete samples to increasing compressive loads until failure occurs, and then calculating the maximum load-bearing capacity, which is nothing but compressive strength.

This data was collected by Professor I.Cheng Yeh in 1998 and published in journal titled, "Modeling of strength of high-performance concrete using artificial neural networks". This project was done in collaboration of Engineering, Materials Science, Computer Science Cement and Concrete Research. A more in-depth research about this topic can be found here (https://www.sciencedirect.com/science/article/abs/pii/S0008884698001653?via%3Dihub) . To understand the test in a better way, this video (https://www.youtube.com/watch?v=iYmil0luMEs) can be looked at. I would like to explore relationship between different materials in concrete mixtures (such as cement, aggregates, and water) and strength. Also, I will look at how age of concrete, or curing time, impacts strength. I wanted to test the hypothesis that *The rate of gain of concrete compressive strength is relatively high during the first 28 days of casting and then it slows down*. i.e., Strength of concrete above 28 days is much more high compared to those with age below 3 days. I also aim to develop and evaluate various predictive models to determine concrete strength based on its composition and comparing performance of these models with metrics like AIC, BIC, MSPE, R_a^2 with aid of visual examination of diagnostic plots.

Methods and Results

The data is obtained from UCI Machine Learning repository link text (https://archive.ics.uci.edu/dataset/165/concrete+compressive+strength). The methodology of how experiment is conducted is already desribed above.

1. ***Exploratory Data Analysis***: Colunms were renamed for simplicity maintaining the meaning the original name is conveying. There are no missing values in this dataset. However, there are few outliers present in some features according to boxplot, however, I feel these are not real outliers in context of experiment

because experiment was intentionally conducted at varying values (extreme low and extreme high) to understand how strength is behaving. So, these are not real outliers and need not be removed.

2. ***Analyses***:

   Hypothesis testing: It is carried out using t-statistic to test the validity of general accepted notion on range of strength values developed over time. Also, second hypothesis test is carried out to check whether addition of blast furnace slag into concrete has improved strength or not. **F-test**: To assess the usefulness of at least one predictor, I conducted a full F-test.
   MLR: I have also conducted Multiple Linear regression and selected features by following backward selection process. Iterations were carried out till my model has all significantly contributing predictors in it. It has taken me around 2-3 iterations to arrive at best model.
   Diagnostics: I have plotted various diagnostic plots on residuals and fitted values to see if there are any structural issues in my linear model. By identifying the deviations in assumptions of linear regession, we will be able to determine whether the fitted model is a proper fit or not. Diagnostic plots for my model revealed that existing model is not a good fit for this dataset and this can be seen in slighter curvature observed in fitted Vs residuals plot. Also, observed Vs fitted plot helps in determining whether our model is overpredicting or underpredicting.
   Model Selection using AIC, BIC, R_a^2: At each size of k, all combinations were tried out and best model at each size is returned by `regsubsets()` function. This is done so as to get best combination of features and then compare its performance based on metrics of AIC, BIC and R_a^2.
   General Additive Modelling: After going through detailed plots in section `Digging Deep` , necessity is felt to restructure my model to non-linear regression. If there exists a simple quadratic or cubic, I would have tried to do polynomial regresion, however, relationship between strength and age kept on varying at various levels of age values. This motivated me to consider GAM to determine complex relationship. Diagnostic plots were made after fitting data through GAM, and they all looked much better in terms of fitted Vs observed values error.

3. **Interpretation of results**:

   - **Hypothesis testing**: Below is the snapshot of result. From this, we can infer that we reject null hypothesis and there is statistically significant difference between means of strength in two age groups:
     - **Null Hypothesis**:$H_0 =$. Both means (mean > 28d, mean < 3d) are equal and not significantly different from each other.
     - **Alternate Hypothesis**: $H_a =$ Both means (mean > 28d, mean < 3d) are not equal and difference in means has not occured by chance and are significantly different from each other

```
Snippet of Hypothesis testing output:
  Welch Two Sample t-test

  data:  age_28_data$strength and age_5_data$strength
  t = 22.073, df = 278.76, p-value < 2.2e-16
  alternative hypothesis: true difference in means is not equal to 0
  95 percent confidence interval:
  20.59446 24.62747
  sample estimates:
  mean of x mean of y
  41.45192  18.84096
```

- **Hypothesis testing**: To test whether addition of slag has any imoact on compressive strength of concrete. Below is the snapshot of result. From this, we can infer that we reject null hypothesis and there is statistically significant difference between means of strength in two groups:
  - **Null Hypothesis**:$H_0 =$. There is no impact of slag on strength
  - **Alternate Hypothesis**: $H_a =$ Slag addition has impacted strength of concrete

```
Welch Two Sample t-test
data:  bf_0$strength and bf_not_0$strength
t = -6.1519, df = 96.506, p-value = 8.682e-09
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -6.734656
sample estimates:
mean of x mean of y
21.49030  30.71543
```

- **F-test**: Below is the snapshot of result. From this, we can infer that we reject null hypothesis and there is statistically significant difference between means of strength in two age groups:
  - **Null Hypothesis**:$H_0 =$. Reduced model is sufficient. None of the predictors is useful in predicting output
  - **Alternate Hypothesis**: $H_a =$ Atleast one among the predictors i.e., $\beta_j \neq 0$. Atleast one of these predictors is significant in predicting response

```
        Snippet of F-test output:
                 Res.Df    RSS Df  Sum of Sq   F   Pr(>F)
                 <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
                 1 1029    287173.0    NA  NA  NA  NA
                 2 1021    110428.2    8   176744.9    204.2691    6.76
    1578e-206
```

- **Choosing MLR model by backward selection**:

  Based on backward selection, the best model chosen is - $strength = 29.75 + 0.102\times$ `cement` $+0.086\times$ `blast_furnace` $+0.069\times$ `fly_ash` $-0.216\times$ `water` $+0.224\times$ `plast` $+0.112\times$ `age` .

  Keeping all other variables fixed, 1 unit increment of cement will cause increment in strength by 0.1 times. Likwise, 1 unit addition of balst furnace slag will help in increasing compressive strength by 0.086 with all other features constant.

- **Diagnostics**: Diagnosis of linear regression model revealed that model has some structural issues and doesn't reflect good fitment. Also, these patterns revealed that there are no influential points, satisfied assumption of normality, deviation from homoscedasticity and non-linearity. The polts are shown above in `Diagnostics` section. The ideal condition is meeting all assumptions of regression, however, this is not the case in this analysis.
- **Using AIC, BIC, R_a^2**: Interestingly, all three metrics have suggested same model with 5 predictors. `strength` = $35.04 + 0.106\times$ `cement` $+0.091\times$ `blast_furnace` $+0.078\times$ `fly_ash` -0.249 `water` $+0.113$ `age`
- **GAM**: Based on comparison of two models (reduced model without cement Vs full model with cement) using ANOVA, the output suggests that full model with cement as predictor is necessary in struture of GAM. We can mathematically write it as - `strength = s(cement)  s(blast_furnace) + s(fly_ash) + s(water) + s(plast) + s(coarse) + s(fine) + s(age)`

- **Comparison of Model**: Below is the table showing MSPE values values acquired from various models. Full GAM stood as the best one among all others.

Model	MSPE
Backward Selection - MLR	107.13
AIC, BIC, R_a^2 - Linear	108.5
GAM-Full	33.17
Reduced GAM excluding cement	45.65
Reduced GAM excluding age	137.01

1. **Conclusion**:

- In estimating compressive strength of concrete, predictor `age` is predominant. Its significance can be understood by looking at substantial impact observed when it is omitted from the model. Removal of this predictor results in a marked decrease in both deviance explained and the adjusted coefficient of determination. When it is removed from model, deviance explained and R_a^2 has reduced sharply from ~84\% to ~55\%.
- Non-linear association exists between `compressive strength` and predictors.
- Concrete strength after period of 28 days significantly surpasses that of samples aged less than 3 days.
- Addition of blast furnace slag to cement impacts the strength by enhancing it.

My learning:

- Generally, it is not that easy to identify non-linear association between response and predictors. We need to look from every angle to diagnose such interpretations.
- Whatever, is the type of problem, GAM/GLMs generally outperforms all other models.
- Potential exclusion of outliers even though they are not real outliers is something which needs domain knowledge and a level of expertise. Also, we need to understand how dataset is made and in what context observations were made.

Below is the image of experimental setup of this activity.

```
In [73]: install.packages("jpeg")
         library(jpeg)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
In [68]: img <- readJPEG("/content/compressive.jpg")
         plot(1:2, type = "n", ann = FALSE, axes = FALSE)
         rasterImage(img, 1, 1, 2, 2)
```