Andrew Liu 525 Final Project

2023-11-09

Loads necessary libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(leaps)
library(caTools)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
```

```
##     some
```

```
library(ggplot2)
```

Part I: Introduction, Motivation, and Analysis Goal

Half of all American adults report delaying medical treatment due to price concerns, and approximately a quarter of US households within the last year alone have had difficulty paying for healthcare costs. Finally, over 60% of all bankruptcies filed in America are medical in nature, due to either high healthcare costs or lost productivity while infirm.

Given these facts, healthcare affordability is a highly relevant topic in our country. In this analysis, we examine a CMS (Centers for Medicare & Medicaid Services) dataset of personal medical expenditures broken down by type of healthcare service, age, sex, and funding source for even years between 2002 and 2020. Our analysis goal is to construct, diagnose, remedy, and assess the accuracy of a multiple linear regression model that predicts personal healthcare expenditures of non-out-of-pocket payers in 2020.

Part II: Data Cleaning

We first read our dataset into R.

```
expenditures <- read.csv("age and Sex.csv")

head(expenditures)
```

```
##      Payer         Service Age.Group   Sex X2002 X2004 X2006 X2008 X2010 X2012
## 1 Medicaid Dental Services     Total Total  3709  4264  4791  6246  8392  8493
## 2 Medicaid Dental Services      0-18 Total  2085  2591  2938  3737  5012  4989
## 3 Medicaid Dental Services     19-44 Total   937   984  1077  1405  1929  1938
## 4 Medicaid Dental Services     45-64 Total   433   440   515   733   974  1071
## 5 Medicaid Dental Services     65-84 Total   207   202   211   291   358   369
## 6 Medicaid Dental Services       85+ Total    48    47    51    80   118   127
##   X2014 X2016 X2018 X2020
## 1  9773 11889 12644 12637
## 2  5216  6195  6340  6246
## 3  2480  3029  3270  3315
## 4  1543  1952  2088  2117
## 5   400   543   728   738
## 6   135   169   218   221
```

Before we begin any type of analysis, we must first clean our data. We will start by removing rows with "Total" entries in the *Age Group, Sex, Payer,* and *Service* categories because entries are not unique observations. Rather, these observations contain the total sum of spending for their respective category. Removing these observations is necessary to ensure independence.

```
expenditures <- filter(expenditures, Age.Group != "Total" & Sex != "Total"
                       & Payer != "Total")

expenditures <- filter(expenditures, Service != "Total Personal Health Care")
```

We will drop any observation with "Out-of-Pocket" in the *Payer* category because the personal healthcare costs are drastically higher than those of any other category; including these observations will unduly skew our model fit. After we do so, we can drop the *Payer* category because we aren't interested in separating individuals by source of insurance.

```
expenditures <- filter(expenditures, Payer != "Out-of-Pocket")
```

```
expenditures <- expenditures[-1]
```

To simplify our analysis, we restrict the observations to Prescription Drugs, Home Health Care, and Dental Services in the *Service* category. We ignore the other entries in the category because their descriptors are either too vague (ie, Other Health Residential and Personal Care and Other Nondurable Medical Products) or because their corresponding costs skew too heavily in the 65-84 and +85 age ranges (ie, Nursing Care Facilities and Continuing Care Retirement Communities).

```
expenditures <- filter(expenditures, Service %in% c("Prescription Drugs",
                                                    "Home Health Care", "Dental Services"))
```

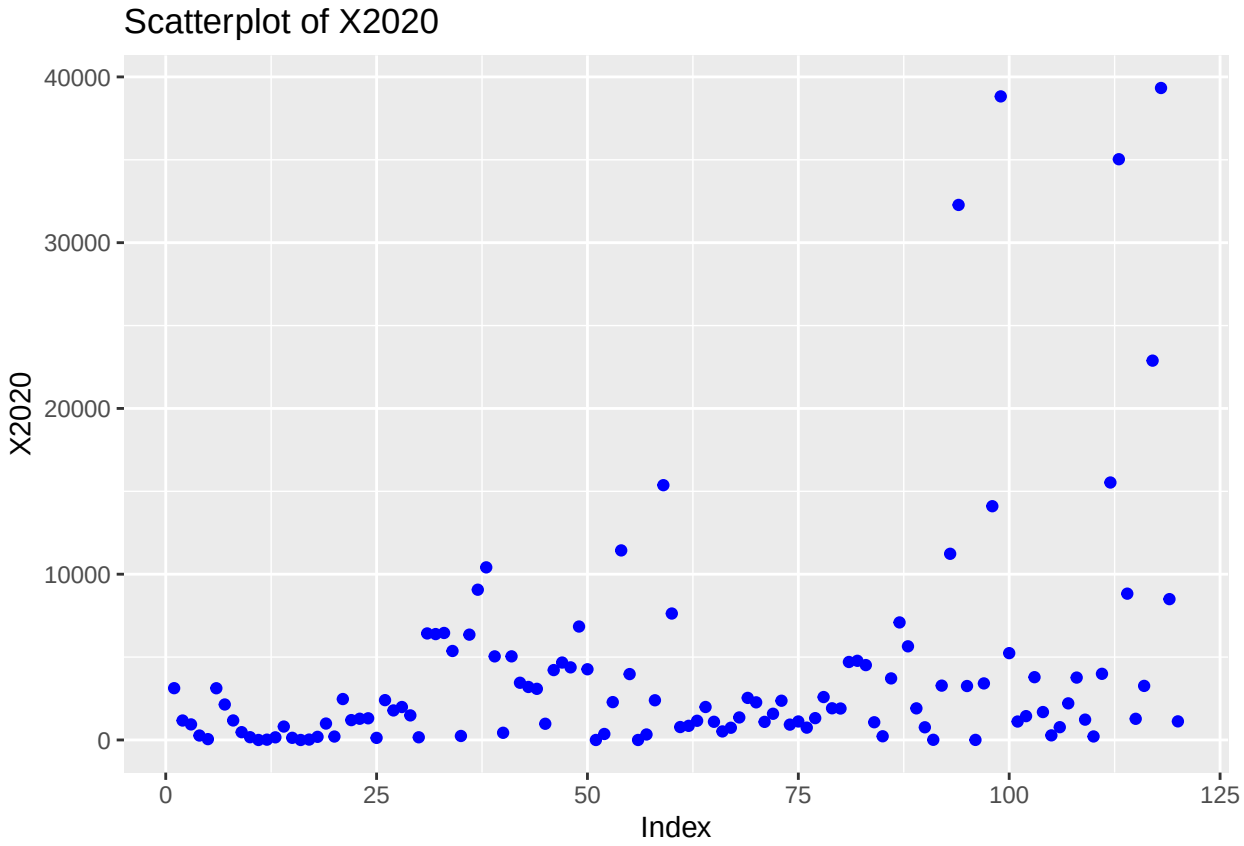
We will further restrict the scope of our analysis to data collected between 2010 until 2020. This is due to the fact that the Affordable Care Act (ACA) was signed into law in 2010. This was a landmark piece of legislation that fundamentally altered the American healthcare system for the first time since the establishment of Medicare and Medicaid 1965. As a result, cost data collected pre-ACA and post-ACA are likely so different such that the former would not aid us in predicting the latter.

```
drop <- c("X2002", "X2004", "X2006", "X2008")
expenditures <- expenditures[,!(names(expenditures) %in% drop)]

summary(expenditures)
```

```
##    Service           Age.Group             Sex                X2010
##  Length:120         Length:120         Length:120         Min.   :    0.0
##  Class :character   Class :character   Class :character   1st Qu.:  237.2
##  Mode  :character   Mode  :character   Mode  :character   Median :  810.5
##                                                           Mean   : 2758.6
##                                                           3rd Qu.: 3056.8
##                                                           Max.   :34911.0
##      X2012             X2014           X2016             X2018
##  Min.   :    0.0   Min.   :    0   Min.   :    0.0   Min.   :    0.0
##  1st Qu.:  274.5   1st Qu.:  285   1st Qu.:  298.8   1st Qu.:  389.8
##  Median :  927.0   Median : 1067   Median : 1268.5   Median : 1380.0
##  Mean   : 2878.1   Mean   : 3214   Mean   : 3507.2   Mean   : 3754.3
##  3rd Qu.: 3324.0   3rd Qu.: 3633   3rd Qu.: 4137.2   3rd Qu.: 4058.2
##  Max.   :32003.0   Max.   :35917   Max.   :39442.0   Max.   :38110.0
##      X2020
##  Min.   :    0
##  1st Qu.:  778
##  Median : 1910
##  Mean   : 4172
##  3rd Qu.: 4415
##  Max.   :39332
```

Let us examine the distribution of our response variable *X2020,* or personal healthcare expenditure in 2020.

```
ggplot(expenditures, aes(x = seq_along(X2020), y = X2020)) +
  geom_point(color = "blue") +
  labs(x = "Index", y = "X2020", title = "Scatterplot of X2020")
```



It is evident that our response takes values across multiple orders of magnitude. We could bypass this issue by standardizing all our numbers, but this would come at the expense of both interpretability and our goal of predict cost. Since linear regression works best when variables are all on a similar scale

Since linear regression works best when variables are all on roughly similar scales, we will remove entries where the expenditure in 2020 were below some amount, say under 100 USD, and above some amount, say 10000 USD. The vast majority of expenditures are in the thousands range and we can improve the efficacy of our regression model by taking this pre-emptive step to reduce the number of outliers.

```
expenditures <- filter(expenditures, X2020 >= 100)
expenditures <- filter(expenditures, X2020 <= 10000)

summary(expenditures)
```

```
##    Service           Age.Group            Sex                X2010
##  Length:100         Length:100         Length:100         Min.   :  12.0
##  Class :character   Class :character   Class :character   1st Qu.: 258.2
##  Mode  :character   Mode  :character   Mode  :character   Median : 806.5
##                                                           Mean   :1682.5
##                                                           3rd Qu.:2574.5
##                                                           Max.   :9062.0
##      X2012             X2014            X2016          X2018
##  Min.   :   9.0   Min.   :  11.0   Min.   :  11   Min.   :  24
##  1st Qu.: 307.5   1st Qu.: 307.8   1st Qu.: 358   1st Qu.: 447
##  Median : 883.5   Median :1029.5   Median :1234   Median :1316
##  Mean   :1768.4   Mean   :1928.7   Mean   :2084   Mean   :2252
##  3rd Qu.:2847.0   3rd Qu.:3145.0   3rd Qu.:3316   3rd Qu.:3436
##  Max.   :8161.0   Max.   :8258.0   Max.   :8662   Max.   :9381
##      X2020
```

```
##  Min.   : 127.0
##  1st Qu.: 936.8
##  Median :1903.5
##  Mean   :2540.9
##  3rd Qu.:3770.0
##  Max.   :9065.0
```

We began with a messy dataset of over 1100 entries and 14 columns, but after trimming away all the unnecessary elements for our analysis, we end up with a relatively clean 100 entries and 9 columns. Our next step is to code categorical variables for the predictors that require them.

Part III: Handling Categorical Predictors

There are 8 potential predictors, of which 5 are quantitative (personal spending amount for even years starting from 2010 until 2018) and 3 are categorical (*Service, Age.Group,* and *Sex*). There are two possible ways to handle these predictors: assigning codes or implementing indicator variables. A predictor with *j* levels requires *j-1* indicators, so coding indicator variables for a predictor with many levels will significantly drop the MSE degrees of freedom as additional parameters are introduced in the model. We can avoid the tradeoffs associated with implementing so many dummy variables by assigning codes to each level of the predictor instead. However, this method implies a potentially false relationship between the mean response and the changing levels of the predictor. Both methods clearly have their advantages and disadvantages, and are likewise appropriate for different types of categorical variables.

Let's start with *Age.Group.* We have five levels for *Age.Group* because the CMS partitions age into 5 categories: 0-18, 19-44, 45-64, 65-84, and 85+. Coding four new indicator variables for this predictor would be cumbersome, and we would lose degrees of freedom, so we will assign codes to each level instead. The CMS data indicates that personal healthcare expenditures generally rise with age, peaking in the 45-64 age range before drastically decreasing. This drop in spending may be explained by Medicare coverage taking effect when people turn 65, thereby allaying the need to personally spend. A notable exception to this trend are Dental Services, which are typically the highest for the 0-18 age range, but we would expect this as this is the age range of significant dental development.

We will use the following assignments to capture this relationship between age range and cost.

```
expenditures_reg <- expenditures %>%
  mutate(Age.Group = case_when(
    Age.Group == "0-18" ~ 3,
    Age.Group == "19-44" ~ 4,
    Age.Group == "45-64" ~ 5,
    Age.Group == "65-84" ~ 1,
    Age.Group == "85+" ~ 2,
  ))
```

Now onto *Sex.* We will treat Males as the reference level for the indicator variable.

```
expenditures_reg <- expenditures_reg %>%
  mutate(Sex = case_when(Sex == "Females" ~ 1, TRUE ~ 0))
```

By our work done in Part 1, we know *Service* has 3 levels: Prescription Drugs, Home Health Care, and Dental Services. We will treat Dental Services as the reference level.

```
expenditures_reg <- expenditures_reg %>%
  mutate(
    Is_Home = case_when(Service == "Home Health Care" ~ 1, TRUE ~ 0),
    Is_Drug = case_when(Service == "Prescription Drugs" ~ 1, TRUE ~ 0)
  )
```

We can now remove the *Service* column because all its information is already encoded in our new indicator

variables.

```
drop1 <- c("Service")
expenditures_reg <- expenditures_reg[,!(names(expenditures_reg) %in% drop1)]
```

We may now proceed to investigate our data for any issues surrounding multicollinearity and transform our response variable, if appropriate.

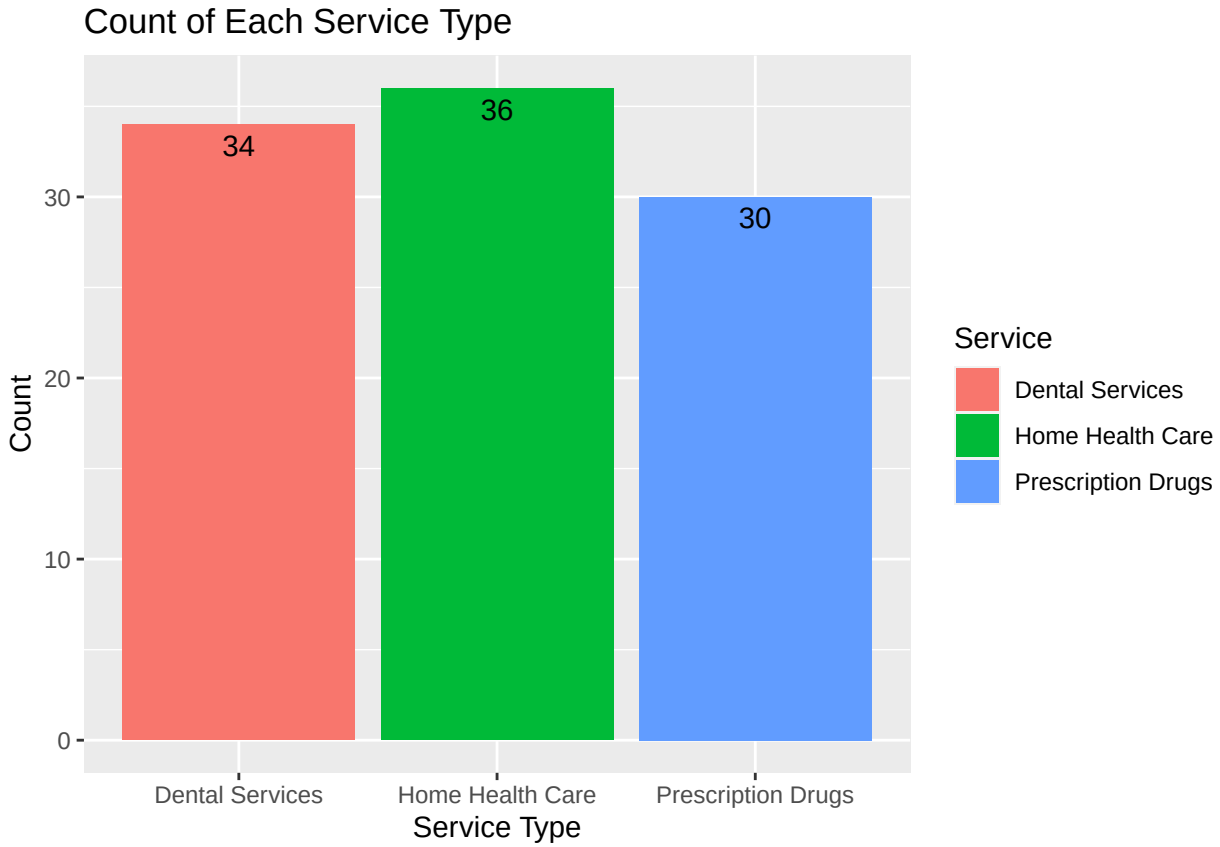
Part IV: Investigating Variables, Verifying Assumptions, and Spotting Unusual Observations

We have 9 predictors, but it is almost guaranteed that only a small subset of them will be statistically significant or useful for predicting personal healthcare expenditures in 2020. Our final or "best" model will only contain a few of these predictors.

Let us check that observations of each level in our categorical variables are equal, or at least roughly so. This step is essential for ensuring that we have a balanced dataset.

We'll start with *Service* and its three levels: Prescription Drugs, Home Health Care, and Dental Services.

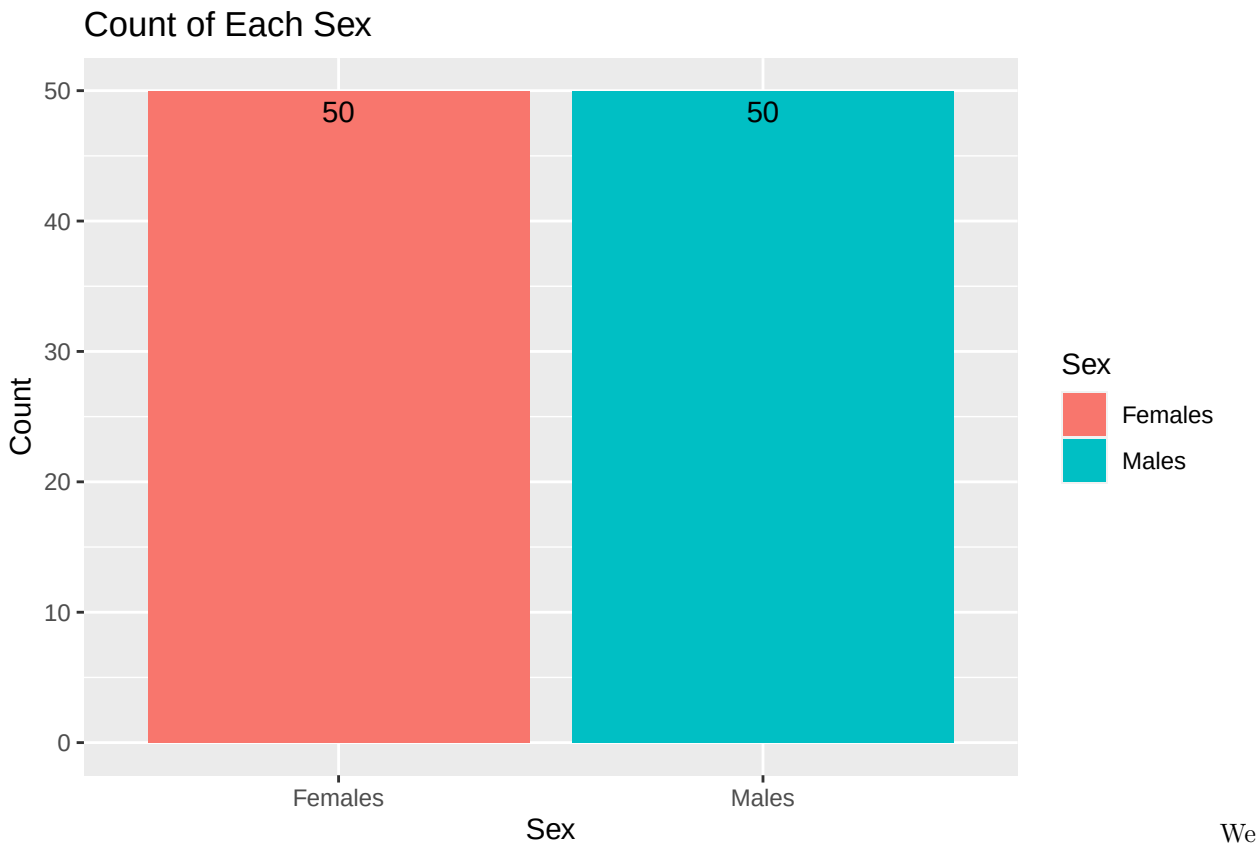
```
service_counts <- expenditures %>%
  count(Service)

ggplot(service_counts, aes(x = Service, y = n, fill = Service)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust = 1.5, position = position_dodge(width = 0.9))   + labs(title = "Coun
```



The numbers of observations for each level of *Service* are not too dissimilar.

Onto *Sex*:

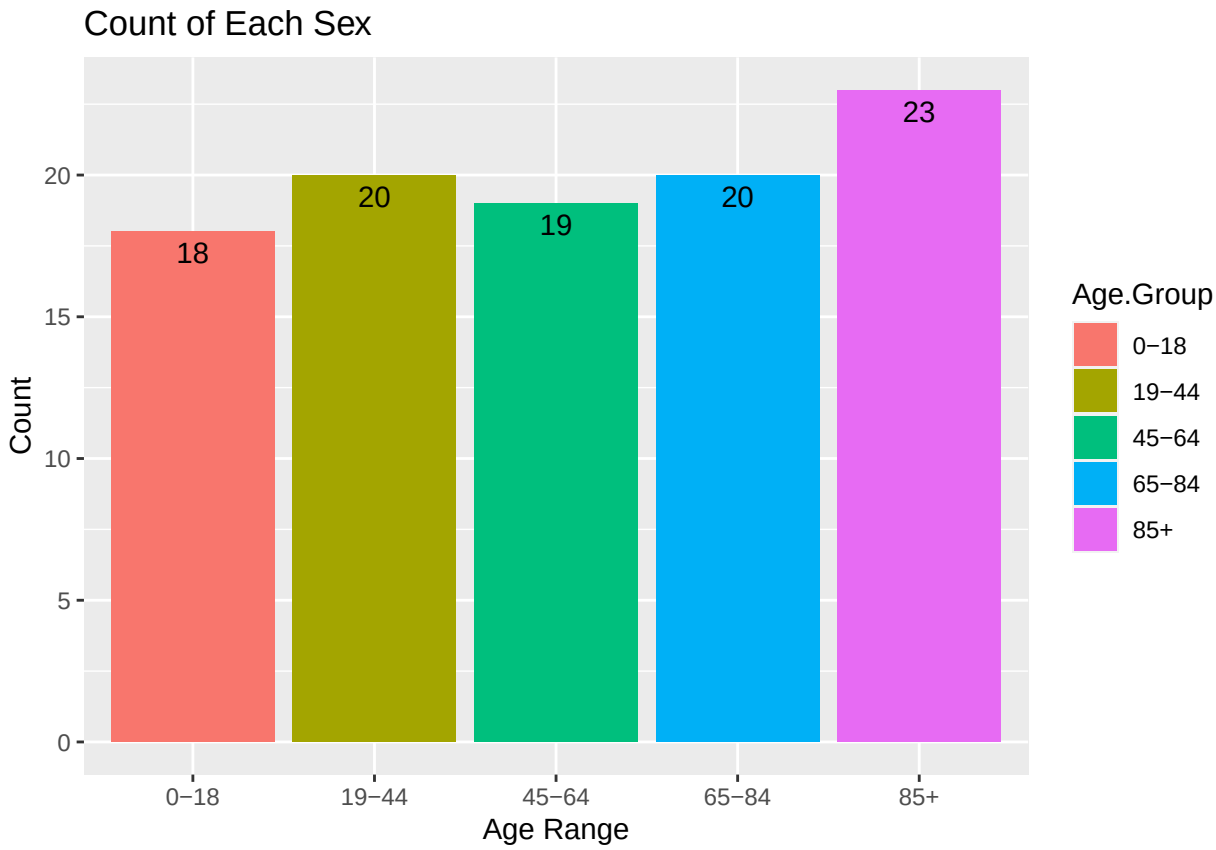
```r
sex_counts <- expenditures %>%
  count(Sex)

ggplot(sex_counts, aes(x = Sex, y = n, fill = Sex)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust = 1.5, position = position_dodge(width = 0.9))   + labs(title = "Coun
```



We have the same number of males and females.

Finally onto *Age.Group*:

```r
age_counts <- expenditures %>%
  count(Age.Group)

ggplot(age_counts, aes(x = Age.Group, y = n, fill = Age.Group)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust = 1.5, position = position_dodge(width = 0.9))   + labs(title = "Coun
```



We have roughly equal counts for each age range.

Hence we may conclude we have a balanced dataset with respect to the levels of each categorical variable.

Now we will investigate our quantitative variables. Our intuition is that we will find high pairwise collinearity among the expenditures by year. To verify this claim, we will first construct a scatterplot matrix to examine the bivariate relationships between our quantitative variables. The upper triangle of the scatterplot matrix is shown below.

```
expenditures_reg_years <- expenditures_reg[3:8]

ggpairs(expenditures_reg_years, upper = list(continuous = "points"), lower = list(continuous = "blank")
```

```
## Warning in check_and_set_ggpairs_defaults("diag", diag, continuous =
## "densityDiag", : Changing diag$continuous from 'blank' to 'blankDiag'
```

One feature that stands out in our scatterplot matrix is how observations generally become more scarce as expenditure increases. This gap reveals the presence of several highly-influential observations. We would need to examine their residuals to properly assess whether they are outliers.

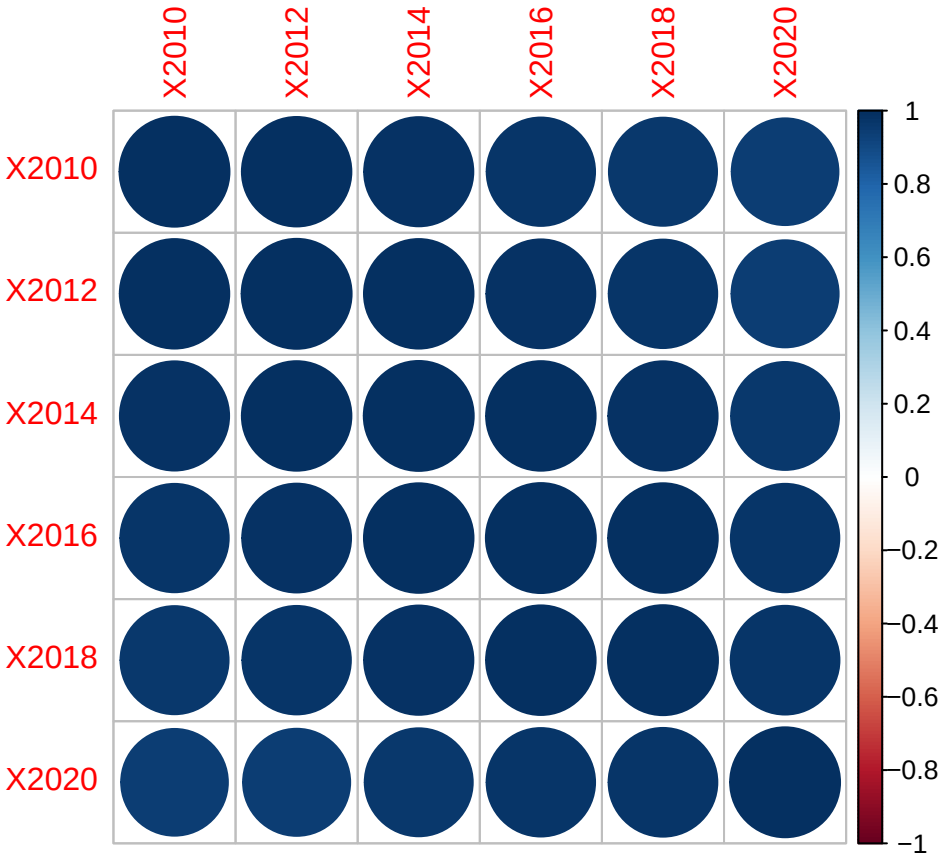
Let us additionally examine a correlation matrix to visualize pairwise relationships.

```
cor_matrix <- cor(expenditures_reg_years)
cor_matrix
```

```
##           X2010     X2012     X2014     X2016     X2018     X2020
## X2010 1.0000000 0.9915510 0.9859351 0.9713223 0.9665964 0.9419784
## X2012 0.9915510 1.0000000 0.9922812 0.9802877 0.9733346 0.9485583
## X2014 0.9859351 0.9922812 1.0000000 0.9925397 0.9877689 0.9644967
## X2016 0.9713223 0.9802877 0.9925397 1.0000000 0.9930394 0.9726975
## X2018 0.9665964 0.9733346 0.9877689 0.9930394 1.0000000 0.9776730
## X2020 0.9419784 0.9485583 0.9644967 0.9726975 0.9776730 1.0000000
```

The same matrix as a heatmap:

```
corrplot(cor_matrix)
```



As evidenced by the heatmap and the correlation matrix, there are extremely strong positive pairwise correlations between personal healthcare costs by year, verifying our earlier intuition. This could be because of a variety of factors: general economic inflation, chronic health conditions that worsen with time and thus require more expensive treatment, etc.

To quantitatively express the amount of multicollinearity in our model, let us examine the VIFs obtained from fitting the model including all possible predictors.

```
grand_model <- lm(X2020 ~ ., expenditures_reg)
vif(grand_model)
```

```
##  Age.Group        Sex      X2010      X2012      X2014      X2016      X2018
##   1.218719   1.041096  68.287875 137.158421 235.667212 136.781300  98.525344
##    Is_Home    Is_Drug
##   1.346552   1.550236
```

A VIF greater than or equal to 10 suggests that substantial multicollinearity is present, and all of our quantitative predictors have VIFs much higher than that. Therefore, we do have serious multicollinearity present in our model.

Multicollinearity is a concern, though not a particularly serious one as our analysis goal is prediction. And as long as expenditures in year 2020 follow the same multicollinear trend as our model, then our model's prediction accuracy will not be impacted. And as evidenced by the pairwise scatterplots, *X2020* does appear to be linearly related to the previous years.

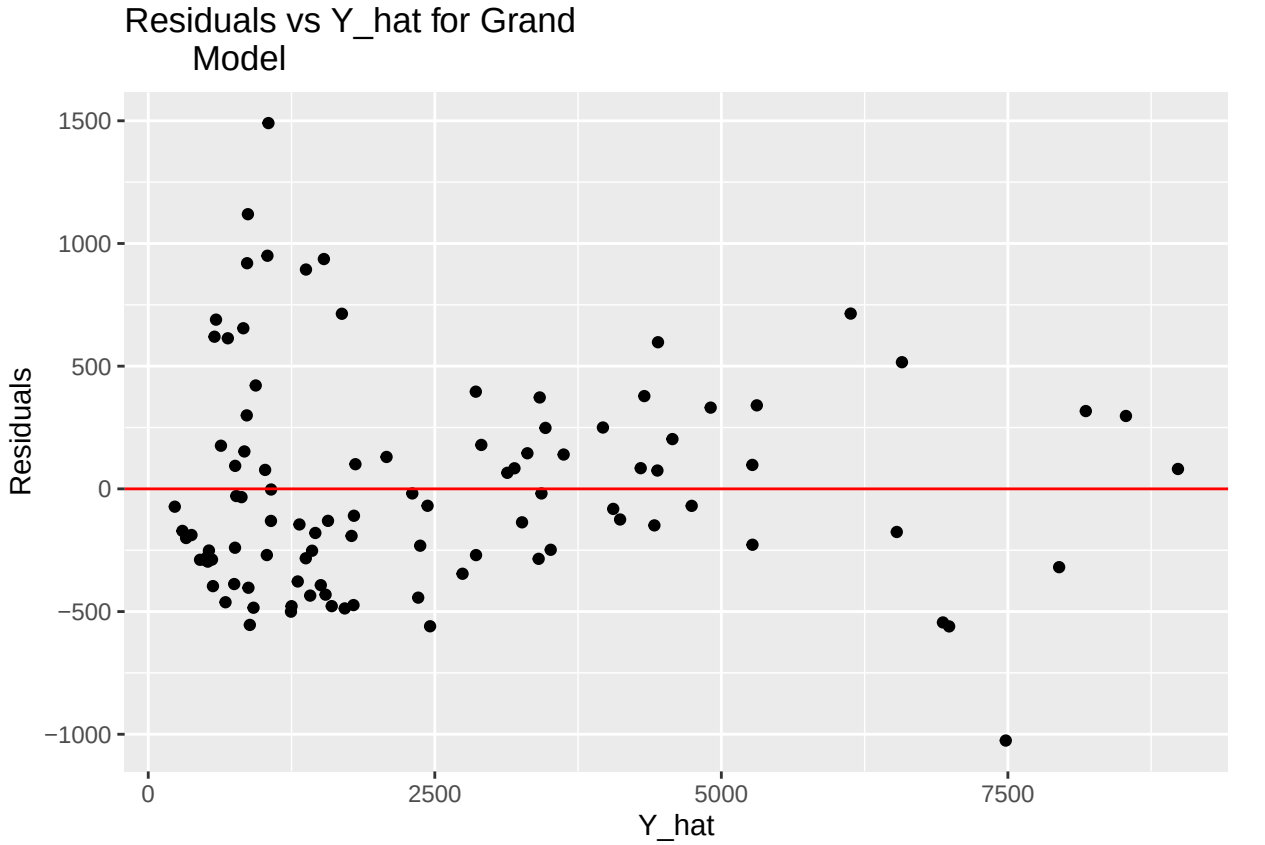
Multicollinearity however does significantly impact parameter estimation; when predictors are highly correlated, it is difficult to isolate the impact of an individual predictor on the response. So if estimation was our goal, we would consider steps such as centralizing our predictors and/or considering LASSO or Ridge regression in lieu of Ordinary Least Squares regression to mitigate the issue. Centering reduces correlation between individual

variables and any potential interaction terms. Ridge regression utilizes a biasing constant that minimizing the variance of predictors at the cost of introducing more bias. Finally, LASSO addresses multicollinearity by either outright removing highly correlated predictors, or by shrinking their relative impact on the response.

Nevertheless, under the grand model, we still need to verify independence, constancy of error variance, and normality.

Let us examine a residuals vs fitted values plot.

```
grand_model <- lm(X2020 ~ ., expenditures_reg)

grand_model_df <- data.frame(Y_hat = fitted(grand_model),
  Residuals = resid(grand_model)
)

ggplot(grand_model_df, aes(x = Y_hat, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Y_hat", y = "Residuals", title = "Residuals vs Y_hat for Grand
       Model")
```



Residuals appear to be more or less evenly distributed about the horizontal line at 0, so we can say they have mean 0 or close to it. There are two obvious outlying observations: one that is approximately 1000 units below 0, and another 1500 units above 0. And as previously evidenced by the scatterplot matrix, but more clearly seen here in the residual vs fitted plot, we have substantially more observations for lower personal expenditures as opposed to higher personal expenditures. Since there is no funnel or megaphone shape in the residual plot, we may verify the assumption of constant variance.

We can construct plots of residuals vs predictors to verify independence. However, since we have nine potential predictors, it is more efficient to use the Durbin-Watson test instead.

We will use a significance level of 0.05. The null hypothesis states that error terms (residuals) are not autocorrelated (independent). The alternative hypothesis states that they are autocorrected (not independent).

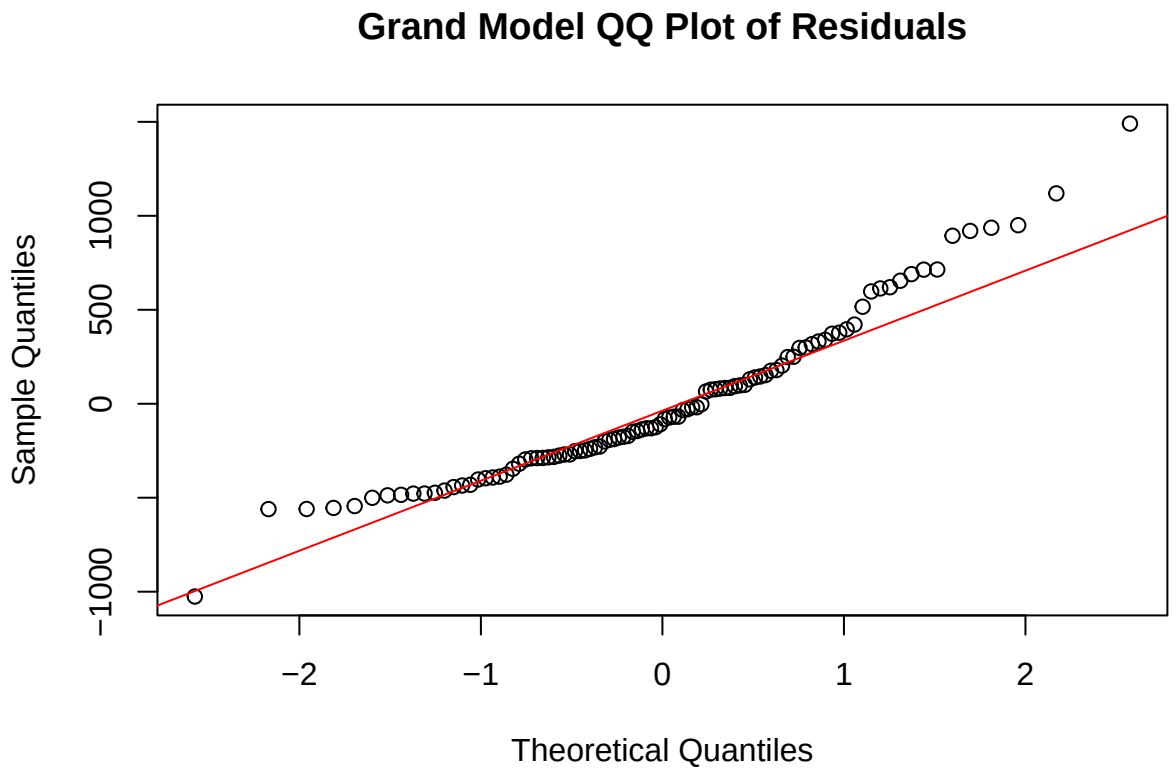
```
durbinWatsonTest(grand_model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.4262994      1.134117       0
##  Alternative hypothesis: rho != 0
```

Since $0 < 0.05$, we reject the null hypothesis and conclude our assumption of independence is not met. A DW statistic of 1.13 falls within 0 and 2, so we know the autocorrelation present in the model is positive. This does not come at a surprise however, given the highly correlated relationship between the expenditures by year that we previously observed in the heatmap and correlation matrix.

We will look to a QQplot to assess normality.

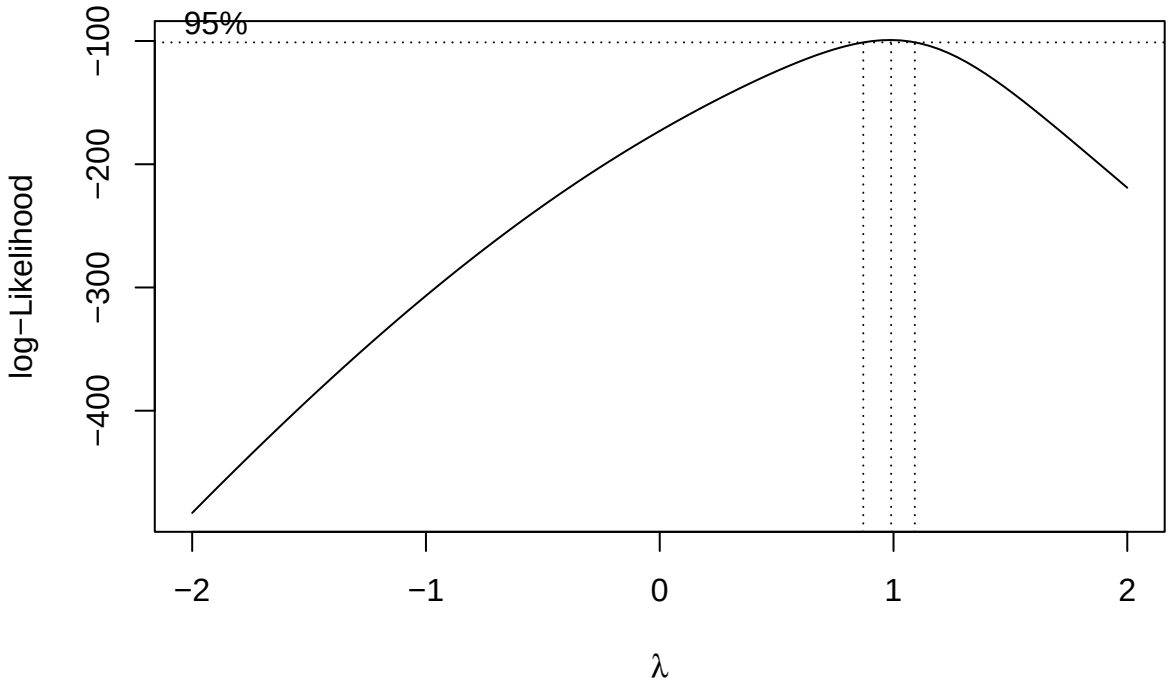
```
qqnorm(resid(grand_model), main = "Grand Model QQ Plot of Residuals")
qqline(resid(grand_model), col = "red")
```



The red line on a qqplot represents ideal normality. Significant deviations from this line therefore represent violations of the normality assumption. In our qqplot, the points at the ends curve away from the red line, indicating that both tails are heavier than those of a normal distribution. The points located in the upper right particularly deviate, indicating that observations of high healthcare expenditures produce notable outliers.

The majority of our points follow the red line, but because our right tail is unusually heavy, we should further investigate the assumption of normality. We can perform this investigation with the help of a Box-Cox plot. If we get an optimal lambda value of 1 or very close to 1, then we know our data is already normally distributed, and we don't need transform our response variable.

```
boxcox_result <- boxcox(grand_model)
```




```
boxcox_result
```

The lambda value that maximizes the log-likelihood and its 95% confidence interval is given by,

```
best_lambda <- boxcox_result$x[which.max(boxcox_result$y)]
best_lambda
```

```
## [1] 0.989899
```

```
alpha <- 0.05
chi_sq <- qchisq(1 - alpha, df = 1) / 2
log_lik_max <- max(boxcox_result$y)
lower_bound <- min(boxcox_result$x[boxcox_result$y > log_lik_max - chi_sq])
upper_bound <- max(boxcox_result$x[boxcox_result$y > log_lik_max - chi_sq])

cat("The 95% confidence interval for the optimal lambda is: [", lower_bound, ",", upper_bound, "]\n")
```

```
## The 95% confidence interval for the optimal lambda is: [ 0.9090909 , 1.070707 ]
```

Because 1 is contained in the optimal lambda's 95% confidence interval, and the optimal lambda itself is very nearly 1, we conclude that no transformation is necessary and that the assumption of normality is met.

We have finished investigating both our numerical and categorical predictors for possible issues. We have also completed checking regression assumptions under the grand model. Our next step is building our multiple linear regression model.

Part V: Model Selection/Construction.

We will first partition our data 80/20 into training and testing sets. The training data will be used to fit our multiple linear regression model and the testing set will be used to assess our final model's prediction accuracy.

```
set.seed(117)

expenditures_reg_train  <- expenditures_reg %>% dplyr::sample_frac(0.8)
```

```
expenditures_reg_test <- dplyr::anti_join(expenditures_reg,
                                          expenditures_reg_train, by = 'X2020')
```

```
head(expenditures_reg_train)
```

```
##   Age.Group Sex X2010 X2012 X2014 X2016 X2018 X2020 Is_Home Is_Drug
## 1         4   1   462   644   889  1052  1256  1317       1       0
## 2         2   0   135   197   220   221   268   241       0       0
## 3         1   0    81   122   131   164   386   810       0       0
## 4         1   1    95   143   158   199   468   991       0       0
## 5         4   1  3965  4088  4959  6128  6372  7092       0       1
## 6         3   1   831   849   870   903   825   772       0       1
```

```
head(expenditures_reg_test)
```

```
##   Age.Group Sex X2010 X2012 X2014 X2016 X2018 X2020 Is_Home Is_Drug
## 1         3   0  2501  2497  2603  3088  3169  3125       0       0
## 2         4   1  1357  1342  1664  1986  2112  2141       0       0
## 3         1   1   235   243   256   349   464   471       0       0
## 4         5   0    13    21    25    33    76   159       0       0
## 5         2   0    12    19    21    27    63   130       0       0
## 6         3   0   806  1026  1074  1255  1403  2469       0       0
```

We will find our best model using all subsets selection. We favor this method because we avoid the errors associated with stepwise selection methods. For example, forward selection may miss important variables that only show their effect when other variables are included, while backward selection might eliminate important variables too early. By not operating in a stepwise manner, all subsets selection bypasses these mistakes.

Additionally, because we have relatively few potential predictors at 9, there are only 2^9 = 512 possible models to consider under this selection method. This is a very reasonable calculation for most modern computers. If we had dozens of potential predictors and if computational cost was an issue, then we would have to consider stepwise model selection methods instead.

```
subset <- regsubsets(X2020 ~ ., expenditures_reg_train, nvmax = 9)
results <- summary(subset)
results
```

```
## Subset selection object
## Call: regsubsets.formula(X2020 ~ ., expenditures_reg_train, nvmax = 9)
## 9 Variables  (and intercept)
##           Forced in Forced out
## Age.Group     FALSE      FALSE
## Sex           FALSE      FALSE
## X2010         FALSE      FALSE
## X2012         FALSE      FALSE
## X2014         FALSE      FALSE
## X2016         FALSE      FALSE
## X2018         FALSE      FALSE
## Is_Home       FALSE      FALSE
## Is_Drug       FALSE      FALSE
## 1 subsets of each size up to 9
## Selection Algorithm: exhaustive
##          Age.Group Sex X2010 X2012 X2014 X2016 X2018 Is_Home Is_Drug
## 1  ( 1 ) " "       " " " "   " "   " "   " "   "*"   " "     " "
## 2  ( 1 ) " "       " " " "   " "   " "   " "   "*"   "*"     " "
## 3  ( 1 ) " "       "*" " "   " "   " "   " "   "*"   "*"     " "
```

```
## 4  ( 1 ) " "       "*" " "   "*"   " "   " "   "*"   "*"     " "
## 5  ( 1 ) " "       "*" " "   "*"   " "   "*"   "*"   "*"     " "
## 6  ( 1 ) "*"       "*" " "   "*"   " "   "*"   "*"   "*"     " "
## 7  ( 1 ) "*"       "*" " "   "*"   " "   "*"   "*"   "*"     "*"
## 8  ( 1 ) "*"       "*" "*"   "*"   " "   "*"   "*"   "*"     "*"
## 9  ( 1 ) "*"       "*" "*"   "*"   "*"   "*"   "*"   "*"     "*"
```

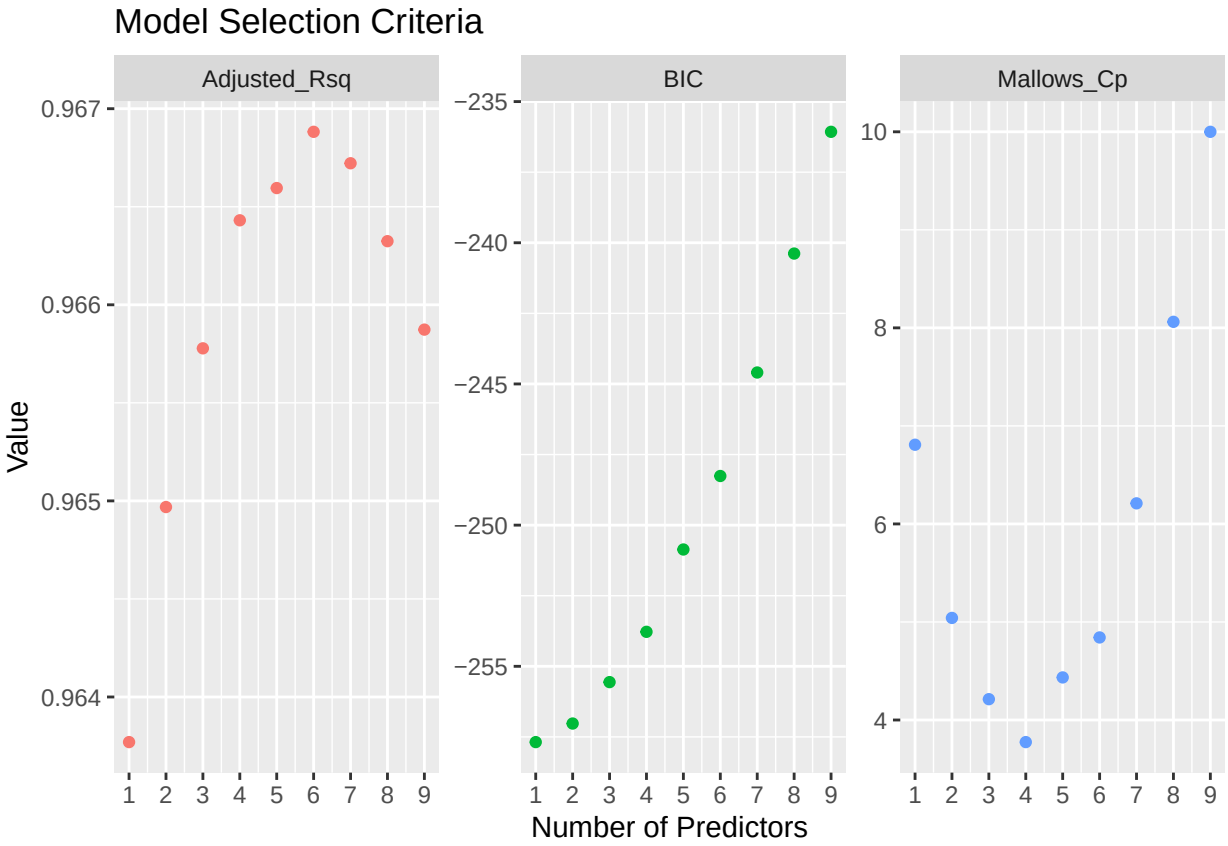
From the output above, we see that *X2020 ~ X2018* is the best one predictor model, *X2020 ~ X2018 + Is_Home* is the best two predictor model, *X2020 ~ X2018 + Is_Home + Sex* is the best three predictor model, etc. Since we found high autocorrelation in the previous section, it is unsurprising that *X2018* is present in all models.

We have a number of criteria for model selection: R^2, adjusted R^2, Mallow's Cp, PRESS, AIC, and BIC. The `results` list contains four of these criteria: AIC, BIC, Mallow's Cp, R^2, and adjusted R^2. However, we can immediately rule out R^2, the coefficient of determination, as a selection criterion because R^2 only increases with number of predictors added to the model. This is because R^2 = 1 - SSE/SSTO, and SSE (Error Sum of Squares) strictly decreases with additional predictors–regardless of their significance. Thus the model that maximizes R^2 is very likely overfitted.

On the other hand, adjusted R^2 corrects for increasing number of predictors by dividing SSE and SSTO by their associated degrees of freedom (n - p and n - 1, respectively). Hence, if the loss a degree of freedom n - p outweighs the additional decrease in SSE, adjusted R^2 can actually go down by introducing more predictors. This adjustment therefore corrects for overfitting.

Thus we will examine Mallow's Cp, BIC, and adjusted R^2 from `results` to select our best model. Mallow's Cp compares total MSE (Mean Squared Error) to the true error variance. The Bayesian Information Criterion (BIC) is closely related to the Akaike Information Criterion (AIC); however, BIC tends to favor more parsimonious models (that is, models with fewer predictors) than AIC does.

```
tibble(predictors = 1:9,
       Mallows_Cp = results$cp,
       Adjusted_Rsq = results$adjr2,
       BIC = results$bic) %>%
  gather(statistic, value, -predictors) %>%
  ggplot(aes(predictors, value, color = statistic)) +
  geom_point(show.legend = F) +
  scale_x_continuous(breaks = 1:9) +
  facet_wrap(~ statistic, scales = "free") +
  labs(title = "Model Selection Criteria", x = "Number of Predictors", y =
      "Value")
```



For our decision criteria, we desire models that maximize adjusted R^2, minimize BIC, and minimize Cp (small total MSE) such that Cp is near p (small bias), where p is the number of coefficients in the model (a model with p - 1 predictors has p coefficients).

As evidenced by the plots above, all of the models have almost identical adjusted R^2 values, differing only by a few thousandths of decimals. BIC suggests that the one-variable model is the best, but we know this criterion favors models with fewer predictors.

Let us examine the Mallow's Cp values along with their corresponding number of predictors more closely:

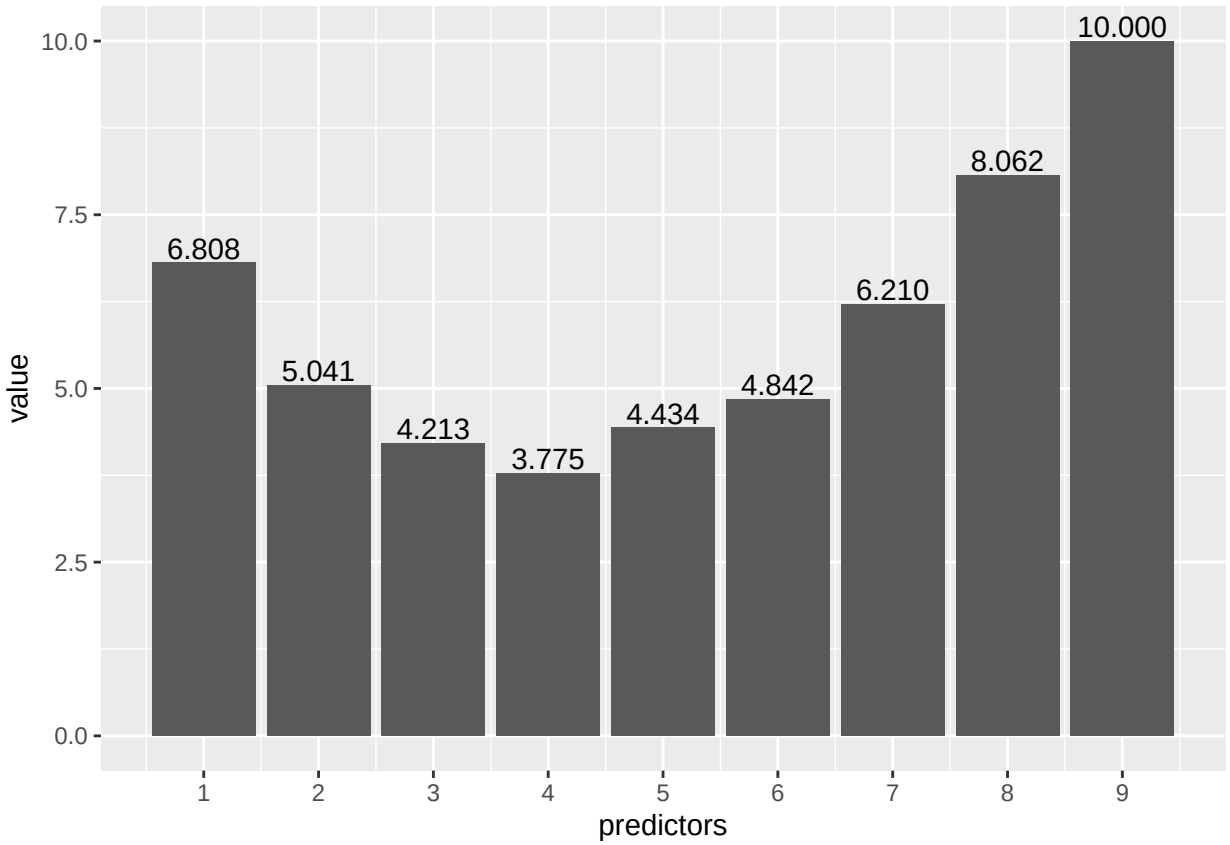
```
results$cp
```

```
## [1]  6.807817  5.041343  4.212502  3.774638  4.434463  4.842133  6.210102
## [8]  8.061624 10.000000
```

```
rep(1:9)
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
tibble(predictors = 1:9, Mallows_Cp = results$cp) %>%
    gather(statistic, value, -predictors) %>%
    ggplot(aes(predictors, value)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = sprintf("%.3f", value)), vjust = -0.2, position =
                position_dodge(width = 0.9)) +
    scale_x_continuous(breaks = 1:9)
```



We have a dilemma here. The four predictor model, *X2020 ~ X2018 + Is_Home + Sex + X2012* has the lowest Cp value at 3.775. However, the three predictor model, *X2020 ~ X2018 + Is_Home + Sex* has a Cp value of 4.213, which is the closest to p + 1 = 3 + 1 = 4. Thus the latter model is less biased, but the former model fewer total MSE.

We will select the model with four predictors, *X2020 ~ X2018 + Is_Home + Sex + X2012*, fit it on the training data, and conduct a t-Test test to ascertain whether we can drop the predictor *X2012*. This test investigate the marginal significance of *X2012* given *X2018 + Is_Home + Sex* already being present in the model.

```
four_var <- lm(X2020 ~ X2018 + Is_Home + Sex + X2012,
                   expenditures_reg_train)
summary(four_var)
```

```
##
## Call:
## lm(formula = X2020 ~ X2018 + Is_Home + Sex + X2012, data = expenditures_reg_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1115.10  -297.82   -66.05   208.32  1056.75
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 240.96528   98.64684   2.443   0.0169 *
## X2018         1.06454    0.08752  12.164   <2e-16 ***
## Is_Home     211.38195  101.06679   2.092   0.0399 *
## Sex         195.78609   98.71198   1.983   0.0510 .
## X2012        -0.15843    0.10064  -1.574   0.1196
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 428.3 on 75 degrees of freedom
## Multiple R-squared:  0.9681, Adjusted R-squared:  0.9664
## F-statistic: 569.6 on 4 and 75 DF,  p-value: < 2.2e-16
```

Our null hypothesis is the regression coefficient (beta_4) for *X2012* is 0, and the alternative hypothesis is that it is not equal to 0. We will use a significance level of 0.05. Our test statistic is t_star = beta_4/(standard error of beta_4). If |t_star| <= t(1 - (0.05/2), 80 - 5), we fail to reject the null; if |t_star| > t(1 - (0.05/2), 80 - 5) we reject the null hypothesis.

```
summary(four_var)
```

```
##
## Call:
## lm(formula = X2020 ~ X2018 + Is_Home + Sex + X2012, data = expenditures_reg_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1115.10  -297.82   -66.05   208.32  1056.75
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 240.96528   98.64684   2.443   0.0169 *
## X2018         1.06454    0.08752  12.164   <2e-16 ***
## Is_Home     211.38195  101.06679   2.092   0.0399 *
## Sex         195.78609   98.71198   1.983   0.0510 .
## X2012        -0.15843    0.10064  -1.574   0.1196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 428.3 on 75 degrees of freedom
## Multiple R-squared:  0.9681, Adjusted R-squared:  0.9664
## F-statistic: 569.6 on 4 and 75 DF,  p-value: < 2.2e-16
```

Then |t_star| = |-0.15843/0.10064| = |-1.57422| = 1.57422. We compare this value to t(0.975, 75).

```
qt(p = 0.05/2, df = 75, lower.tail = FALSE)
```

```
## [1] 1.992102
```

Since 1.57422 < 1.992102, we fail to reject the null. We conclude that there is significant statistical evidence to suggest that regression coefficient for *X2012* is 0, so we can drop this predictor. However, the validity of this test is confounded by the fact that we have very high multicollinearity, as evidenced by the VIF values below (VIF values for *X2018* and *X2012* both exceed 10).

```
vif(four_var)
```

```
##     X2018   Is_Home       Sex     X2012
## 20.243528  1.029377  1.059649 20.391780
```

Having found our "best" model, we proceed with diagnostics in the next section.

Part VI: Model Diagnostics

First we fit our selected model on the training set.

```r
selected_model <- lm(X2020 ~ X2018 + Is_Home + Sex, expenditures_reg_train)
summary(selected_model)
```

```
##
## Call:
## lm(formula = X2020 ~ X2018 + Is_Home + Sex, data = expenditures_reg_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1184.72  -312.81   -73.54   210.14  1047.58
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 277.63238   96.78559   2.869  0.00534 **
## X2018         0.93027    0.01981  46.963  < 2e-16 ***
## Is_Home     215.46329  102.01146   2.112  0.03796 *
## Sex         163.81810   97.53576   1.680  0.09715 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 432.5 on 76 degrees of freedom
## Multiple R-squared:  0.9671, Adjusted R-squared:  0.9658
## F-statistic: 744.2 on 3 and 76 DF,  p-value: < 2.2e-16
```

Our overall model is significant as evidenced by the p-value less than 2.2e-16.

Let us examine the VIF of each predictor.

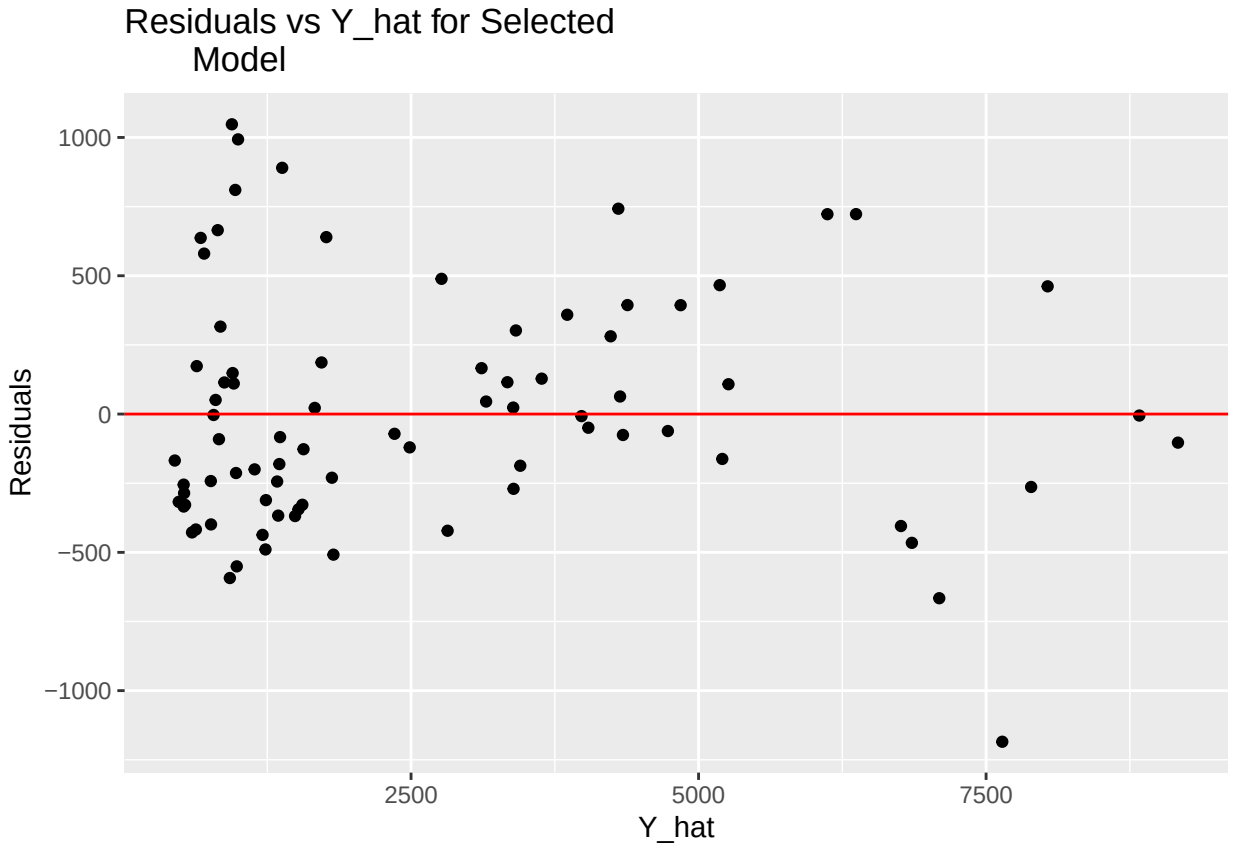
```r
vif(selected_model)
```

```
##    X2018  Is_Home      Sex
## 1.017283 1.028700 1.014807
```

Since all these values are less than 10, our issues with multicollinearity are substantially alleviated by removing *X2012* from our model. But as stated before, since our analysis goal is prediction, multicollinearity was not a much of an issue to begin with.

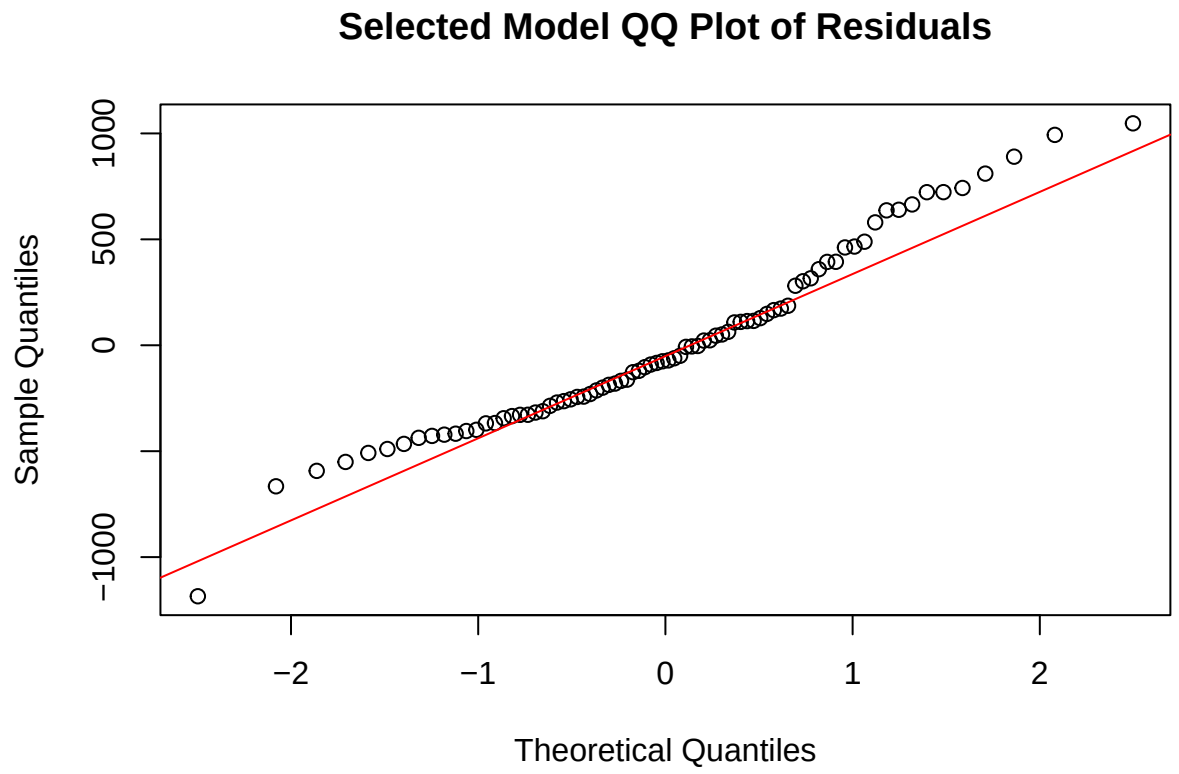
Akin to what we did for the grand model, let us verify constant error variance and normality for our selected model.

```r
selected_model_df <- data.frame(Y_hat = fitted(selected_model),
  Residuals = resid(selected_model)
)

ggplot(selected_model_df, aes(x = Y_hat, y = Residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Y_hat", y = "Residuals", title = "Residuals vs Y_hat for Selected
       Model")
```



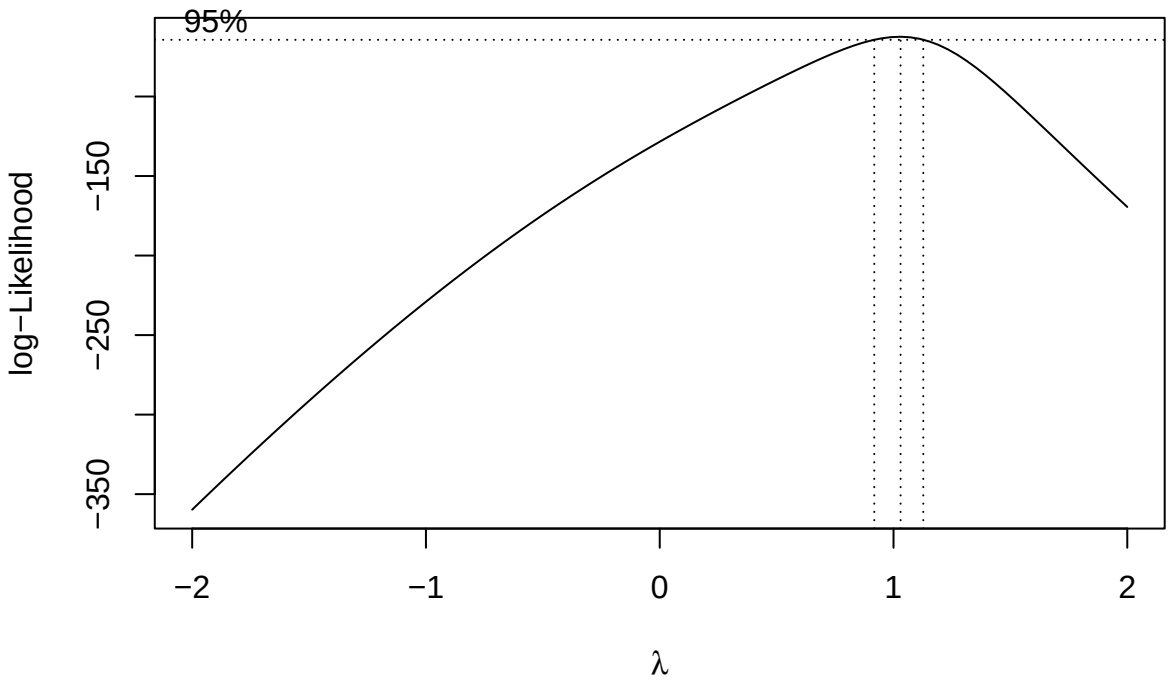
With the exception of some outlying values, residuals appear to be more or less evenly distributed about the horizonal line, so we conclude they have mean 0. Since there is no funnel or megaphone shape in the residual plot, we may also conclude constant variance (ie no hetereoskedascity).

```
qqnorm(resid(selected_model), main = "Selected Model QQ Plot of Residuals")
qqline(resid(selected_model), col = "red")
```



We have unusually heavy tails in our qqplot, so before we conclude normality, let us conduct a further investigation by examining lambda from a boxcox transformation.

```
boxcox_select <- boxcox(selected_model)
```




```
boxcox_select
```

The lambda value that maximizes the log-likelihood and its 95% confidence interval is given by,

```
best_lambda_select <- boxcox_select$x[which.max(boxcox_select$y)]
best_lambda_select
```

```
## [1] 1.030303
```

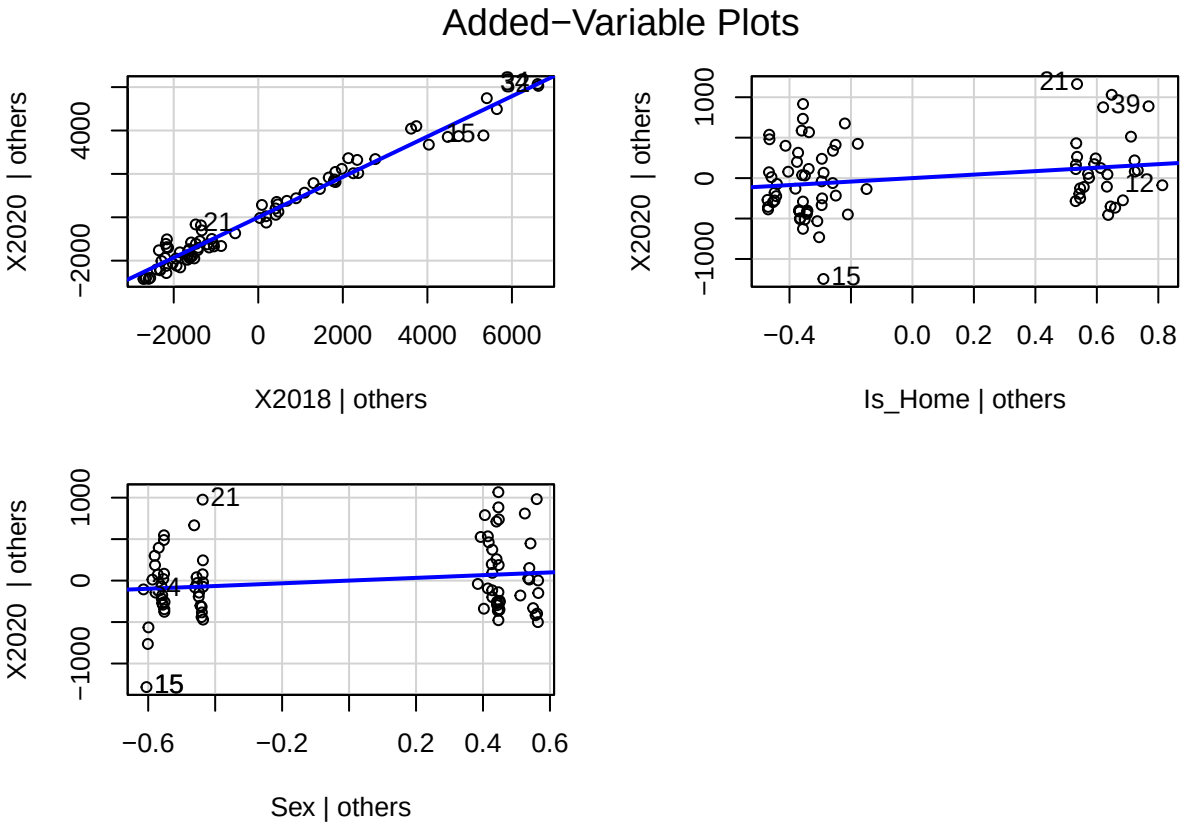
```
alpha <- 0.05
chi_sq <- qchisq(1 - alpha, df = 1) / 2
log_lik_max <- max(boxcox_select$y)
lower_bound <- min(boxcox_select$x[boxcox_select$y > log_lik_max - chi_sq])
upper_bound <- max(boxcox_select$x[boxcox_select$y > log_lik_max - chi_sq])

cat("The 95% confidence interval for the optimal lambda is: [", lower_bound, ",", upper_bound, "]\n")
```

```
## The 95% confidence interval for the optimal lambda is: [ 0.9494949 , 1.111111 ]
```

Because our optimal lambda value is extremely close to 1, and 1 itself is contained in the 95% confidence interval, we conclude no transformation is necessary and the assumption of normality is satisfied.

To examine the marginal relationship between the response variable and each of our three predictors, *X2018*, *Is_Home*, *Sex*, we can analyze their respective partial regression plots

```
car::avPlots(selected_model)
```



Each plot shows the relationship between *X2020* and one independent variable, while controlling for other variables in the model. The blue line represents the linear regression line between the two. A linear relationship is suggested if the points roughly cluster around this line.

The top left partial regression shows a reasonably strong positive linear relationship. Our partial regression plots for our two categorical predictors (*Is_Home* and *Sex* on the top right and bottom left, respectively) indicate a very weak positive linear relationship.

We found some unusual observations in our residuals vs fitted plot, so we will proceed to look for any outliers or influential points.

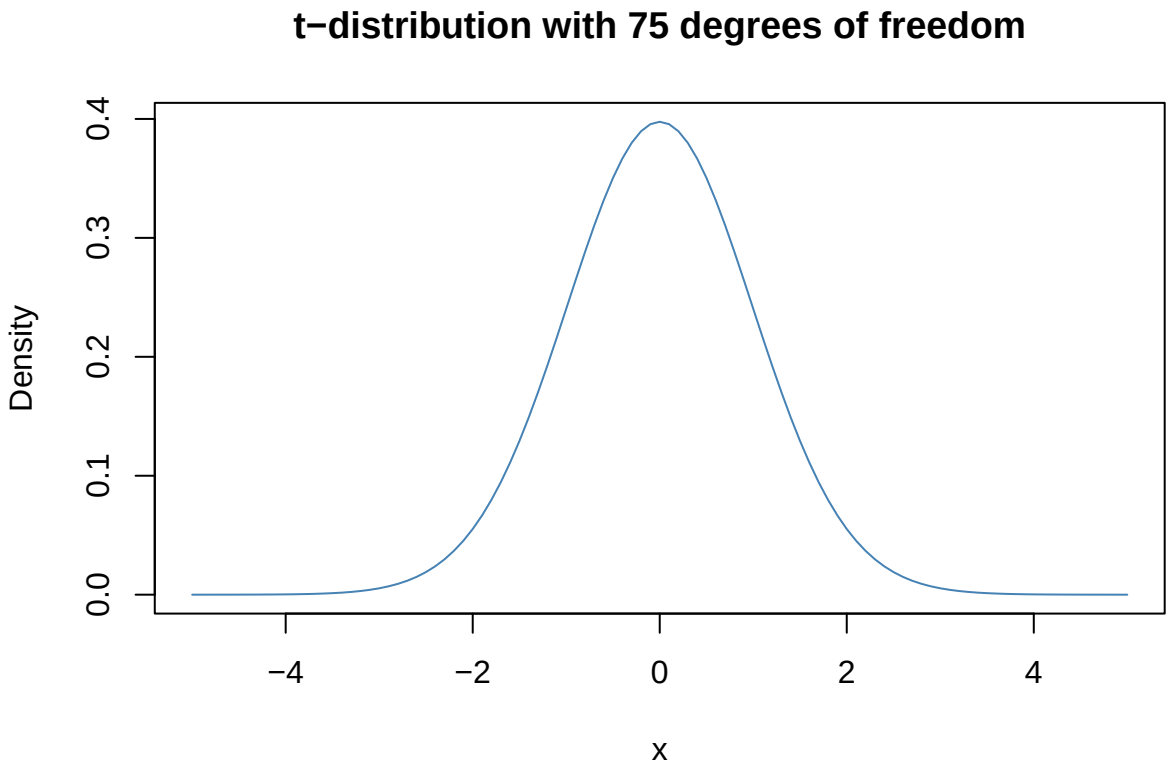
Outlying Y observations will have large corresponding absolute values of studentized deleted residuals. If there are no outliers, each studentized residual will follow a t distribution with n - p - 1 = 80 - 4 - 1 = 75 degrees of freedom.

```
stud_resids <- studres(selected_model)
stud_resids_df <- data.frame(stud_resid = stud_resids) %>%
  arrange(desc(abs(stud_resid)))

head(stud_resids_df, 10)
```

```
##    stud_resid
## 15  -3.031911
## 21   2.574773
## 7    2.415772
## 73   2.167208
## 64   1.946180
## 9    1.790780
## 39   1.765479
## 5    1.743779
## 70  -1.625108
## 17   1.585188
```

The 10 observations with the largest studentized deleted residuals by absolute value are shown above. We will compare the values in *stud_resid* to the following t-distribution.

```
curve(dt(x, df=75), from=-5, to=5,
      main = 't-distribution with 75 degrees of freedom',
      ylab = 'Density',
      col = 'steelblue')
```



Observation 15 appears concerning. Its studentized deleted residual of -3.031911 is very close to the left tail of the t-distribution above. We will test the probability of observing that value or more extreme at alpha = 0.05.

```
pt(-3.031911, 75, lower.tail = TRUE) * 2
```

```
## [1] 0.003335417
```

Because $0.003335417 < 0.05$, we conclude that Observation 15 is indeed a Y-outlier.

We just manually performed the Bonferroni outlier test. We can verify our results with the output of the `outlierTest` function in R, which performs the same test.

```
outlierTest(selected_model)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 15 -3.031911          0.0033354      0.26683
```

our conclusions are indeed consistent with the output of `outlierTest`.

To spot influential cases, we will consider their DFFITS values. For our small dataset of only 80 observations, if the DFFITS absolute value exceeds 1, then the observation associated with it is considered influential.

```
dffits_vals <- dffits(selected_model)
dffits_vals_df <- data.frame(dffits = dffits_vals) %>%
  arrange(desc(abs(dffits)))
head(dffits_vals_df, 10)
```

```
##        dffits
## 15 -0.9855981
## 21  0.5862994
## 39  0.5180851
## 73  0.5168222
## 70 -0.4893546
## 7   0.4857758
## 9   0.4287681
## 5   0.4266203
## 64  0.3925889
## 30  0.3534654
```

The 10 observations with the largest DFFITS by absolute value are shown above. None of the DFFITS values exceed 1 in absolute value, but Observation 15's DFFITS comes close at -0.9855981. So while we previously confirmed Observation 15 to be an outlier, it is not considered unduly influential.

If our DFFITS results revealed any influential observations, we would have to find ways to remedy them.

Ordinary Least Squares is susceptible to outlying cases since they can result in a distorted model that doesn't fit non-outlying cases very well. We may drop outlying cases if we know they are a product of erroneous data collection. However, since we don't have access to the CMC's data collection methods, we should consider alternative ways to handle these outliers instead of omitting them.

That is, we need to make our model less susceptible to the sway of influential cases. Robust regression is particularly helpful toward this end because it diminishes the influence of outliers without excluding them from the dataset. Under robust regression, weights are assigned to each observation, with lower weights given to data points that are identified as outliers based on their residuals. This weighting scheme reduces the impact of these extreme values on the overall model fit.

Thus, we would consider fitting $X2020 \sim X2018 + Is\_Home + Sex$ via Robust regression if we spotted some particularly influential cases.

This concludes our model diagnostics section. In this section, we verified our regression assumptions, checked for multicollinearity, examined the marginal relationships between the predictor and each response, and looked for outlying/influential cases. We do not have any serious model issues to remedy, so we can move on to evaluating our model performance on the testing set.

PART VII: EVALUATING MODEL PERFORMANCE

These are our model's predicted values for 2020 personal healthcare expenditures.

```
prediction_2020 <- round(predict(selected_model,
                                 expenditures_reg_test))
prediction_2020
```

```
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 3226 2406  873  348  336 1583  646  300 2850 1021  983 1493 2930 2306 2437 4225
##   17   18   19   20
##  437 1405 3541 2099
```

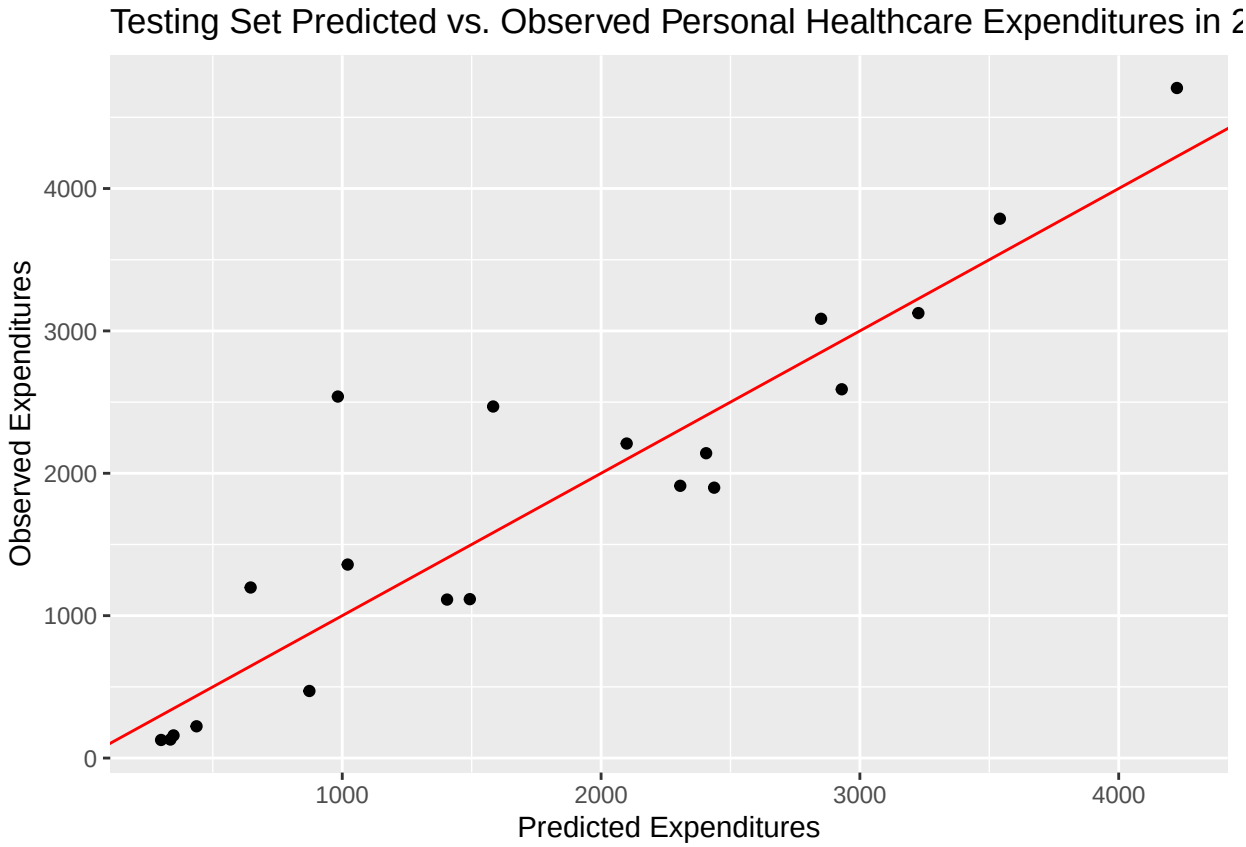
Let us compare them to the observed values in the test dataset.

```
obs_and_pred <- data.frame(observed = expenditures_reg_test$X2020,
                           predicted = prediction_2020)
obs_and_pred
```

```
##    observed predicted
## 1      3125      3226
## 2      2141      2406
## 3       471       873
## 4       159       348
## 5       130       336
## 6      2469      1583
## 7      1198       646
## 8       127       300
## 9      3085      2850
## 10     1359      1021
## 11     2539       983
## 12     1116      1493
## 13     2590      2930
## 14     1912      2306
## 15     1899      2437
## 16     4706      4225
## 17      223       437
## 18     1113      1405
## 19     3788      3541
## 20     2209      2099
```

We can examine a predicted vs observed plot to evaluate the appropriateness of a linear fit.

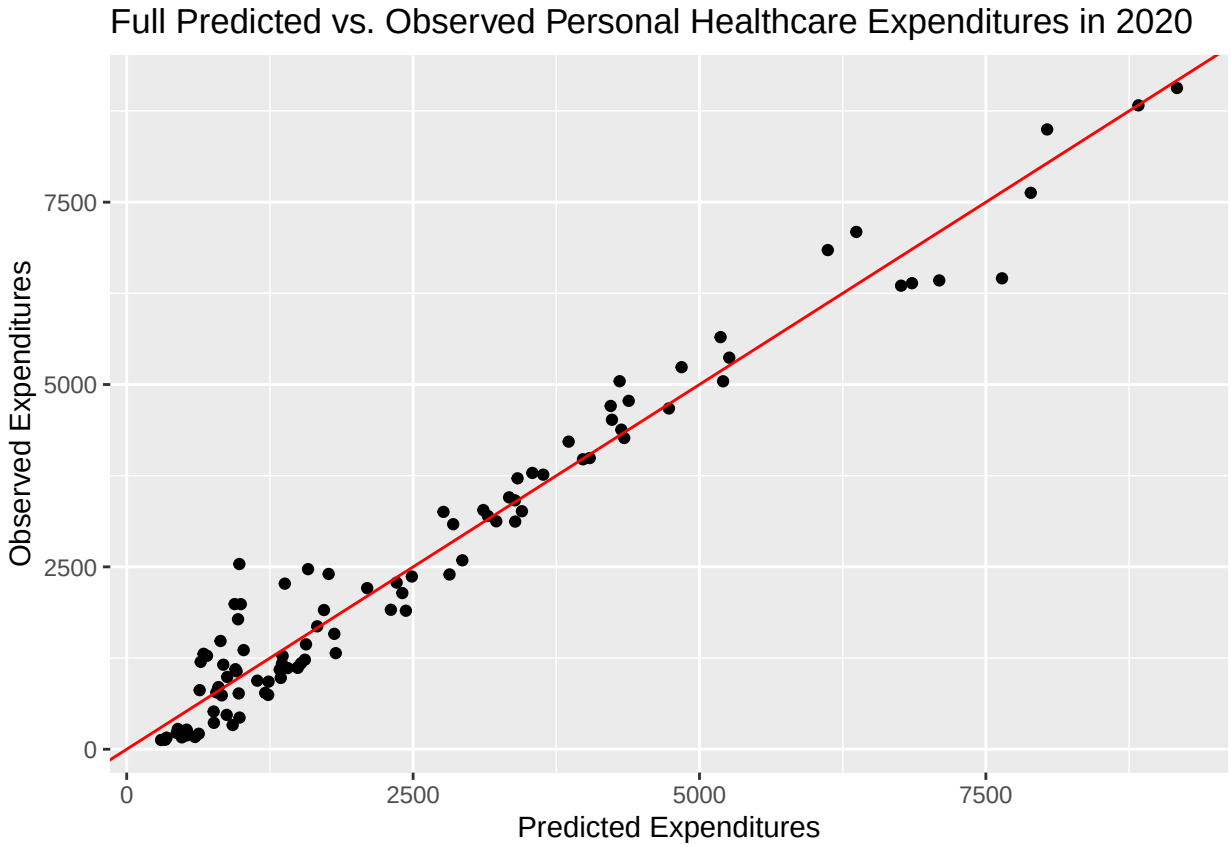
```
ggplot(obs_and_pred, aes(x = predicted, y = observed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  labs(x = 'Predicted Expenditures',
       y = 'Observed Expenditures',
       title = 'Testing Set Predicted vs. Observed Personal Healthcare Expenditures in 2020')
```



Barring a few outlying values, each observation lines up nicely with the line y=x, so we can conclude that our model fits the testing data fairly well.

Since our model performs well on our testing data, let's examine a fitted vs observed plot for the full 100 observation dataset.

```
prediction_2020_full <- round(predict(selected_model,
                                      expenditures_reg))

obs_and_pred_full <- data.frame(observed = expenditures_reg$X2020,
                                predicted = prediction_2020_full)

ggplot(obs_and_pred_full, aes(x = predicted, y = observed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red") +
  labs(x = 'Predicted Expenditures',
       y = 'Observed Expenditures',
       title = 'Full Predicted vs. Observed Personal Healthcare Expenditures in 2020')
```



We observe that our model performs comparably well on the full dataset as it does on the testing dataset.

PART VIII: CONCLUSION

Statistical tools/techniques to predict personal healthcare expenditures are valuable not only to health insurers, but also to public policymakers and most importantly, consumers themselves. In this analysis, we examined a CMS dataset of medical expenditures by factors such as age range, year, sex, etc. Our goal was to build an appropriate multiple linear regression model to predict personal healthcare expenditures of non-out-of-pocket payers in 2020. The statistical concepts taught in STAT 52500 were reflected in every step of this final project, demonstrating the practical application of statistical theory to real-world problems such as healthcare expenditure modelling.

We had to first clean our dataset, filtering a subset of data appropriate for regression and implementing code schemes for categorical variables. We then had to check our regression assumptions and verify whether we needed to transform our data. Next we had to select a model and remove any superfluous predictors. We verified regression assumptions for our chosen model and investigated for any outlying/influential points before finally evaluating its performance.