Math 3330: Regression Notes

Kelly Ramsay

2024-06-10

Table of contents

Preface

Welcome to the MATH 3330 Notes! Please install R and R studio (Or you can use VSCode if you're comfortable there.)

In order to make the most of these notes, do all of the exercises in the order they appear.

There will likely be typos and some errors, please let me know if you encounter any.

These notes should be used in conjunction with the book:

Title: Introduction to Linear Regression Analysis, *Volume 821 of Wiley Series in Probability and Statistics* Authors: Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining Edition: 5, illustrated, reprint Publisher: John Wiley & Sons, 2012 ISBN: 0470542810, 9780470542811

1 Introduction

1.1 What is the course about?

1.1.1 The main question

The whole course is concerned with the following problem: Suppose that X and Y are some attributes of a population. What is the relationship between X and Y. How can we use X to predict Y, or how can we use X to explain Y?

For example, questions of this form include:

- How is location, square feet, parking available related to the price of an Airbnb?
- How is hours played and age related to win rate in League of Legends?
- How are creatine and protein consumption related to deadlift 1RM?
- How is treatment (A or B) related to pain levels of patients?

All of these can be answered with regression!

Exercise 1.1. What is X and what is Y here?

- X: location, square feet, parking available Y: price of an Airbnb
- X: hours played and age Y: win rate in League of Legends
- X: creatine and protein consumption Y: deadlift 1RM
- X: treatment (A or B) Y: pain levels of patients

We suppose at the population level, **on average** that $Y = f(X)$. By on average, we mean that each person may not have exactly $Y = f(X)$,, but if we we average out Y for many people, we will have that the average is approximately $f(X)$. (This will be made more formal later).

For instance, consider the pain level question in the above example. Suppose that $f(A) = 2$ and $f(B) = 5$. Then, if we average the pain level of many patients who take treatment B, it should be close to 5.

Obviously, we cannot observe the whole population, and so we will assume that we have observed X and Y for a set of n individuals. Specifically, we observe some outcome $Y_1, \ldots, Y_n$, which is a real number and some attributes (categorical or numeric) about the n individuals, denoted by $X_1, \ldots, X_n$. Note that here X_i can be vectors or single numbers.

1.1.2 Using our data, how can we determine f?

Other, related questions:

- What is the form of f? Is it linear?
- How can we estimate f, say with $\hat{f}$? What is the best $\hat{f}$? What is the error of $\hat{f}$ on average?
- How can we tell if our model is good? i.e. how does $\hat{f}$ fit the data?
- How can we tell which X values are important? How can we tell if X is related to Y at all?
- What is the effect of correlation of X values?

These are all questions we will answer in this course.

Statistical modelling starts as follows:

1. Question about a population, e.g., "How are hours played and age related to win rate in League of Legends?"
2. Data: $(Y_1, X_1), \dots, (Y_n, X_n)$
3. Explore data with graphs and summary stats
4. Use exploratory data analysis to posit a model for the population.

Note that step 4 is necessary! Letting f be anything is too general and won't work well, so we need to use the data to give us a hint at the form of f! For instance, we might suppose that f is a linear function! That is, $f \in \{g(X) = X\beta \colon \beta \in \mathbb{R}^d\}$.

Next, we proceed with the following steps:

5. Estimation: How to get an estimate $\hat{\beta}$ of β?
6. Inference: What is the error of $\hat{\beta}$? Is f degenerate? I.e., is $\beta = 0$?
7. Fit: Does our fitted line match up with the data? What about the normality assumption? Do the errors appear normal?
8. Prediction: Predict any values if necessary.

1.1.3 Comparison with means example

Let's compare to what we learned in previous statistics courses about two sample testing with the above steps in mind. Below we have different hours of extra sleep for two different treatments. Let's see if the sleep for groups 1 and 2 differ.

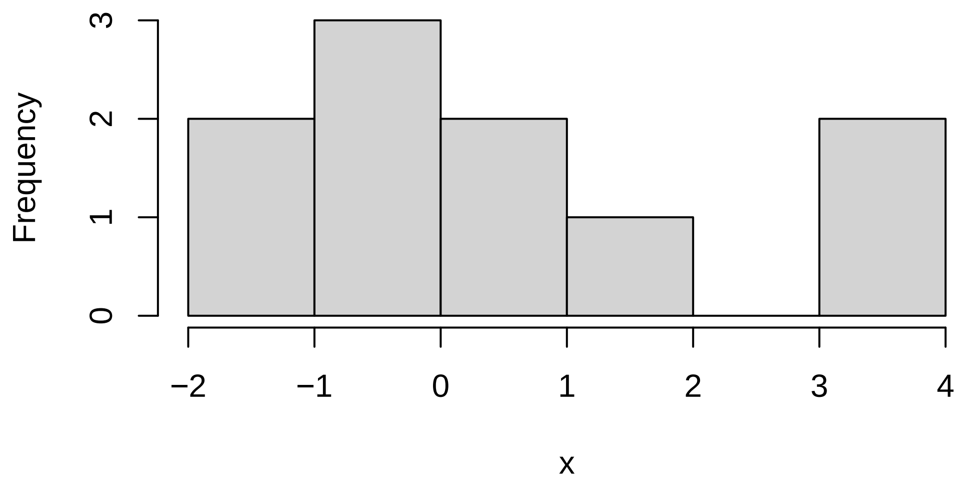
1. Do the counts for A and B differ?

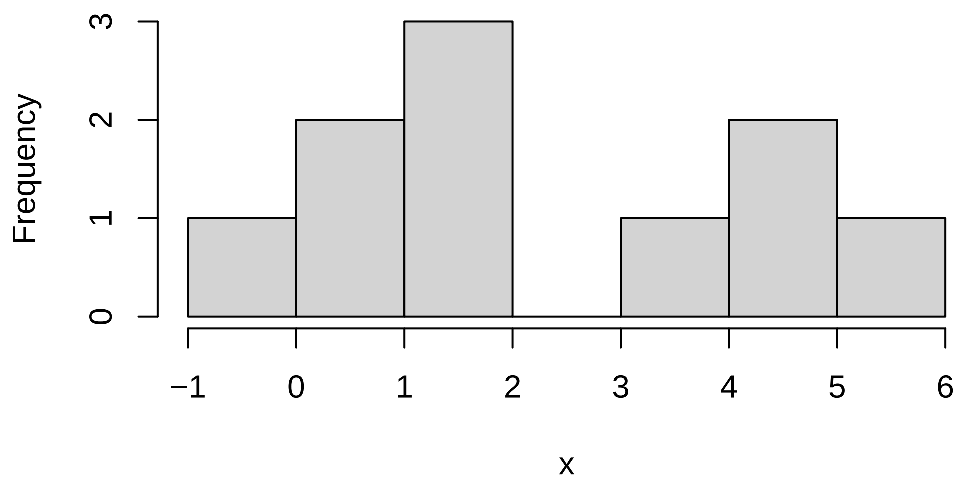
```r
# 2.
data('sleep')
```

```r
head(sleep)
```

```
  extra group ID
1   0.7     1  1
2  -1.6     1  2
3  -0.2     1  3
4  -1.2     1  4
5  -0.1     1  5
6   3.4     1  6
```

```r
# 3.
aggregate(extra ~ group, data = sleep, FUN = function(x){hist(x,main=names(x))})
```






```
Warning in format.data.frame(if (omit) x[seq_len(n0), , drop = FALSE] else x, :
corrupt data frame: columns will be truncated or padded with NAs
```

```
  group                    extra
1     1   -2, -1, 0, 1, 2, 3, 4
2     2 -1, 0, 1, 2, 3, 4, 5, 6
```

```
summary_stats = aggregate(extra ~ group, data = sleep,  FUN = summary)
print(summary_stats)
```

```
  group extra.Min. extra.1st Qu. extra.Median extra.Mean extra.3rd Qu.
1     1     -1.600        -0.175        0.350      0.750         1.700
2     2     -0.100         0.875        1.750      2.330         4.150
  extra.Max.
1      3.700
2      5.500
```

```
aggregate(extra ~ group, data = sleep, FUN = length)
```

```
  group extra
1     1    10
2     2    10
```

We will assume that the extra hours are normal from the histograms.

Recall then that the pooled standard deviation is $\hat{\sigma}_p = \sqrt{((n_x-1)\hat{\sigma}_x^2 + (n_y-1)\hat{\sigma}_y^2)\big/(n_x+n_y-2)}$ and the test statistic is:

$$T = \frac{\bar{X}-\bar{Y}}{\hat{\sigma}_p \times \sqrt{1/n_x + 1/n_y}}.$$

In addition, we have that $T \sim t_{n_x+n_y-2}$.

```r
# 5 and 6 - here these steps are the same, since we are only doing inference
t.test(sleep$extra[sleep$group==1],sleep$extra[sleep$group==2])
```

```

	Welch Two Sample t-test

data:  sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y 
     0.75      2.33 
```

```r
# 7 - we checked normality earlier, 8 is not applicable
```

Here, we fail to reject the null hypothesis, and there is not enough evidence to suggest that there is a different between the groups. Notice that the p-value is 0.08, which is moderately low, so there is some evidence of a difference between the groups.

1.2 Important course information and preparation tasks

1.2.1 Prerequisite review

If you have forgotten, you should review the following concepts:

- Sample vs. population, estimates vs. parameters, hypothesis testing and confidence intervals
- Normal theory, random variables, conditional variance and expectation.
- CLT, LLN
- Linear algebra: Matrix operations, inverse, transpose etc.

1.2.2 Software

Download RStudio/R. You can use python, but I'll use R in class. If you are not familiar with R, please follow this tutorial here.

1.2.3 Outline

The course will proceed as follows:

- Review
- Core linear regression concepts
- Special Cases
- Advanced

1.2.4 Homework tasks:

- Download and install RStudio and R Software
- Think of a relationship you would want to model, what is X? what is Y?
- Review prerequisites as stated above

2 Review material

2.1 Review of random variables

Recall that

Definition 2.1. A random variable X is a function which maps outcomes $\omega \in \Omega$ to the real numbers, i.e., $X: \Omega \to \mathbb{R}$.

> **Note**
>
> Note that the notation $f: A \to B$ means that f is a function whose domain is A and range is B. That is, f takes a value from A and outputs some value in B.

Generally, we will just write X, and ignore the fact that X is a function.

We can categorize a random variable X as follows:

- If $X: \Omega \to S$ where S is countable, then X is a *discrete random variable*
- We say X is a *continuous random variable* if $\Pr(X = r) = 0$ for all $r \in \mathbb{R}$.
- Otherwise, X is a *mixed random variable* (which we won't worry about in this course)

2.1.1 Discrete Random Variables

If $X: \Omega \to S$ where S is countable, then X is a discrete random variable. S can be finite, but can also be any infinite subset of the integers $\mathbb{Z}$. The distribution of X is given by its PMF, denoted by $f(x)$. For any $x \in S$, $f(x) = \Pr(X = x)$. (Note that '$\in$' means the word "in".)

We must have that:

- $\sum_{x \in S} f(x) = 1$, (This notation means summing over all the elements in S.)
- $\forall x \in S$, $0 \leq f(x) \leq 1$. (This notation means for all x in S, $0 \leq f(x) \leq 1$.)

Examples: Binomial random variables, Poisson random variables and Geometric random variables are all discrete random variables.

Exercise 2.1. What is the PMF of a Binomial random variable? Can two different random variables have the same PMF? Why or why not?

First: $\Pr(X = x) \binom{n}{x} p^x (1-p)^{n-x}$ Second: Yes. Two random variables can be different random variables, but have the same distribution.

2.1.2 Continuous Random Variables

We say X is a *continuous random variable* if $\Pr(X = r) = 0$ for all $r \in \mathbb{R}$. If $X : \Omega \to S$ and X is a continuous random variable, then S is typically the real numbers, denoted by $\mathbb{R}$, but can be any uncountable subset of $\mathbb{R}$. The distribution of X is given by the PDF $f(x)$. For any interval $(a, b) \subset\in S$, $\Pr(X \in (a, b)) = \int_a^b f(x)dx$.

We must have that:

- $\int_{-\infty}^{\infty} f(x)dx = 1$,
- $\forall x \in \mathbb{R}$, $f(x) \geq 0$.

Examples: Normal random variables, Chi-squared random variables, t random variables, Cauchy random variables, F random variables are all continuous random variables. Generally, we will focus on continuous random variables.

Exercise 2.2. What is the PMF of a Normal random variable?

$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$

2.1.3 Properties of Random Variables

Let X, X_1, X_2 be random variables.

Recall the important quantities $\mathrm{E}X$, $\mathrm{var}X$, $\mathrm{cov}(X_1, X_2)$, $\mathrm{corr}(X_1, X_2)$. Recall expectation:

Definition 2.2. The expectation of a random variable X is

$$\mathrm{E}X = \sum_{x \in S} x \Pr(X = x),$$

if X is discrete and is

$$\mathrm{E}X = \int_{-\infty}^{\infty} f(x)dx,$$

if X is continuous.

This is the “average’ ’ value of the random variable. Note that it is possible for it to be impossible for $X = \mathrm{E}X$. Try to come up with an example of this!

Definition 2.3. The variance of a random variable X is

$$\operatorname{Var}[X] = \mathrm{E}\left[|X - \mathrm{E}[X]|^2\right] = \sum_{x \in S} (x - \mathrm{E}[X])^2 \Pr(X = x),$$

if X is discrete and is

$$\operatorname{Var}[X] = \mathrm{E}\left[|X - \mathrm{E}[X]|^2\right] = \int_{-\infty}^{\infty} (x - \mathrm{E}[X])^2 f(x) dx,$$

if X is continuous.

The variance describes the variation of X about its mean. In other words, it describes on “average’ ’, how far is X from its mean.

Definition 2.4. The covariance between two random variables X and Y is

$$\operatorname{cov}[X, Y] = \mathrm{E}\left[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\right].$$

The covariance describes the unnormalised linear association between X and Y.

Definition 2.5. The correlation between two random variables X and Y is

$$\operatorname{corr}[X, Y] = \operatorname{cov}[X, Y] / \sqrt{\operatorname{Var}[X] \operatorname{Var}[Y]}.$$

The correlation describes the normalized linear association between X and Y.

Next, recall that for a random variable X, its cumulative distribution function (CDF) is given by $F_X(x) = \Pr(X \leq x)$. The joint CDF of X and Y is given by $F_{XY}(x, y) = \Pr(X \leq x, Y \leq y)$.

Lastly, for a vector of d random variables $\mathbf{X} = (X_1, \ldots, X_d)$, let its CDF by $F_{\mathbf{X}}(\mathbf{x}) = \Pr(X_1 \leq x_1, \ldots X_d \leq x_d)$, where here $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x} = (x_1, \ldots, x_d)$.

We next present the concept of independence of random variables. Let $F_{XY}(x, y)$ be the joint CDF of X and Y and let F_X and F_Y be the univariate CDFs of X and Y, respectively. For two random variables X and Y, we say that X and Y are independent if $F_{XY}(x, y) = F_X(x)F_Y(y)$. More generally, two vectors of random variables $\mathbf{X}$ and $\mathbf{Y}$ are independent if $F_{(\mathbf{X},\mathbf{Y})}(\mathbf{x}, \mathbf{y}) = F_{\mathbf{X}}(\mathbf{x})F_{\mathbf{Y}}(\mathbf{y})$, where $F_{(\mathbf{X},\mathbf{Y})}$ is the CDF of the vector $(\mathbf{X}, \mathbf{Y})$. A set of random variables $\{X_i\}_{i=1}^n$ are mutually independent if for any two mutually exclusive subsets of $\{X_i\}_{i=1}^n$ are also independent. Note that we write $X \perp Y$ if X is independent of Y.

We have that:

Theorem 2.1. *The following holds:*

- $X_1 \perp X_2 \implies E[X_1 X_2] = E[X_1]\, E[X_2]$
- $X_1 \perp X_2 \implies corr[X_1, X_2] = 0$
- $corr[X_1, X_2] = 0$ *does not imply* $X_1 \perp X_2$

Exercise 2.3. Prove Theorem 2.1 .

Let $X, X_1, X_2, \ldots X_n$ be random variables. Recall the linearity of expectation property:

Theorem 2.2. *For* $a, b \in \mathbb{R}$*, it holds that* $\mathrm{E}aX + b = a\mathrm{E}X + b$*.*

Exercise 2.4. Prove Theorem 2.2 .

As a corollary of Theorem 2.2 , we have that

- $\mathrm{E}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i \mathrm{E}\left[X_i\right]$
- $\mathrm{Var}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i^2 \mathrm{Var}\left[X_i\right] + \sum_{i \neq j} a_i a_j \mathrm{cov}\left[X_i, X_j\right]$
- $\mathrm{Var}\left[aX_1 + bX_2 + c\right] = a^2 \mathrm{var} X_1 + b^2 \mathrm{Var}\left[X_2\right] + 2ab\,\mathrm{cov}\left[X_1, X_2\right]$

Exercise 2.5. What happens to $\mathrm{Var}\left[aX_1 + bX_2 + c\right]$ when $\{X_i\}_{i=1}^{n}$ are mutually independent?

Exercise 2.6. Let $X_1, X_2, \ldots X_n$ be i.i.d. (independent and identically distributed) random variables with expectation (also known as mean) μ and variance σ^2. What is the expectation and variance of

$$\bar{X} = \sum_{i=1}^{n} X_i / n?.$$

2.1.4 Useful properties of normal and related random variables

Let

- $\mathcal{N}(\mu, \sigma^2)$ represent the normal distribution with mean μ and variance σ^2.
- χ_k^2 be the Chi-squared distribution with k degrees of freedom
- t_n be the student-t distribution with n degrees of freedom
- $F_{m,n}$ be the F distribution with m numerator degrees of freedom and n denominator degrees of freedom

We have the following results:

Theorem 2.3. *Suppose that* $X \sim \mathcal{N}(\mu, \sigma^2)$*, then*

- $Z = \frac{X-\mu}{\sigma} \sim \mathcal{N}(0,1)$
- $Z^2 \sim \chi^2_1$.

Let $[n] = \{1, \dots, n\}$. We also have that

Theorem 2.4.

- *If for $i \in [n]$ $Y_i \sim \chi^2_{k_i}$ and $Y_i \perp Y_j$ for $i \neq j$ then $\sum_{i=1}^n Y_i \sim \chi^2_{k_1+\dots+k_n}$.*
- *If $Y \sim \chi^2_k$ and $Y \perp Z$, then $Z/\sqrt{Y/k} \sim t_k$.*
- *If $Y_1 \sim \chi^2_{k_1}$, $Y_2 \sim \chi^2_{k_2}$ and $Y_1 \perp Y_2$ then $\frac{Y_1/k_1}{Y_2/k_2} \sim F_{k_1,k_2}$.*

Define

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^n (X_i - \bar{X})^2.$$

Theorem 2.5. *Suppose that $X_1, X_2, \dots X_n \sim \mathcal{N}(\mu, \sigma^2)$ and are independent, then $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0,1)$, $\bar{X} \perp \hat{\sigma}^2$, $(n-1)\hat{\sigma}^2/\sigma^2 \sim \chi^2_{n-1}$ and $\frac{\bar{X}-\mu}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$.*

2.1.5 Central Limit Theorem

CLT: If $X_1, X_2, \dots X_n$ are i.i.d. with mean μ and variance $\sigma^2 < \infty$, then $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \xrightarrow{\mathrm{d}} \mathcal{N}(0,1)$ as $n \to \infty$.

We have that in general, for large n, regardless of the distribution of the random variables, the sample mean is approximately normally distributed.

2.1.6 Homework stop 1

Review your material and complete the above exercises before continuing to the next section.

2.2 Review of introductory statistics

The followings are some concepts that you have learned from prerequisites, and/or we have reviewed in the last two lectures.

- Sample vs. Population
- Observation vs. Random variable
- Statistic vs. Parameter
- Estimate vs. Estimator

- Estimator is a random variable and estimate is a number calculated from data
- Mean and variance of random variable
- Relationships between Normal, t , χ^2, F etc.

2.2.1 Basic premise of statistics

The whole purpose of statistics is to learn something about a population using only a sample of units from that population. A **sample** is a smaller, typically randomly selected, subset of a population. A **population** is a collection of units which we would like to know something about. For example, we may collect a sample of hamburgers from McDonald's if we want to learn something about the population of McDonald's hamburgers.

In general, at least for this course, we assume that we have access to a sample of units from a given population. Furthermore, we assume that that sample is a **random sample**. Specifically, we assume that these units in the sample are realizations of random variables. In addition, we also assume that these random variables are mutually independent. For example, we could assume that our sample $X_1, \ldots, X_n$ is Normally distributed with some fixed mean μ and fixed variance σ^2. In this case, μ and σ^2 are unknown **parameters** of the population. A parameter of a population is some quantity that is a function of the distribution of our given sample. For instance, $\mathrm{E}[X_i] = \mu$. Generally, we are concerned with unknown population parameters, which are parts of the distribution that are unknown, and can only ever be estimated. For example, we may know our data is normal, but not know the mean parameter. In that case, we need to use an **estimate** of the parameter. We use a function of the data, typically called the estimator, say T, which produces the estimate, given by T computed at the sample we observed: $T(X_1, \ldots, X_n)$.

For example, to estimate μ, we typically use the sample mean. Here, the estimate is given by $\bar{X} = \sum_{i=1}^{n} X_i/n$. To be specific, the estimate is the value of $\bar{X}$ and the estimator T is the function that maps n real numbers to their mean. In general, estimates are used to give our 'best guess' at population parameters.

2.2.2 Confidence intervals

Recall from the previous section that our estimate of a parameter is only that, an estimate. In other words, it is not exactly equal to the population parameter. For instance, if we drew a different sample our estimate would change. A confidence interval is used to acknowledge this phenomenon in the reporting of our statistics. Its used to give a range of estimates that we might have obtained from any "regular" sample we might observe. It is ultimately used to quantify the error (sometimes called uncertainty) in our estimate.

Confidence intervals consist of a level, usually denoted by $(1 - \alpha)100\%$ and two end points. For example, you have learned confidence intervals for the population mean. When we say $(-1, 1)$ is 95% confidence interval for the population mean, what does this mean? Colloquially, it means that we expect the sample mean to be somewhere within $(-1, 1)$ with high confidence. Note that confidence intervals are computed from the data, which means also that for each new sample, we would get a different confidence interval. However, the population parameter never changes. Therefore, the interval is what is varying from sample to sample. This impacts the interpretation of a confidence interval.

Continuing our example, we have that the interval $(-1, 1)$ can be interpreted as: "if we drew many more samples, 95% of the **intervals** will contain the population parameter." We **do not** say that the parameter has a 95% chance of falling in (-1,1), since the parameter is not random, the interval end points are.

For example, we have the formula for a confidence interval for the population mean is given by: $\bar{X} \pm 1.96\hat{\sigma}$. Notice that it is based only on the data. Therefore, it will change if we drew a new sample.

To summarize this section, a confidence interval is used to quantify the uncertainty in our reported estimates. By uncertainty, we specifically mean the uncertainty resulting from the fact that we have only a sample of the population, and our estimate varies depending on the sample.

2.2.3 Hypothesis tests

Hypothesis tests are used to determine whether an effect is spurious or a real property of the population. A spurious effect is one that is specific to the sample we observed, and is not a real property of the population. For example, if the heights of males and female students are measured, and we observe that the sample mean of both male and females are equal, then this would be a spurious effect. We know that the population heights of males and females are substantially different. If we drew a new sample, we would likely observe something that mirrors the population reality (provided it is large enough).

Formally, a hypothesis test compares two competing beliefs about a population parameter, called the null and alternative hypothesis. For instance, we may wish to test whether the population heights of men is greater than women, vs. the heights being less than or equal to that of men.

We write this as follows: $H_0 \colon \mu_{men} \leq \mu_{women}$ vs. $H_a \colon \mu_{men} > \mu_{women}$.

The null hypothesis is usually chosen to be one such that if we make a mistake, the error is most serious. However, it is usually clear from the context.

In general, we compute a test statistic and its distribution **under the null hypothesis.** Then we compute how likely it was to see the observed test statistic we saw, if the null hypothesis was true. This likelihood is given by the **p-value**. If it was sufficiently unlikely (in other words, the p-value is less than the threshold α), then we reject the null hypothesis. Otherwise, we fail to reject the null hypothesis. If we fail to reject the null hypothesis then either the null hypothesis is true, it is not true, but there was not enough data collected to show the effect.

There are two types of errors we can make in a hypothesis test: Type 1 and Type 2 error. Type one error occurs when we reject the null hypothesis when it is true. Type two error occurs when we fail to reject the null hypothesis when the alternative is true.

Let's do an example.

Exercise 2.7. In a study about online dating, you are interested in determining the average age of individuals who use online dating platforms. You want to know whether the average age of online daters is significantly different from 30. You have a dataset of 40 ages of people using online dating platforms.

How would you answer this question?

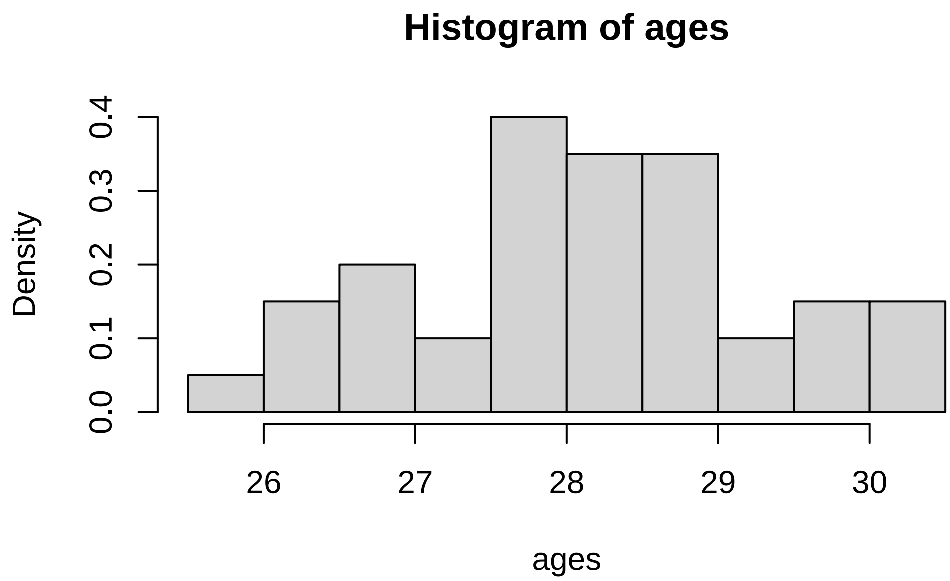
$$H_0: \mu = 30 \qquad vs. \qquad H_1: \mu \neq 30.$$

First, we can explore the data:

```
getwd()
```

```
[1] "C:/Users/12RAM/OneDrive - York University/Teaching/Courses/Math 3330 Regression/Math 33
```

```
ages=read.csv('C:\\Users\\12RAM\\OneDrive - York University\\Teaching\\Courses\\Math 3330


hist(ages,freq=F)
```



Now, assume that $X_1, \ldots, X_{40} \sim \mathcal{N}(\mu, \sigma^2)$, and independent. (We can justify normality with the histogram, or we could also invoke the CLT to get normality of the sample mean (not the data itself).) Therefore, we can do a one sample t-test. Recall that, under the null hypothesis, we have $\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$. This means that if $\left|\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}}\right| \geq t_{n-1,1-\alpha/2}$, then we reject the null hypothesis! Here, $t_{n-1,1-p}$ is the $(1-p)$th quantile of the t_{n-1} distribution. For large n and $p = 0.025$, this is roughly equal to 2.

Now, recall that

$$H_0 : \mu = 30 \qquad vs. \qquad H_1 : \mu \neq 30.$$

```
# Calculate the mean of the 'ages' data and assign it to xbar
xbar = mean(ages)
xbar  # Print the mean
```

```
[1] 28.16378
```

```
# Calculate the variance of the 'ages' data and assign it to ssq
ssq = var(ages)
ssq  # Print the variance
```

```
[1] 1.277377
```

```r
# Calculate the length (number of observations) of the 'ages' data and assign it to n
n = length(ages)
n  # Print the number of observations
```

```
[1] 40
```

```r
# Set the significance level
alpha = 0.05

# print test statistic
ts=(xbar -30)/(sqrt(ssq/n))
print(ts)
```

```
[1] -10.27532
```

```r
# compute p-value
pval=pt(ts,n-1)*2
pval
```

```
[1] 1.179149e-12
```

```r
# Perform a two-sided t-test to check if the mean of 'ages' is significantly different fro
# t.test() is the function for performing t-tests in R
test = t.test(ages, mu = 30, alternative = 'two.sided')
test  # Print the test result
```

```

	One Sample t-test

data:  ages
t = -10.275, df = 39, p-value = 1.179e-12
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 27.80232 28.52524
sample estimates:
mean of x 
 28.16378 
```

We have that $\left|\frac{\bar{X}-30}{\hat{\sigma}/\sqrt{n}}\right| = -10.28$. Using R, we get that the p-value is 1.179×10^{-12}.

Here the p-value measures how much evidence there is against the null hypothesis. If the p-value is very small, then this constitutes strong evidence against the null hypothesis. If the p-value is small, but closer to 0.05, then there is evidence against the null. If it is larger, but still small, say 0.1, then this is weak evidence against the null hypothesis. It is not helpful to throw it away if it is above 0.05, therefore we should not just take $\alpha = 0.05$. Choosing α depends on how serious a type 1 error is. If it is not that serious, we can take α larger. If it is very serious, we can take α smaller.

In this example, there is very strong evidence against the null hypothesis.

> **Note**
>
> Note also that we can use the confidence interval method with
>
> $$\bar{X} \pm t_{n-1,1-\alpha/2}\sqrt{\hat{\sigma}^2/n}.$$

```
# Alternative method to calculate the confidence interval
# ci will store the confidence interval values
ci = xbar + c(-1, 1) * qt(1 - alpha / 2, n - 1) * sqrt(ssq / n)
ci  # Print the confidence interval
```

```
[1] 27.80232 28.52524
```

> **Note**
>
> Moving beyond the one-sample testing problem, we might be interested in other population parameters, say $\theta \in \Theta$. Think Lecture 1: $\mathrm{E}[Y|X] = \beta_0 + X\beta_1$, we might want to estimate $\mathrm{E}[Y|X]$, which amounts to $\beta_0, \beta_1 \in \mathbb{R}$! In general, we may estimate θ by $\hat{\theta}$. Then we may compute the variance and distribution of $\hat{\theta}$. From there, we can make confidence intervals and conduct hypothesis tests etc.

Let's do another example:

Exercise 2.8. In a study about online dating, you are interested in determining if the average age of those who identify as men who use online dating platforms differs from those who identify as women. You have a dataset of 20 ages of each group using online dating platforms.

What is the population parameter of interest here? It is $\Delta = \mu_1 - \mu_2$, the difference in means between the two populations. Now, suppose that $X_1, \ldots, X_{20} \sim \mathcal{N}(\mu_1, \sigma^2)$ and $Y_1, \ldots, Y_{20} \sim$

$\mathcal{N}(\mu_2, \sigma^2)$, and are mutually independent. (We could also invoke the CLT instead of assuming normality.) We can estimate those parameters with **estimates.** For instance, $\bar{X}$, $\bar{Y}$,

$$\hat{\sigma}^2 = \frac{(n_1 - 1)\hat{\sigma}_1^2 + (n_2 - 1)\hat{\sigma}_2^2}{n_1 + n_2 - 2}.$$

Exercise 2.9. Suppose that $X_1, \dots, X_{20} \sim \mathcal{N}(\mu_1, \sigma^2)$ and $Y_1, \dots, Y_{20} \sim \mathcal{N}(\mu_2, \sigma^2)$, and are mutually independent. Compute $\text{Var}\left[\bar{X} - \bar{Y}\right]$.

Using independence of $\bar{X}$ and $\bar{Y}$ and the result of the Exercise 2.6 , we have that

$$\text{Var}\left[\bar{X} - \bar{Y}\right] = \text{Var}\left[\bar{X}\right] + \text{Var}\left[\bar{Y}\right] = \sigma_1^2/n_1 + \sigma_1^2/n_2.$$

First, we write down the null and alternative hypothesis:

$$H_0\colon \Delta = 0 \qquad vs.. \qquad H_1\colon \Delta \neq 0.$$

Here, we can do a two sample t-test.

Recall that the pooled variance is given by:

$$\hat{\sigma}_p^2 = \frac{(n_1 - 1)\hat{\sigma}_1^2 + (n_2 - 1)\hat{\sigma}_2}{(n_1 + n_2 - 2)}$$

We previously said that a multiple of a one sample standard deviation follows a Chi-squared distribution. It follows that $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 \sim \chi^2_{n_1 - 1}$ and $(n_2 - 1)\hat{\sigma}_2^2/\sigma^2 \sim \chi^2_{n_2 - 1}$. Using the theory from here, specifically, $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 + (n_2 - 1)\hat{\sigma}_2^2/\sigma^2$ is a sum of independent Chi-squared random variables, and so we have $(n_1 - 1)\hat{\sigma}_1^2/\sigma^2 + (n_2 - 1)\hat{\sigma}_2^2/\sigma^2 \sim \chi^2_{n_1 + n_2 - 2}$.

Again, using the theory from here, under the null hypothesis, we have that

$$\frac{\bar{X} - \bar{Y}}{\hat{\sigma}_p\sqrt{1/n_1 + 1/n_2}} = \frac{(\bar{X} - \bar{Y})/\sigma\sqrt{1/n_1 + 1/n_2}}{\hat{\sigma}_p/\sigma} \sim t_{n_1 + n_2 - 2}.$$

This follows from 3 facts, first, letting $Z = (\bar{X} - \bar{Y})/\sqrt{\text{Var}\left[\bar{X} - \bar{Y}\right]}$, note that $Z \sim \mathcal{N}(0, 1)$. We have that

$$Z = (\bar{X} - \bar{Y})/\sqrt{\text{Var}\left[\bar{X} - \bar{Y}\right]} = (\bar{X} - \bar{Y})/\sigma\sqrt{1/n_1 + 1/n_2}.$$

Next, we said earlier that $\bar{X}$ is independent of $\hat{\sigma}_1$ and $\bar{Y}$ is independent of $\hat{\sigma}_2$. Now, recall that if two random variables are independent, then any function of them is also independent. In other words, if X and Y are independent, then for real functions f and g, we have that $g(X)$

is independent of $f(Y)$. It follows that $\bar{X}$ is independent of $\hat{\sigma}_2$ and $\bar{Y}$ is independent of $\hat{\sigma}_1$. It follows that $\bar{X}-\bar{Y}$ is independent of $\hat{\sigma}_p$. Then,

$$\frac{(\bar{X}-\bar{Y})/\sigma\sqrt{1/n_1+1/n_2}}{\hat{\sigma}_p/\sigma}$$

is a ratio of a standard normal random variable and the square root of a Chi-squared random variable, divided by its degrees of freedom. Further, the numerator and denominator are independent. Therefore, the above quantity follows a t distribution with n_1+n_2-2 degrees of freedom.

This means that if $\left|\frac{\bar{X}-\bar{Y}}{\hat{\sigma}_p\sqrt{1/n_1+1/n_2}}\right| \geq t_{n_1+n_2-2,1-\alpha/2}$, then we reject the null hypothesis.

Let's execute the test in R:

```
# Normally, I will give you a dataset. Here I generate the data
set.seed(440)
female_ages=rnorm(20,28,4)
male_ages=rnorm(20,32,4)



# Check for equal variance
var(female_ages)
```

```
[1] 15.72805
```

```
var(male_ages)
```

```
[1] 26.22371
```

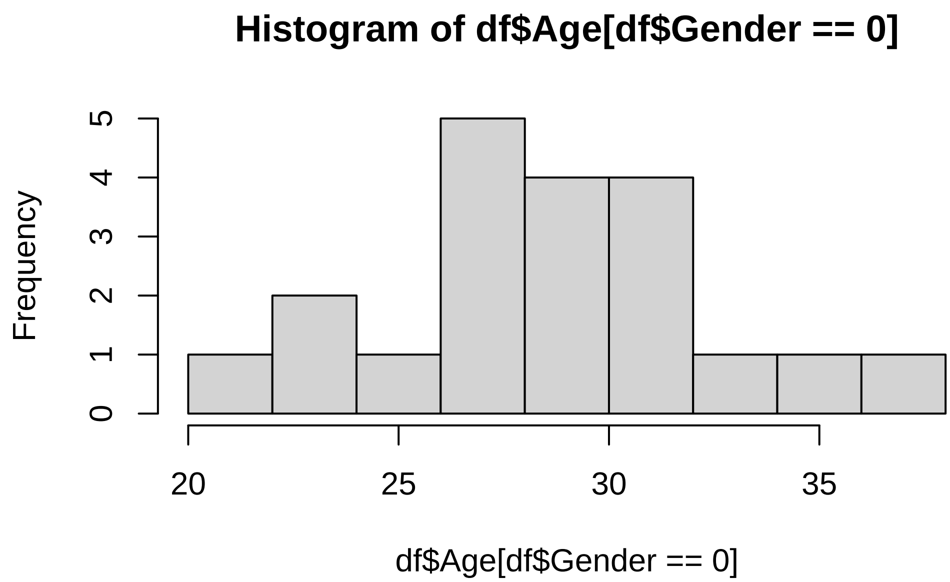
```
## Putting the data in a dataframe
cbind("Age"=c(female_ages,male_ages),"Gender"=rep(c(0,1),each=20))
```

```
          Age Gender
 [1,] 37.19809      0
 [2,] 20.69693      0
 [3,] 27.80284      0
 [4,] 27.69463      0
 [5,] 29.53143      0
```

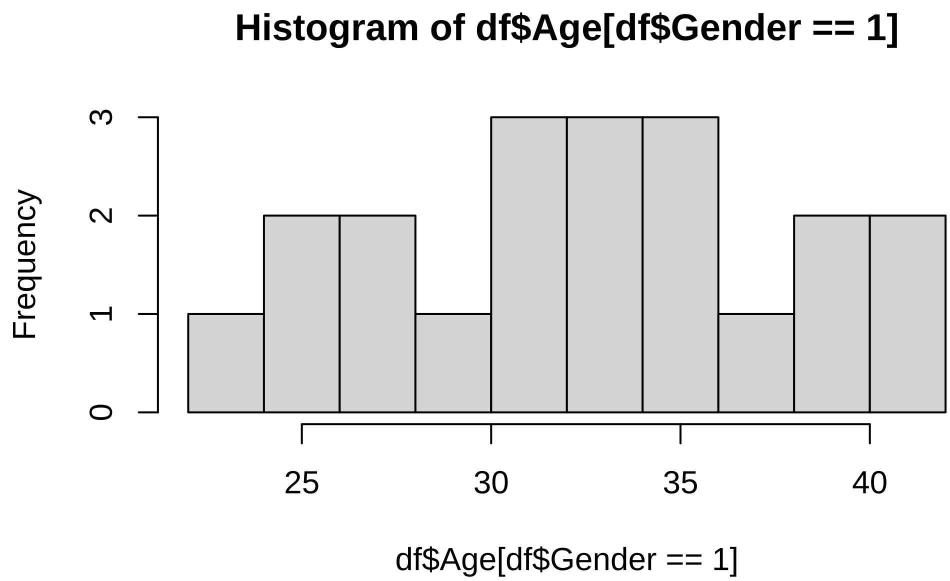
```
 [6,] 29.46190      0
 [7,] 30.41164      0
 [8,] 33.27790      0
 [9,] 22.65974      0
[10,] 30.73540      0
[11,] 34.08564      0
[12,] 27.58077      0
[13,] 23.26108      0
[14,] 30.94523      0
[15,] 31.52404      0
[16,] 29.13246      0
[17,] 26.95470      0
[18,] 24.80749      0
[19,] 28.60051      0
[20,] 26.76294      0
[21,] 25.94775      1
[22,] 40.16080      1
[23,] 25.58905      1
[24,] 32.16780      1
[25,] 29.87934      1
[26,] 35.46593      1
[27,] 35.71651      1
[28,] 37.76510      1
[29,] 27.23068      1
[30,] 33.41994      1
[31,] 40.43822      1
[32,] 31.04841      1
[33,] 32.66165      1
[34,] 38.28678      1
[35,] 34.72411      1
[36,] 39.57994      1
[37,] 26.85585      1
[38,] 31.87533      1
[39,] 23.71793      1
[40,] 30.54803      1
```

```r
df=data.frame(cbind("Age"=c(female_ages,male_ages),"Gender"=rep(c(0,1),each=20)))

#exploring the data
#hist(x) creates a histogram of the vector x
```

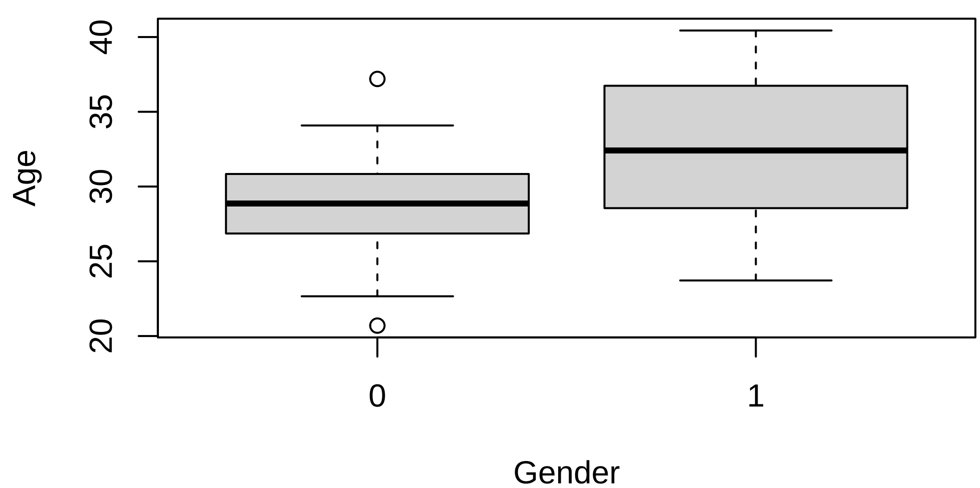
```r
hist(df$Age[df$Gender==0])
```




```r
hist(df$Age[df$Gender==1])
```




```
#boxplot creates boxplots of Age against gender
boxplot(Age~Gender, df)
```



```r
test=t.test(Age~Gender,data=df,var.equal=TRUE)
test
```

```

	Two Sample t-test

data:  Age by Gender
t = -2.7603, df = 38, p-value = 0.008841
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -6.929630 -1.065749
sample estimates:
mean in group 0 mean in group 1 
       28.65627        32.65396 
```

```r
#Interpret the P value, and CI, what are we going to say to a stakeholder?

#e.g.
test$estimate
```

```
mean in group 0 mean in group 1 
       28.65627        32.65396 
```

> **Note**
>
> Note also that we can use the confidence interval method, meaning that if if 0 is in the interval:
> $$\hat{\Delta} \pm t_{n_1+n_2-2,1-\alpha/2}\hat{\sigma}_p\sqrt{1/n_1 + 1/n_2},$$
> then we fail to reject the null hypothesis.

2.2.4 Homework stop 2

Exercise 2.10. IBM Human Resources (HR) department is evaluating job applicants from York University.
They are interested to know if the 2020 ITEC graduating class has an average GPA higher than 6 (i.e. average GPA higher than "B''). They collected the GPA of 25 ITEC students graduated in 2020.

4.92	4.79	6.76	5.64	6.12	7.37	6.45	6.31	6.68
6.30	4.91	6.95	5.87	6.18	6.60	6.71	6.69	5.62
6.40	5.51	6.44	6.13	8.55	7.94	4.78	-	-

Tip

Use chatGPT to convert the above table to an R vector, so you don't have to waste time!

- For the one sample testing problem, i.e., you have a sample of n normal random variables, with unknown mean and variance and you want to test whether $H_0 : \mu = 0$ vs. $H_0 : \mu \neq 0$, show that $\frac{\bar{X}}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$ under the null hypothesis.
- What is the distribution of each of the following: $\bar{X}, \bar{Y}, \hat{\sigma}$ under the assumption of normal data with unknown mean and variance?

Compare and contrast the following concepts. That is, define them and explain the difference between them.

- Sample vs. Population
- Observation vs. Random variable
- Statistic vs. Parameter
- Estimate vs. Estimator

2.3 Review of matrices and linear algebra

Recall that

Definition 2.6. An $(n \times m)$ matrix A takes the form

$$
\begin{aligned}
A &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \\
&= ((\ a_{ij}\ )) \qquad i = 1, \dots, n, \quad j = 1, \dots, m
\end{aligned}
$$

and a_{ij} is the element in the i^{th} row and j^{th} column of the matrix A

We also define the following:

- An $(n \times 1)$ matrix is also known as a n dimensional column vector. Note: in this course, a vector means a column vector.

- A $(1 \times m)$ matrix is also known as a m dimensional row vector
- The n dimensional one vector, 1_n, (sometimes the subscript n is suppressed when the dimension is obvious), is an n dimensional column vector with all entries being 1.
- The $(n \times n)$ identity matrix, I_n, is the $(n \times n)$ matrix with diagonal entries set equal to 1 and the off diagonal entries set equal to 0

Throughout this section, we will use the following matrices to demonstrate the numerical calculations:

$$U = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & -2 \end{pmatrix},\ V = \begin{pmatrix} 2 & 4 \\ 1 & -2 \\ -1 & 0 \end{pmatrix},\ k = 4$$

2.3.1 Matrix properties

First, we define the transpose of a matrix:

Definition 2.7. Let $A = ((\ a_{ij}\ ))$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, is an $(n \times m)$ matrix. Then $A^\top = A$ transpose $= ((\ a_{ji}\ ))$ for $j = 1, \ldots, m$ and $i = 1, \ldots, n$, and $A^\top$ is an $(m \times n)$ matrix.

When we transpose a matrix A, the rows of A becomes the columns of $A^\top$ and the columns of A becomes the rows of $A^\top$.

Example 2.1. Using our example matrices, we have that

$$U^\top = \begin{pmatrix} 1 & -1 \\ 2 & 4 \\ 3 & -2 \end{pmatrix},\ V^\top = \begin{pmatrix} 2 & 1 & -1 \\ 4 & -2 & 0 \end{pmatrix}$$

Definition 2.8. Let $A = ((\ a_{ij}\ ))$ and $B = ((\ b_{ij}\ ))$ be two $(n \times m)$ matrices. Then

$$A \pm B = ((\ a_{ij} \pm\ b_{ij}\ )).$$

Addition and subtraction of matrices required the matrices to have the same dimension.

Example 2.2. Using our example matrices, we have that: $U + V$ is undefined because they are not of the same dimension, and

$$U + V^\top = \begin{pmatrix} 1+2 & 2+1 & 3+(-1) \\ (-1)+4 & 4+(-2) & (-2)+0 \end{pmatrix} = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 2 & -2 \end{pmatrix}$$

Definition 2.9. Let $A = ((\ a_{ij}\ ))$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, is an $(n \times m)$ matrix and k is a constant. Then

$$kA = ((\ ka_{ij}\ )) = Ak,$$

i.e. each element of the matrix A is multiplied by k.

Example 2.3. Using our example matrices, we have that:

$$kU^\top = 4 \begin{pmatrix} 1 & -1 \\ 2 & 4 \\ 3 & -2 \end{pmatrix} = \begin{pmatrix} 4(1) & 4(-1) \\ 4(2) & 4(4) \\ 4(3) & 4(-2) \end{pmatrix} = \begin{pmatrix} 4 & -4 \\ 8 & 8 \\ 12 & -2 \end{pmatrix}$$

Definition 2.10. Let A and B be two matrices. Then A multiplied by B, AB, is defined only if (number of columns of A) = (number of rows of B).
The product is a ((number of rows of A) $\times$ (number of columns of B)) matrix.

More precisely, let $A = ((\ a_{ij}\ ))$ be an $(n \times m)$ matrix and $B = ((\ b_{ij}\ ))$ be an $(m \times p)$ matrix. Then $C = AB = ((\ c_{ij}\ ))$ is an $(n \times p)$ matrix with

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj}$$

> **Note**
>
> In matrix algebra, AB is not necessarily equal to BA.

Example 2.4. Using our example matrices, we have that:

$$\begin{aligned} UV &= \begin{pmatrix} 1 & 2 & 3 \\ -1 & 4 & -2 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 1 & -2 \\ -1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1(2) + 2(1) + 3(-1) & 1(4) + 2(-2) + 3(0) \\ (-1)(2) + 4(1) + (-2)(-1) & (-1)(4) + 4(-2) + (-2)(0) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 4 & -12 \end{pmatrix} \end{aligned}$$

Assume all the matrix multiplication works. Let I_n be an $(n \times n)$ identity matrix. Then

$$AI_n = A, \quad \textit{and} \quad I_nB = B.$$

Definition 2.11. Let A be an $(n \times n)$ matrix. The inverse of A, A^{-1}, if exists satisfies

$$AA^{-1} = A^{-1}A = I_n$$

and if A^{-1} does not exist, then A is a singular matrix.

❗ Important

From your linear algebra course, a prerequisite, you have learned the condition(s) for the existence of an inverse, The Invertible Matrix Theorem and you have learned how to obtain an inverse. You should review them.
Specifically, you should know how to obtain inverse of any diagonal matrix and any (2×2) non-singular matrix, i.e.,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Example 2.5. Using our example matrices, let

$$W = UV = \begin{pmatrix} 1 & 0 \\ 4 & -12 \end{pmatrix}$$

Then

$$W^{-1} = \frac{1}{1(-12)-0(4)}\begin{pmatrix} -12 & 0 \\ -4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1/3 & -1/12 \end{pmatrix}.$$

You can verify that $WW^{-1} = W^{-1}W = I_2$.

2.3.2 Important identities

Lastly, we introduce some important identities:

$$X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \quad \textit{and} \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Then

$$X^\top X = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix}, \quad \text{and} \quad X^\top y = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}.$$

Also $\bar{y} = \frac{1}{n}1^\top y$ and $\sum_{i=1}^n y_i = n\bar{y}$ and, finally $\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n y_i^2 - n\bar{y}^2$. These are useful identities that we will use throughout this course.

Exercise 2.11. Prove the previously introduced identities.

Lastly, we recall an important application of matrices. An application of matrices: Suppose that we want to solve for x_1, x_2, x_3 where they satisfy the following set of linear equations:

$$
\begin{aligned}
2x_1 + 3x_2 - 4x_3 &= 0 \\
-x_1 + 4x_2 &= -1 \\
5x_1 + x_2 - 2x_3 &= 4
\end{aligned}
$$

We can set it up in matrix form as follows:

$$
\begin{pmatrix} 2 & 3 & -4 \\ -1 & 4 & 0 \\ 5 & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 4 \end{pmatrix}
$$

Or it can be presented as $Ax = b$. If A is not a singular matrix, then $x = A^{-1}b$. Since $det(A) = 62$, it is not a singular matrix. Solving the above equation this using $x = A^{-1}b$ yields that $x = (1, 0, 0.5)^\top$. Keep this in mind, we will see it return in the next chapter.

Lastly - I will remind you of two definitions:

Definition 2.12. A set of vectors $v_1, \ldots, v_d$ is linearly independent if the only d-dimensional set of scalars $a_1, \ldots, a_d \in \mathbb{R}$ such that $\sum_{i=1}^{d} a_i v_i = 0$ implies is the d-dimensional 0 vector.

Definition 2.13. A $d \times d$ symmetric matrix A is positive-definite if for all $x \in \mathbb{R}^d \setminus \{0\}$ it holds that $x^\top A x > 0$.

2.3.3 Homework stop 3

Exercise 2.12. Let

$$
W = \begin{pmatrix} 3 & 2 \\ -4 & 6 \end{pmatrix}
$$

and $x = (2, 1)^\top$. Compute W^{-1}, $xx^\top$ and $x^\top W$. Verify that $WW^{-1} = W^{-1}W = I_2$.

- Prove each of the important identities.
- Verify $X^\top A = (A^\top X)^\top$.
- What is the rank of a matrix? Is a matrix's rank related to whether or not a matrix is invertible? Why?
- Define a positive definite matrix. When is $X^\top X$ positive definite?

2.4 Review Random Vectors

2.4.1 Definition of random vectors

Definition 2.14. Let $Y_1, \cdots, Y_n$ be random variables. Then

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}$$

is an n-dimensional random vector.

Similar to a random variable, a random vector also comes with a probability mass function (if all the Y_i are discrete) or a probability density function (if all the Y_i are continuous), or a "mixture" distribution (if some Y_i are discrete and others are continuous). In general, a random vector is drawn from a multivariate distribution, defined by the PMF or PDF. Just as before, the PMF and PDF range is non-negative, the PMF sums to 1 over all outcomes, and the PDF integrates to 1 over $\mathbb{R}^n$. One discrete multivariate distribution you have learned in 1131 is the Multinomial distribution. We will learn about the multivariate normal distribution soon.

2.4.2 Expected Value and Covariance

Definition 2.15. Let Y be an n-dimensional random vector, then the mean (expected value) of Y is defined as

$$\mathrm{E}(Y) = \begin{pmatrix} \mathrm{E}(Y_1) \\ \vdots \\ \mathrm{E}(Y_n) \end{pmatrix} = \mu$$

and the covariance matrix of Y is defined as

$$\mathrm{cov}\left[Y\right] = \mathrm{E}[(Y-\mu)(Y-\mu)^\top] = ((\mathrm{cov}\left[Y_i, Y_j\right])) = \Sigma.$$

Sometimes $\mathrm{cov}\left[Y\right]$ is written as $\mathrm{Var}\left[Y\right]$.

The following are some facts about Σ:

Σ is an $n \times n$ matrix with the diagonal elements being the variances, $\mathrm{Var}\left[Y_i\right]$ for $i = 1, \dots, n$, and the off-diagonal elements being the covariances, $\mathrm{cov}\left[Y_i, Y_j\right]$ for $i, j = 1, \dots, n$ and $i \neq j$. Σ is a symmetric, non-negative definite matrix. In this course, we further restrict it to be a positive definite matrix. Σ is referred to as the **covariance matrix**.

2.4.3 Properties of expected value and covariance

Let $X, Y \in \mathbb{R}^d$ be random vectors with $A \in \mathbb{R}^d$ and $B \in \mathbb{R}^{n \times d}$ be matrices. It holds that

- $\mathrm{E}(X+Y) = \mathrm{E}(X) + \mathrm{E}(Y)$
- $\mathrm{E}(A+BY) = A + B\mathrm{E}(Y)$
- $\mathrm{cov}\left[A+BY\right] = B\mathrm{cov}\left[Y\right]B^\top$.

Exercise 2.13. Let $Y = (Y_1, \dots, Y_n)^\top$ be a random vector, where Y_i are i.i.d. random variables with mean μ and variance σ^2. What are the mean and covariance of Y? Use properties of random vectors to compute the mean and variance of the sample mean.

First, $\mathrm{E}(Y) = \mu 1$ and $\mathrm{cov}\left[Y\right] = \sigma^2 I$. Note that $\bar{Y} = (Y_1 + \dots + Y_n)/n = \frac{1}{n} 1^\top Y$. Now, we have

$$\mathrm{E}(\bar{Y}) = \mathrm{E}\left(\frac{1}{n}1^\top Y\right) = \frac{1}{n}\left(1^\top \mathrm{E}(Y)\right) = \frac{1}{n}(n\mu) = \mu$$

and,

$$\begin{aligned}
\mathrm{cov}\left[\bar{Y}\right] &= \mathrm{cov}\left[\frac{1}{n}1^\top Y\right] \\
&= \left(\frac{1}{n}\right)^2 \left(1^\top \mathrm{cov}\left[Y\right] 1\right) \\
&= \left(\frac{1}{n}\right)^2 (n\sigma^2) = \frac{\sigma^2}{n}.
\end{aligned}$$

2.4.4 Multivariate normal distribution

We say that a random vector $X \sim \mathcal{N}_d(\mu, \Sigma)$ follows a multivariate normal distribution if X has PDF:

$$\phi(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{d/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)\right\}.$$

If $X \sim \mathcal{N}_d(\mu, \Sigma)$ and $c \in \mathbb{R}^d$, $A \in \mathbb{R}^{m \times d}$ then:

- $AX \sim \mathcal{N}(A\mu, A\Sigma A^\top)$.
- $c^\top X \sim \mathcal{N}(c^\top \mu, c^\top \Sigma c)$.
- Any conditional distribution for a subset of the variables conditional on another subset of variables is a multivariate distribution.

Using random vectors is a simple way of deriving lots of equations for this course. Working with vectors also allows those who are "geometrically gifted" to view the whole regression concepts geometrically! If not, not to worry!

2.4.5 Homework stop 4

Exercise 2.14. For a (full-rank) non-random matrix $X \in \mathbb{R}^{n\times p}$ with $n > p$, and random vector $Y \in \mathbb{R}^{n\times 1}$ with mean μ and covariance Σ, compute the following:

- Expected value and covariance matrix of $(X^\top X)^{-1}X^\top Y$
- Expected value of $Y^\top Y$
- Expected value and covariance matrix of $X^\top X$
- Expected value and covariance matrix of $X(X^\top X)^{-1}X^\top Y$

3 Linear Regression

3.1 Basics of linear regression

By the end of this section, you should be able to say what the linear and normal linear regression models are. As well as what it means to assume either of these models.

3.1.1 The linear regression model

Consider the following example.

Example 3.1. It is difficult to accurately determine a person's body fat percentage without immersing them in water. However, we can easily obtain the weight of a person. A researcher would like to know if weight and body fat percentage are related? If so, for a given weight, can the person's body fat percentage be predicted? If so, how accurate is the prediction? This researcher collected the following data:

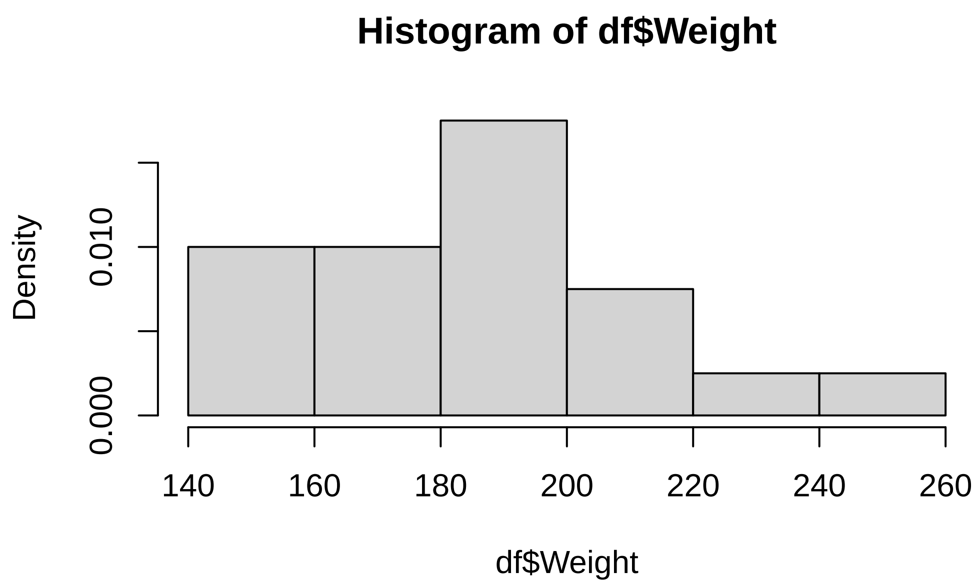
Individual	1	2	3	4	5	6	7	8	9	10
Weight (lb)	175	181	200	159	196	192	205	173	187	188
Body Fat (%)	6	21	15	6	22	31	32	21	25	30

Individual	11	12	13	14	15	16	17	18	19	20
Weight (lb)	188	240	175	168	246	160	215	159	146	219
Body Fat (%)	10	20	22	9	38	10	27	12	10	28

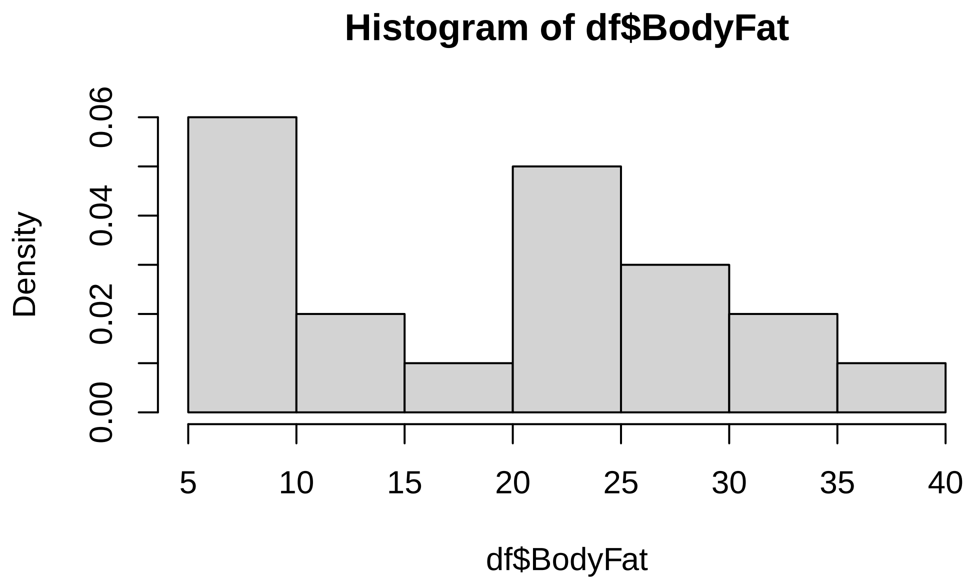
How can we (as statisticians / data scientists) answer the questions raised by the researcher?

The first thing we might do is explore the data:

```r
#################### Exploratory analysis

# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))

# make some histograms
hist(df$Weight,freq=F)
```



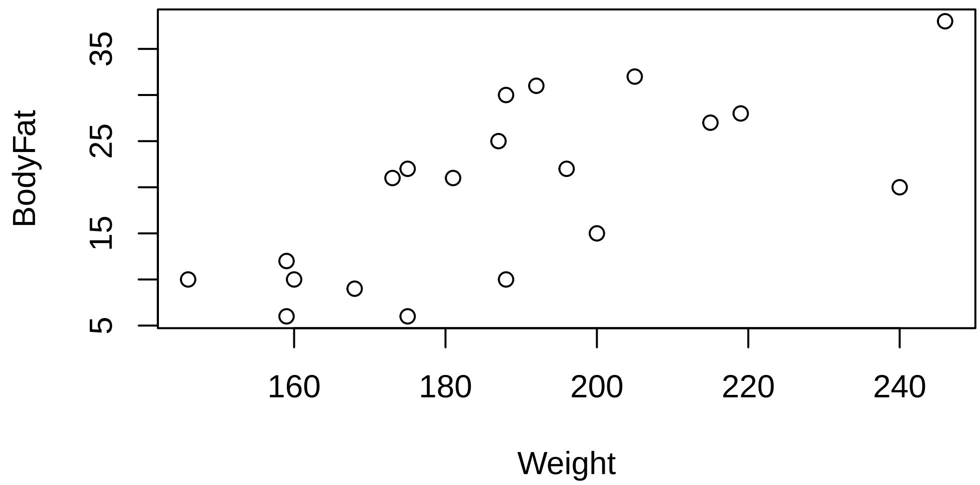

```r
hist(df$BodyFat,freq=F)
```




```r
# print summary statistics
summary(df)
```

```
     Weight         BodyFat     
 Min.   :146.0   Min.   : 6.00  
 1st Qu.:171.8   1st Qu.:10.00  
 Median :187.5   Median :21.00  
 Mean   :188.6   Mean   :19.75  
 3rd Qu.:201.2   3rd Qu.:27.25  
 Max.   :246.0   Max.   :38.00  
```

```r
# There seems to be some relationship here
plot(df)
```




```r
# Here is the correlation matrix, notice it is high!
cor(df)
```

```
           Weight   BodyFat
Weight  1.0000000 0.6966328
BodyFat 0.6966328 1.0000000
```

We have observed that there is a relatively strong linear relationship between these two variables. What next? We might ask, what is this relationship precisely?

In particular, note that we have observed a sample of vectors $(Y_1, X_1), \dots, (Y_1, X_n)$. Now, we want to say something about the relationship between X and Y in general. One way to do that is to suppose at the **population** level that

$$\mathrm{E}\left[Y|X\right] = f(X).$$

That is, on average, Y is equal to $f(X)$. One way to do that is to assume that $Y|X = f(X) + \epsilon$, where ϵ is a random variable that satisfies $\mathrm{E}\left[\epsilon\right] = 0$. This assumption means that, for each Y_i, given X_i, we have that $Y_i = f(X_i) + \epsilon_i$. Note that we do not observe ϵ_i, but we can assume it exists. We can read this as Y_i is equal to $f(X_i)$, plus some random, individual error ϵ_i. The next step is to use the data to determine f.

Using the data analysis steps from the Introduction we can write out the first few steps:

- Question about a population: "How can we use weight to determine body fat percentage?' '
- Data: $(Y_1, X_1), \dots, (Y_{20}, X_{20})$, (Y_i, X_i) are the body fat percentage and weight of individual $i \in [20]$.

We have explored the data with graphs and summary statistics. Now, we have posited the model $Y|X = f(X) + \epsilon$. Letting f be any function is too general. In fact, we can use the data to learn more about what f might be. Recall that earlier, we saw the scatter plot, where it looked like there was a linear relationship, (with some error), between Y and X. (We can draw a straight line through the middle of the data.)

Let's make some assumptions that make the statistical analysis easier:

1. Assume that $\forall i \in [20]$, it holds that

$$Y_i|X_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

   This means that we assume that f is a line.

2. Next, we assume $\forall i \in [20]$, $\mathrm{E}[\epsilon_i] = 0$ and $\mathrm{Var}[\epsilon_i] = \sigma^2$. That is, the random error have the same mean and variance for each individual. In addition, the random errors average to 0.

3. We also assume that the individuals' Body fat percentage, weights and random errors are independent, that is, $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [20]$.

This is the **simple linear regression model**. That is, the simple linear regression model is the set of assumptions 1-3 given above.

It is often also assumed:

4. $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$,

but not always. Including the normality assumption is known as the **simple normal linear regression model.**

In general, a model is a set of assumptions about a population. The particular set of assumptions 1-3 is the simple linear regression model.

The following is some terminology used in regression analysis:

- Here, Y_i is the **response variable**, also known as the dependent variable, or the outcome variable.
- Here, X_i is the **covariate**, also known as the explanatory variable, or the independent variable.

Given a “question about a population” which involves regression, you should immediately identify the response variable and the covariates.

Now, how can we interpret this model? That is, what does it mean to assume this model?

First, observe that we assume that $\mathrm{E}[Y|X]$ is a line. This means there is a linear relationship between the average body fat percentage and weight.

Next, observe that for any individual, their actual body fat percentage is given by $Y = \mathrm{E}[Y|X] + \epsilon_i = \beta_0 + \beta_1 X_i + \epsilon_i$. Therefore, their body fat percentage will not fall exactly on the line $\beta_0 + \beta_1 X_i$. Rather, it will fall above or below the line, depending on ϵ_i. Furthermore, if we assume that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, then we know from the properties of the Normal distribution that this random error will not exceed 2σ with high probability. Therefore, most of the time, an individual's body fat percentage will fall within 2σ of the line.

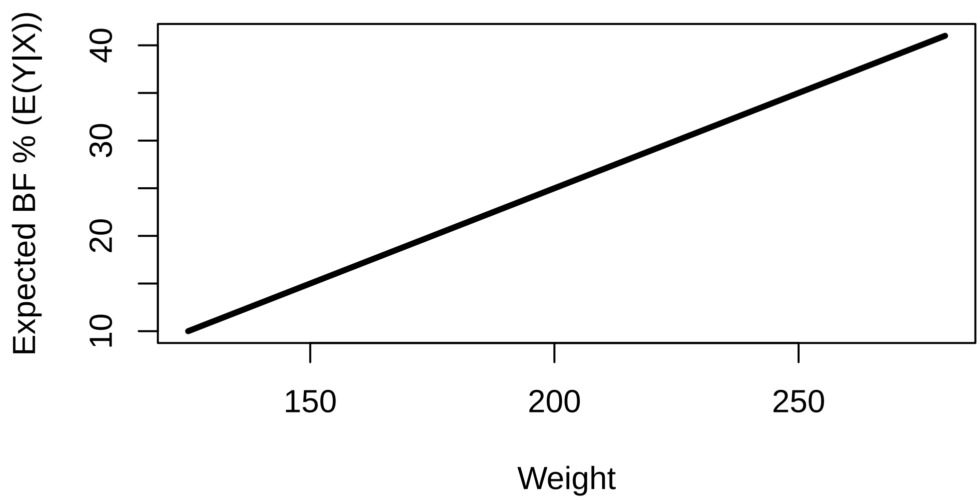
Third, notice that this quantity, 2σ, does not depend on X. That is, for any weight, we still expect an individual's body fat percentage to be within 2σ of the line, regardless of the value of weight.

Fourth, if we knew β_0, β_1, then given someone's weight, we could try to predict their body fat percentage given their weight. That is, we could calculate the expected body fat $\mathrm{E}[Y|X]$. There would still be their individual random error ϵ, so we would not be able to predict it exactly. However, if σ^2 isn't too big, then we could produce an accurate prediction.

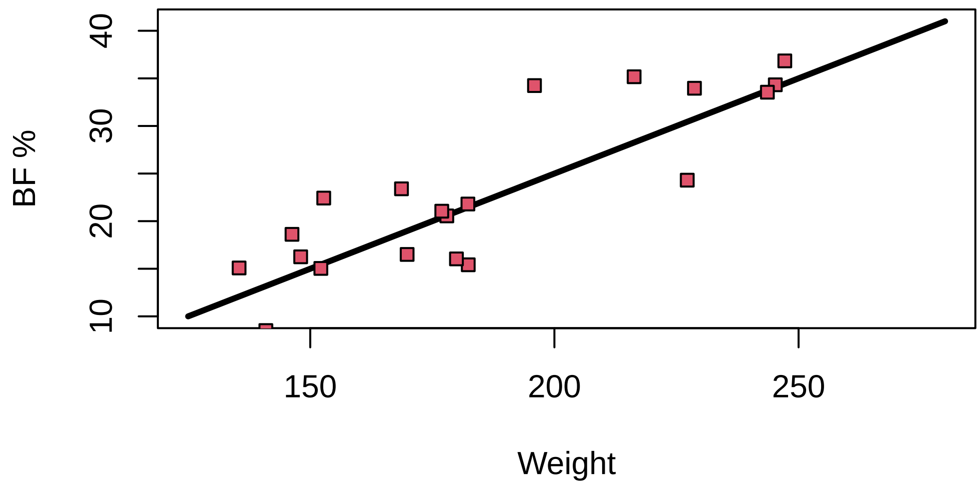
Therefore, if the model assumptions are correct, we assume there exists some line, around which the body fat percentages are scattered uniformly.

Next, we will simulate data from the normal simple linear regression model to gain a better understanding of this model. Suppose that $\beta_0 = -15$, $\beta_1 = .2$ and $\sigma = 5$. Then we would observe the following.

```
############################# Simulation
set.seed(3252)

# Suppose that beta_0=-15 and beta_1=0.2 and sigma=5,
# then we would have that the mean function E(Y|X) is given by the following line:
curve(-15+.2*x,125,280,lwd=3,xlab="Weight",ylab="Expected BF % (E(Y|X))")
```




```
# Next, let's simulate some body weights from the uniform distribution
Weight2=runif(20,135,250)

# Then, we can simulate the population body fat percentages according to the model as foll

# Simulating 20 values of the random error,
epsilons=rnorm(n=20,mean=0,sd=5)

# Computing the simulated Body fat percentages:
Bfs=-15+.2*Weight2+epsilons

# Plot the simulated values, and the mean function
curve(-15+.2*x,125,280,lwd=3,xlab="Weight",ylab="BF %")
points(Weight2,Bfs,pch=22,bg=2)
```



Notice how the data are scatted around the line uniformly? This is what data from a simple linear regression model looks like. Try changing the value in `set.seed()` and re-running the code. Notice how the data changes, but it is always scatted around the line uniformly? This is what we expect to see if the data follow a simple linear regression model.

Notice how the data simulated from our model appears similar to the body fat percentage and weights data we observed? That means this model (set of assumptions) is a good fit for our data.

> **Caution**
>
> In this model, and in regression in general, the response Y is not exactly equal to some function of X given by $f(X)$. The model assumes that **on average** $Y = f(X)$. Therefore, knowing someones "X" value will not exactly give us their Y value, but it would give us a good guess at it. The error ϵ is used to model the fact that someones "X" value will not exactly give us their Y value. Notice above how the actual points are scatted around the line, and not exactly equal to it! This is due to the errors ϵ.

3.1.2 The multiple linear regression model

But what about matrices? Why did we study matrices then? We can write the regression model in terms of matrices and vectors, to make it more compact.

Now, recall

$$Y_i|X_i = \beta_0 + \beta_1 X_i + \epsilon_i,$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. It is more convenient mathematically to let $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$,

$$\mathbf{X} = \begin{bmatrix} 1 & X_1 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} = \left[1_n \mid (X_1, \dots, X_n)^\top\right],$$

$\beta = (\beta_0, \beta_1)^\top$ and $\quad = (\epsilon_1, \dots, \epsilon_n)^\top$. Then we can write

$$\mathbf{Y}|\mathbf{X} = \mathbf{X}\beta + \quad .$$

Often, we overload the notation Y, and use Y instead of $\mathbf{Y}$, and X instead of $\mathbf{X}$.

This form allows us to go beyond one explanatory variable very easily! Just add one column to X and one entry to β for each new variable. Observe the following model:

$$Y_i|(X_{i1}, \dots, X_{ik}) = \beta_0 + \beta_1 X_{i1} + \dots + \beta_k X_{ik} + \epsilon_i,$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$. This is known as the **multiple linear regression model** (MLR), or just the linear regression model for short. We can write this model in the same for as above: Let

$$\mathbf{X} = \begin{bmatrix} 1 & X_{11} & X_{1k} \\ \vdots & \vdots & \vdots \\ 1 & X_{n1} \dots & X_{nk} \end{bmatrix},$$

and $\beta = (\beta_0, \dots, \beta_k)^\top$. Then we can write the MLR as

$$\mathbf{Y}|\mathbf{X} = \mathbf{X}\beta + \quad ,$$

where $\mathrm{E}\,[(]\,\epsilon) = 0$ and $\mathrm{Var}\,[\epsilon] = \sigma^2 I$. Notice how compact this is! As in the simple case, there is also the **normal MLR**, which further assumes that $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$.

We can then study the mathematical properties of

$$Y|X = X\beta + \epsilon$$

for general but fixed k, under the normal or vanilla MLR, which will cover many models.

3.1.3 Homework stop 1

Exercise 3.1. Try adjusting the parameters β_0, β_1, σ in the simulation, what happens to the data? What happens to the line?

Exercise 3.2. Is β an estimate or a population parameter? Why?

Exercise 3.3. Come up with another possible form of f that is not linear. Adjust the simulation to include this form of f.

Exercise 3.4. Write down the assumptions of the MLR and the normal MLR. What is the difference between the two models?

3.2 Least Squares

Now that we have settled on a model for the population, the next step is to use the data to estimate the model parameters. In particular, we need to estimate β. That will allow us to estimated $\mathrm{E}[Y|X]$ for any value of X.

Recall that we want to study the **population** model:

$$Y|X = X\beta + \epsilon.$$

3.2.1 Notation

For the model $Y|X = X\beta + \epsilon$, we have

- $Y \in \mathbb{R}^n$ is the response variable (a continuous random variable).
- $X \in \mathbb{R}^{n\times p}$ is the covariate matrix (Note that the first column is often 1_n).
- $X_i \in \mathbb{R}^p$ is the i^{th} observed explanatory variable ($i = 1, \dots, n$) (not a random variable, in the sense that we condition on it).
- $\beta \in \mathbb{R}^{p\times 1}$ is the coefficient vector .
- $\epsilon \in \mathbb{R}^n$ is the random error (continuous random variable) .

We may also refer to the actual observed values (versus the abstract mathematical concept of a random variable) as follows:

- $y = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ is the observed response variable (fixed/observed)
- x_{ij} is the i^{th} observation of the j^{th} explanatory variable (fixed/observed) Data:

Observation	Observed data point
1	$(y_1, x_{11}, x_{12}, \dots x_{1p})$
2	$(y_2, x_{21}, x_{22}, \dots x_{2p})$
$\vdots$	$\vdots$
n	$(y_n, x_{n1}, x_{n2}, \dots x_{np})$

We posit that

$$Y|X = X\beta + \epsilon,$$

where we assume that

- $\forall i \in [n]$, $\mathrm{E}[\epsilon_i] = 0$.
- $\forall i \in [n]$, $\mathrm{Var}[\epsilon_i] = \sigma^2$ (constant variance and is also known as homogeneity.)
- We also would assume that $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.
- $\beta \in \mathbb{R}^{p \times 1}$ is the unknown, population coefficient vector.
- $X \in \mathbb{R}^{n \times p}$ is a covariate matrix.

Let's talk about β. How do we interpret β? Suppose we know β. Then:

Note that

$$\mathrm{E}[Y_i|X_i] = \mathrm{E}\left[\beta^\top X_i + \epsilon\right] = \beta^\top X_i = \beta_1 X_{1,1} + \ldots + \beta_p X_{i,p}$$

What does each β_j mean? Suppose that X_j is a continuous covariate.

We can interpret (β_j) as follows:

Holding $X_{i,1}, \ldots, X_{i,j-1}, X_{i,j+1}, \ldots, X_{i,p}$ constant, a one unit increase in $X_{i,j}$ causes, on average, a β_j unit increase in Y_i.

From another angle, we have that $\partial \mathrm{E}[Y] / \partial X = \beta$, therefore, the rate of change with respect to the j^{th} covariate is β_j.

Caution

The "on average" and "holding other covariates constant" are very important components of the interpretation. First, the on average acknowledges the random error ϵ. In other words, a one unit increase in $X_{i,j}$ will not certainly increase Y_i, but it will on average. Next, the "holding other covariates constant" is used to mention how correlations between covariates are handled by the model. Some of the covariates in the model may be correlated, so increases in a given covariate may often be associated with changes in another covariate. This is not accounted for in the coefficient vectors β. That is why we must specify "holding other covariates constant".
For instance, if a model includes a terms for years of education attained and income, we know that as the number of years of education increase we expect to see a rise in income levels. As a result, to interpret the effect of coefficient on income, we must "hold years of education constant", comparing what is expected with income changes but education does not.

Caution

For now, we can assume that all of the covariates X_j are continuous variables. Later in the course, there may be categorical covariates. In this case, the β_j corresponding to the categorical covariates have a different interpretation. We will return to this later.

Recall Example 3.1. We assume $\forall i \in [20]$, it holds that

$$Y_i|X_i = \beta^\top X_i + \epsilon_i,$$

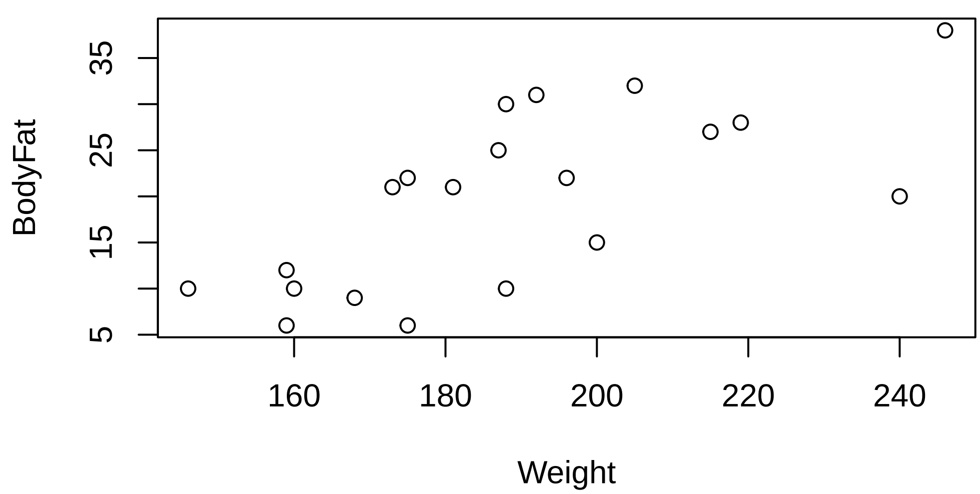
with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [20]$. A one unit increase in weight causes, on average, a β_2 unit increase in body fat percentage. Since β_1 is the intercept, it has a special interpretation. β_1 is the average value of Y_i given $X_i = 0$. It is also helpful to note that $\text{cov}(Y) = \sigma^2 I$.

3.2.2 Least squares estimation

Okay, but we don't know β! Just like we estimate the population mean with the sample mean, we need to estimate β. We would like an estimate $\hat{\beta}$, so that we can predict body fat percentage from weight. What is our best guess at β, given the data? One way to answer this, is through the method of **least squares**.

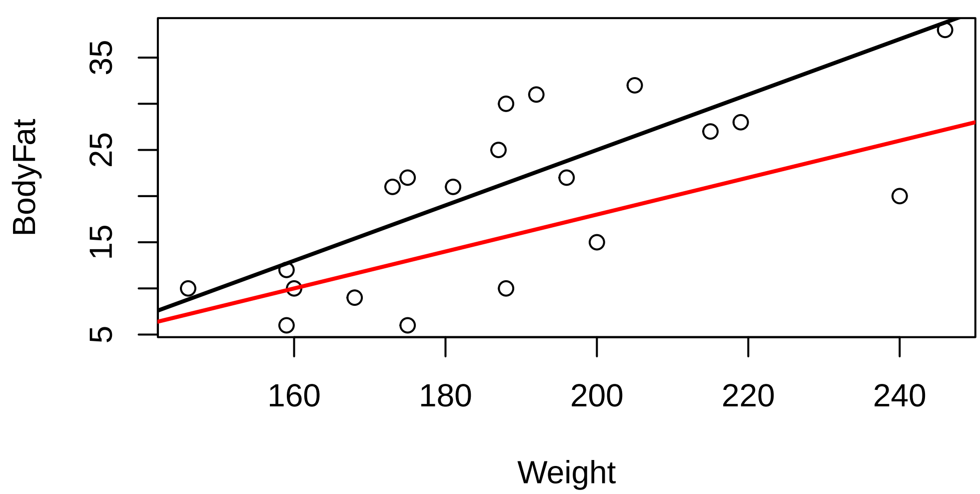
Returning to our example, recall that:

```
plot(df)
```



For example, suppose we want to determine if β is more likely to be $(-35, 0.3)^\top$ or $(-22, 0.2)^\top$. How can we say which line is a better to our data? One way is to graph them on top of the data and determine which one looks better. Let's plot these lines.

```r
plot(df)
# plot Y=-35+0.3X
abline(-35,0.3,lwd=2)
# plot Y=-25+0.2X
abline(-22,0.2,col='red',lwd=2)
```



Its not clear which one fits the data better. Even if it was clear, obviously, we cannot plot all possible lines. So how can we determine which line fits the data the "best"?

To do this, we have to define what "best" means quantitatively. For instance, one might ask which line minimizes the sum of the squared distances of the observed data points to the line? This line is then said to be the "best" line. Mathematically, given a proposed value of β, say $\beta_0 \in \mathbb{R}^p$, the signed distance to the hyperplane $X\beta_0$ is $\epsilon_0 = Y - X\beta_0$. The squared distances to the hyperplane $X\beta_0$ is then $\epsilon_0^\top \epsilon_0 = (Y - X\beta_0)^\top (Y - X\beta_0)$. We can then formulate this as a math problem: Which $\beta_0 \in \mathbb{R}^p$ minimizes $\epsilon_0^\top \epsilon_0$? i.e., $\hat{\beta} = \operatorname{argmin}_{\beta_0 \in \mathbb{R}^p} \epsilon_0^\top \epsilon_0$. It is more convenient to just write

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} (Y - X\beta)^\top (Y - X\beta).$$

In this framework, the "best" estimate is given by

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} (Y - X\beta)^\top (Y - X\beta).$$

Note best is in the sense of minimizing the average squared distance to the hyperplane/line. We could also define best in terms of some other metric, such as average absolute distance to the hyperplane/line. For now, we will stick with this metric.

The next step is to solve:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} (Y - X\beta)^\top (Y - X\beta).$$

How do we minimize a function???

RECALL in calculus, to find the minimum of a function we:

1. Obtain the first two derivatives of the function.
2. Set the first derivative to zero and solve for the critical value.
3. Use the second derivative to verify the critical value minimized the function.

Goal: Compute $\hat{\beta}$ – Minimize $g(\beta) = (Y - X\beta)^\top (Y - X\beta)$. (It may be useful to review taking derivatives with respect to vectors here.

Step 1a:

$$\begin{aligned}
\frac{\partial g}{\partial \beta} &= \frac{\partial g}{\partial \beta}(Y - X\beta)^\top (Y - X\beta) \\
&= \frac{\partial g}{\partial \beta}\left[Y^\top Y - 2(X\beta)^\top Y + (X\beta)^\top X\beta\right] && \text{(Transpose and distribute)} \\
&= -2\frac{\partial g}{\partial \beta}\beta^\top X^\top Y + \frac{\partial g}{\partial \beta}\beta^\top X^\top X\beta && ((AB)^\top = B^\top A^\top) \\
&= -2X^\top Y + 2X^\top X\beta && \left(\tfrac{\partial}{\partial x}x^\top Ax = 2Ax \text{ if } A \text{ symmetric, } \tfrac{\partial}{\partial x}x^\top a = a\right) \\
&= -2X^\top (Y - X\beta).
\end{aligned}$$

Step 1b: (Do this for homework)

$$\frac{\partial^2 g}{\partial \beta \partial \beta^\top} = 2X^\top X.$$

Step 2: We now need $X^\top X$ to be invertible, so we will assume that X is full rank and $n \geq p$.

$$\begin{aligned}
-2X^\top (Y - X\beta) = 0 & \\
&\implies X^\top Y = X^\top X\beta \\
&\implies \beta = (X^\top X)^{-1} X^\top Y.
\end{aligned}$$

Step 3:

Recall that **if the Hessian matrix is positive definite at a critical point, then that critical point is a local minimum.** Since we have assumed X is full rank, this implies that $X^\top X$ is positive definite.

To summarize, the steps have proceeded as follows:

- Step 1a: $\frac{\partial g}{\partial \beta} = -2X^\top(Y - X\beta)$
- Step 1b: $\frac{\partial^2 g}{\partial \beta \partial \beta^\top} = 2X^\top X$ (Do this for homework)
- Step 2: $-2X^\top(Y - X\beta) = 0 \implies X^\top Y = X^\top X\beta \implies \beta = (X^\top X)^{-1}X^\top Y$
- Step 3: $2X^\top X$ is positive definite, and so

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y.$$

The estimate $\hat{\beta}$ is known as the **least squares estimate** of the regression coefficients.

Definition 3.1. The **least squares estimate** of the regression coefficients is

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y.$$

3.2.3 Example

Example 3.2. In the body weight example Example 3.1, write down X, y and compute $\hat{\beta}$. Interpret $\hat{\beta}$.

First, we have that

$$y = (6, 21, 15, 6, 22, 31, 32, 21, 25, 30, 10, 20, 22, 9, 38, 10, 27, 12, 10, 28)^\top$$

$$X = \left[1_{20} \mid (175, 181, 200, 159, 196, 192, 205, 173, 187, 188, 188, 240, 175, 168, 246, 160, 215, 159, 146, 219)^\top\right]$$

> **Note**
>
> For matrices A, B which have the same number of rows, $C = [A|B]$ is horizontal concatenation of A and B. This notation indicates that the matrix C is formed by placing A an B side by side, joining them horizontally. Therefore, X is the matrix whose first column is made up of ones, and second column is made up of the body weights.

Let's use R to compute $\hat{\beta}$.

```
#Define X and Y
X=cbind(rep(1,nrow(df)), df$Weight)
Y=df$BodyFat

# cast to column vec
Y=matrix(Y,ncol=1)

#X'X
X_p_X=t(X)%*%X
```

```r
#X'X inverse
X_p_X_inverse=solve(X_p_X)


#LS
beta_hat= X_p_X_inverse%*%t(X)%*%Y
beta_hat
```

```
            [,1]
[1,] -27.3762623
[2,]   0.2498741
```

```r
# We can also use Rs lm() function to do this:
# This code is essential for the course.
# The first argument is the formula
model=lm(BodyFat ~ Weight, data=df)

#The summary function prints the model output.

summary(model)
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```
# The least squares estimates are given in the Estimate column of the summary.
```

The `lm()` function is used to fit multiple linear regression models in R. The basic usage involves specifying a formula and a data frame. The syntax is given by `lm(formula, data, ...)`.

The data argument should be the dataframe which contains your data. The formula argument is used to specify the model to be fitted. It provides a symbolic description of the model, indicating the response variable and the predictors/covariates, as well as the relationships between them. The left-hand side should be the name of your response variable, as it is named in your dataframe. To see the names of your variables use the `names()` function, e.g., `names(df)`. The right-hand side contains the covariates you want to include in your model. For instance, above, the formula is given by `BodyFat ~ Weight` . Note that `BodyFat` is the response and `Weight` is the covariate.

We now list some important properties of the least squares estimator.

Exercise 3.5. Compute $\mathrm{E}\left[\hat{\beta}\right]$ and $\mathrm{cov}(\hat{\beta})$.

It holds that $\mathrm{E}\left[\hat{\beta}\right] = \beta$ and $\mathrm{cov}(\hat{\beta}) = \sigma^2(X^\top X)^{-1}$.

Recall that an estimator is **unbiased** if its expectation equals the population parameter it is trying to estimate. After completing Exercise 3.5 you will see that $\hat{\beta}$ is unbiased for the parameter β.

The least squares estimator is also the "best linear unbiased estimator", or the BLUE. This is known as the Gauss–Markov theorem. This means that under the assumptions of the linear regression model, over any unbiased estimator of β we can construct, which is a linear combination of $Y_1, \dots, Y_n$, the estimator $\hat{\beta}$ has the smallest variance (and therefore, the smallest mean squared error. Recall that for an estimator $\hat{\alpha}$, the mean squared error is given by $\mathrm{E}\left[||\beta - \hat{\alpha}||^2\right]$.)

The Gauss–Markov theorem does not require the random error to be normally distributed. If we are willing to assume that $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, then $\hat{\beta}$ is also the maximum likelihood estimator and the "uniformly minimum-variance unbiased estimator", or UMVUE. This means that $\hat{\beta}$ has lower variance than any other unbiased estimator, no matter what the true value of β is.

One might ask, how can we use $\hat{\beta}$ to predict body fat percentage given weight? The estimate $\hat{\beta}$ gives us a best guess at the coefficients. Therefore, our best guess at someones body fat is given by

$$Best\ Guess = -27.3762623 + 0.2498741 \times Weight.$$

For instance, for someone who is 170 pounds, we would guess that their body fat percentage is $-27.3762623 + 0.2498741 \times 170 = 15.1023347$.

3.2.4 Homework stop 2

Exercise 3.6. Why do we need $\hat{\beta}$, why not use β?.

Exercise 3.7. Is $\hat{\beta}$ an estimate or a population parameter? What about β?

Exercise 3.8. Compute, X, Y and $\hat{\beta}$ in the following real data example:

It is challenging to assess a student's understanding of a subject without administering an exam. However, we can easily record the number of hours a student studies. A researcher would like to know if the number of hours studied and exam scores are related. This researcher collected the following data:

Student	Hours Studied	Exam Score (%)
1	5	55
2	8	65
3	12	78
4	6	58
5	10	72
6	9	68
7	15	85
8	7	60
9	11	74
10	13	80
11	14	82
12	20	90
13	5	55
14	6	59
15	18	88
16	7	62
17	16	86
18	4	50
19	3	45
20	19	89

To help you, here is some R code the dataset:

```
# Data
study_data <- data.frame(
  Student = 1:20,
  Hours_Studied = c(5, 8, 12, 6, 10, 9, 15, 7, 11, 13, 14, 20, 5, 6, 18, 7, 16, 4, 3, 19),
  Exam_Score = c(55, 65, 78, 58, 72, 68, 85, 60, 74, 80, 82, 90, 55, 59, 88, 62, 86, 50, 4
```

```
)
```

3.3 Least squares inference

Recall we **estimate** the parameter β using least squares:

Recall that $\hat{\beta} = (X^\top X)^{-1} X^\top Y$. We can predict a new weight $Y_{new}|X = x$ with $\hat{y}_{new} = x^\top \hat{\beta}$. We may be interested in the following questions: How good is $\hat{y}_{new}$ as a prediction, on average? How will new observations vary about the line? For example, given a specific weight, how will does body fat percentage vary around the regression line? How does $\hat{\beta}$ vary around β? Is there strong evidence that Y has a relationship with X? Is X adding information about Y at all?

To answer these questions, we need to look at the variation of our estimates and our data.

3.3.1 Important quantities: Residuals and fitted values

We now introduce some very important quantities: We call the estimated values given our observed X the fitted values: $\hat{Y} = X\hat{\beta}$. The fitted values are what our model would estimate the vector Y to be. We call $\hat{\epsilon} = Y - \hat{Y}$ is the **residual vector**. The ith entry of $\hat{\epsilon}$, say $\hat{\epsilon}_i$, is the ith **residual**. The residuals are the signed distances from the response variable to the estimated regression hyperplane. The **sum of squared error** or **sum of squared residuals** (SSE) is given by $\hat{\epsilon}^\top \hat{\epsilon} = \sum_{i=1}^n \hat{\epsilon}_i^2$. Note that since we estimated β using the least squares method, $\hat{\epsilon}^\top \hat{\epsilon}$ is minimized (with respect to varying β).

Example 3.3. Recall Example 3.1. What is the residual of individual 3? How can we interpret this value?

```
residuals=Y-X%*%beta_hat; residuals
```

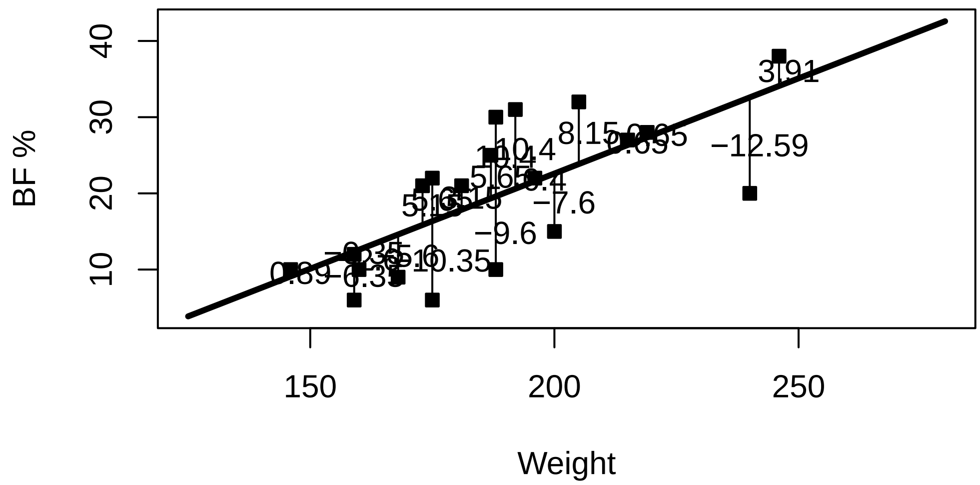
```
             [,1]
 [1,] -10.3517117
 [2,]   3.1490434
 [3,]  -7.5985652
 [4,]  -6.3537255
 [5,]   0.4009314
 [6,]  10.4004279
 [7,]   8.1520641
 [8,]   5.1480365
 [9,]   5.6497986
[10,]  10.3999245
```

```
[11,]  -9.6000755
[12,] -12.5935307
[13,]   5.6482883
[14,]  -5.6025928
[15,]   3.9072245
[16,]  -2.6035997
[17,]   0.6533228
[18,]  -0.3537255
[19,]   0.8946383
[20,]   0.6538262
```

```
# This means that individual 3's body fat is 7.5 percentage points lower than the fitted l

residuals[3]
```

```
[1] -7.598565
```

```
# We can go further and and plot all of the residuals
curve(beta_hat[1]+beta_hat[2]*x,125,280,lwd=3,xlab="Weight",ylab="BF %")
points(Weight,BodyFat,pch=22,bg=1)
Yvals=cbind(BodyFat,model$fitted.values)
Xvals=cbind(Weight,Weight)
for(i in 1:nrow(Yvals)){
  lines(Xvals[i,],Yvals[i,])
  text(Xvals[i,1]+2,mean(Yvals[i,]),round(residuals[i],2))
}
```




```
# Then, the population body fat percentages, given weights will look like this:
#
# Bfs=-15+.2*Weight+rnorm(20,0,sd=5)
```

3.3.2 Variation decomposition

Variance decomposition is a fundamental concept that explains how the total variation in the response variable can be partitioned into different sources. This decomposition is crucial for evaluating the performance of the regression model and understanding the contributions of various factors.

The residuals describe one type of variation of the response values. We can also consider the total variation of the response. The total variation of the response, or the **sum of squares total/total sum of squares** (SST) is given by $SST = (n-1)\hat{\sigma}_y = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 = (Y - \bar{Y}1)^\top(Y - \bar{Y}1)$. It can be shown that the SST can be decomposed as follows:

$$SST = (Y-\bar{Y}1)^\top(Y-\bar{Y}1) = (Y-\hat{Y})^\top(Y-\hat{Y}) + (\hat{Y}-\bar{Y})^\top(\hat{Y}-\bar{Y}) = \hat{\epsilon}^\top\hat{\epsilon} + (\hat{Y}-\bar{Y})^\top(\hat{Y}-\bar{Y}).$$

That is, $SST = SSE + SSModel$ where

- $SSModel$, OR SSM measures the total variations of the response explained by the covariates X via the model based on $\hat{\beta}$.

- SSE measures the total variations of the response unexplained by the covariates X via the model based on $\widehat{\beta}$.
- Note there are sometimes other names for SSE and $SSModel$, such as $SSRegression$, $SSwithin$ and $SSbetween$, etc.

So, we have that the total variation in the response can be broken down into that which is explained by the X values, and that which is unexplained.

An interesting observation is given as follows: The first column of the X matrix is given by 1_n, which implies that

$$\bar{Y}1 = X \begin{bmatrix} \bar{Y} \\ 0 \end{bmatrix}.$$

This means that if we let $\widehat{\beta}_* = (\bar{Y}, 0, \dots, 0)^\top$, then $(Y - \bar{Y}1)$ would be the signed distances to (or the residuals of) the regression hyperplane corresponding to $\widehat{\beta}_*$. Since $\widehat{\beta}$ minimizes the sum of squared residuals, we must have that the hyperplane corresponding to $\widehat{\beta}$ has a smaller sum of squared residuals than the regression hyperplane corresponding to $\widehat{\beta}_*$. Therefore, we must have that $\hat{\epsilon}^\top \hat{\epsilon} \leq (Y - \bar{Y}1)^\top (Y - \bar{Y}1)$.

Each of these terms in the decomposition is associated with a certain number of **degrees of freedom**.

- Total: $dfT = n - 1$.
- Model: $dfM = \#$ **non-zero** $\beta - 1$.
- Error: $dfE = n - \#$ **non-zero** β.

Intuitively, since the SSE is the variance unexplained by the model/covariates, the SSE is related to the error variance σ^2. In fact, to estimate σ^2, we use

$$\hat{\sigma}^2 = MSE = \frac{SSE}{dfE}.$$

The null model is defined as $Y|X = \beta_0 + \epsilon$. This is the model where the last $p - 1$ terms in the true vector β are 0. This model says that Y does not depend on X. In the null model, we only need to estimate the mean, so $df = n - 1$. Therefore, under the null model,

$$\begin{aligned} \hat{\sigma}^2 = (n-1)^{-1} SST = \hat{\sigma}_Y^2 & \\ &= (n-1)^{-1} \sum_{i=1}^{n} (Y_i - \bar{Y})^2 = (n-1)^{-1} (Y - \bar{Y}1)^\top (Y - \bar{Y}1). \end{aligned}$$

Therefore, in the null model, the estimate of σ^2 via the MSE is just the usual estimate of the variance of the response. This is intuitive!

The following table can be used to summarize the variation in the response:

Source	SS	df	MS
Model	SSM	dfM	$MSModel = SSM/dfM$
Residual	SSE	dfE	$MSE = SSE/dfE$
Total	SST	dfT	

Note

It is very important to be able to interpret these terms! The derivation is also important. However, we can use a machine to compute anything for us, so memorizing the formula is not helpful.

3.3.3 Coefficients of determination

A model is a good model if it can explain a fair amount of the variation in the response. (You can think that the model explains "changes" in the response.) In other words, $SSModel$ should be as close to $SSTotal$ as possible; or equivalently, $SSError$ should be as close to 0 as possible. Now, "close" is a relative term, and so we need another value to reference to. This is where the R^2 comes in:

$$R^2 = \frac{SSModel}{SST},$$

and is the proportion of variation explained by the model. It is clear that $0 \leq R^2 \leq 1$, and so rescaling the data will not affect R^2 (like it would affect the sum of squares terms SST, SSE, SSM). If R^2 is close to 1, it is large – "close to 1" is a subjective/area dependent. Generally, the larger the R^2, the better the model!

To compare different models, we could potentially add different covariates and see if R^2 improves. However, every time you add any variable, R^2 will always increase. Therefore, it is common to use the adjusted coefficient of determination:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p}.$$

Thus, the (adjusted) coefficient of determination can be used as a measure of how well the regression model fits the data (how much variance is explained). It could also be used to compare models.

3.3.4 The F test

The coefficients of determination are summary statistics which give an idea of the fit of the model. We would also like a significance test that tells us whether the covariates explain Y, or what we observed was simply due to sampling variation.

If $\beta = (\beta_1, \dots, \beta_p)^\top$ then let $\tilde{\beta} = (\beta_2, \dots, \beta_p)^\top$. That is $\tilde{\beta}$ is the regression coefficients without the intercept term. Similarly, let $\tilde{\hat{\beta}} = (\hat{\beta}_2, \dots, \hat{\beta}_p)^\top$. Now, we want to avoid the situation where $\tilde{\beta} = 0$ but $\tilde{\hat{\beta}} \neq 0$ due to sampling variation.

To do this, we perform a significance test:

$$H_0 : \tilde{\beta} = 0 \qquad vs \qquad H_1 : \tilde{\beta} \neq 0.$$

First, we need the normality assumption to perform significance test: Assume $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. With this assumption, the model is then known as the **Normal Multiple Linear Regression Model**. It is important to note that the least squares method does not require this assumption, and this assumption is required only for the significance test to be valid. To test the hypothesis stated above, we use the overall F test and the observed test statistic is $F_{obs} = MSModel/MSE$. Why?

With the extra normality assumption, we have the following holds:

- $Y|X$ is normally distributed.
- We have that $SSM/\sigma^2 \sim \chi^2_{dfM}$ and $SSE/\sigma^2 \sim \chi^2_{dfE}$.
- Furthermore, $SSM \perp SSE$.

Recall that the ratio of two independent χ^2 distributions divided by their respective degrees of freedom follows an F distribution. Therefore, we have that $F_{obs} \sim F_{dfM,dfE}$. The corresponding p-value is $\Pr(W > F_{obs})$ where $W \sim F_{dfM;dfE}$. We can alternatively reject the null hypothesis if $F_{obs} > F_{dfM;dfE,1-\alpha}$, where $F_{obs} > F_{dfM;dfE,1-\alpha}$ is the $1-\alpha$ quantile of the $F_{dfM,dfE}$ distribution.

We can now present the complete ANOVA table

Source	SS	df	MS	F	p-value
Model	SSM	dfM	$MSModel = \frac{SSR}{dfM}$	$F = \frac{MSModel}{MSE}$	$\Pr(W > F_{obs})$
Residual	SSE	dfE	$MSE = \frac{SSE}{dfE}$		
Total	SST	dfT			

Example 3.4. In Example 3.1, compute and interpret the coefficients of determination. Compute and interpret the ANOVA table. Test whether the regression model is significant. (This means perform the F test.)

```
# recall
head(df)
```

```
  Weight BodyFat
1    175       6
2    181      21
3    200      15
4    159       6
5    196      22
6    192      31
```

```r
# The F test results are given in the summary
summary(model)
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```r
# The ANOVA table is given below

# First define the null model object using lm()
# This line fits a model with only the intercept term
null_model=lm(BodyFat~1,data=df)

# This line gets the ANOVA table
anova(null_model,model)
```

```
Analysis of Variance Table
```

```
Model 1: BodyFat ~ 1
Model 2: BodyFat ~ Weight
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     19 1737.75
2     18  894.42  1    843.33 16.972 0.0006434 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# We can also do this by hand:


# Store sample size
n=nrow(df)
p=2

# Compute sum of squares
SST=t(Y-mean(Y)*rep(1,n))%*%(Y-mean(Y)*rep(1,n))
Yhat=X%*%beta_hat
res=Y-Yhat
SSE=t(res)%*%res
SSM=SST-SSE

dfe=n-p
dfm=p-1
MSM=SSM/dfm


MSE=SSE/dfe

Fv=MSM/MSE

p.val=1-pf(Fv,dfm,dfe)

# ANOVA Table:
ANOVA_Table=rbind(c(SSM,dfm,MSM,Fv,p.val),c(SSE,dfe,MSE,NA,NA),c(SST,n-1,NA,NA,NA))
rownames(ANOVA_Table)=c("Model","Error","Total")
colnames(ANOVA_Table)=c("SS","df","MS","F","p-value")
ANOVA_Table
```

```
             SS df         MS         F      p-value
```

```
Model  843.3252  1 843.32521 16.97164 0.0006434484
Error  894.4248 18  49.69027       NA           NA
Total 1737.7500 19        NA       NA           NA
```

3.3.5 Homework stop 3

Exercise 3.9. In the following real data example: **Compute and interpret** the coefficient of determination, the adjusted coefficient of determination and perform the F test for model significance. Including printing the ANOVA table, the null and alternative hypothesis, an interpretation of the p-value and the conclusion of the test.

It is challenging to assess a student's understanding of a subject without administering an exam. However, we can easily record the number of hours a student studies. A researcher would like to know if the number of hours studied and exam scores are related. This researcher collected the following data:

Student	Hours Studied	Exam Score (%)
1	5	55
2	8	65
3	12	78
4	6	58
5	10	72
6	9	68
7	15	85
8	7	60
9	11	74
10	13	80
11	14	82
12	20	90
13	5	55
14	6	59
15	18	88
16	7	62
17	16	86
18	4	50
19	3	45
20	19	89

To help you, here is some R code the dataset:

```
# Data
study_data <- data.frame(
  Student = 1:20,
  Hours_Studied = c(5, 8, 12, 6, 10, 9, 15, 7, 11, 13, 14, 20, 5, 6, 18, 7, 16, 4, 3, 19),
  Exam_Score = c(55, 65, 78, 58, 72, 68, 85, 60, 74, 80, 82, 90, 55, 59, 88, 62, 86, 50, 4
)
```

Exercise 3.10. Write down the interpretations of: SSE, MSE, R^2, $\bar{R}^2$, SSM.

Exercise 3.11. What is the interpretation of the p-value in the ANOVA table?

Exercise 3.12. What extra assumption is needed to perform the F-test?

3.3.6 Significance of one variable

So far, we have learned that the least squares method yields the following estimate of $\hat{\beta} = (X^\top X)^{-1} X^\top Y$ with $\mathrm{E}\left[\hat{\beta}\right] = \beta$ and $\mathrm{cov}(\hat{\beta}) = (X^\top X)^{-1}\sigma^2$. Moreover, we use MSE to estimate σ^2. Next, we learned that we can summarize the SS, df, and MS in an ANOVA table. We used the F test and the coefficient of determination to evaluate the quality of the model, i.e., to see the amount of information X provides about Y.

When the model is a significant model, then, at least one of the individual explanatory variables is useful in explaining the response. We may be interested in whether a specific covariate, or set of covariates is useful in explaining the response variable. We now learn how we can test for the significance of each individual explanatory variable separately and how we can test for the significance of a subset of explanatory variables. Note that these tests also require that the random error is normally distributed.

To test for significance and compute confidence intervals of a single variate, we have to compute the distribution of $\hat{\beta}_j$. We first compute the mean and variance of $\hat{\beta}_j$. First, given that $\mathrm{E}(\hat{\beta}) = \beta$, we have $\mathrm{E}(\hat{\beta}_j) = \beta_j$. Next, $\mathrm{Var}\left[\hat{\beta}_j\right]$ is the $(j,j)^{th}$ entry of $\mathrm{cov}(\hat{\beta})$. In addition, we have derived that $\mathrm{cov}(\hat{\beta}) = (X^\top X)^{-1}\sigma^2$.

Now, recall that if Z is multivariate normal, i.e., $Z \sim \mathcal{N}(\mu, \Sigma)$, then $b + AZ \sim \mathcal{N}(b + A\mu, A\Sigma A^\top)$, i.e., $b + AZ$ is also multivariate normal. Therefore, since we have assumed that $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I)$ and that $Y|X = X\beta + \epsilon$, it follows that $Y|X \sim \mathcal{N}_n(X\beta, \sigma^2 I)$. Next, we may recall that $\hat{\beta} = (X^\top X)^{-1} X^\top Y$. Let $A = (X^\top X)^{-1} X^\top$. Then $\hat{\beta} = AY$. It follows that $\hat{\beta}$ is also multivariate normal! Putting everything together, we have that $\hat{\beta} \sim \mathcal{N}_p(\beta, (X^\top X)^{-1}\sigma^2)$.

Theorem 3.1. *Under the assumptions of the* ***normal linear regression model*** *it holds that* $\hat{\beta} \sim \mathcal{N}_p(\beta, (X^\top X)^{-1}\sigma^2)$.

Now that we have the distribution of $\hat{\beta}$, we can use it to compute the confidence intervals for β_js.

Recall from introductory statistics (MATH 1131) that you learned that if we want to compute a confidence interval for the sample mean and the sample variance was unknown, we had to estimate the variance. Similarly, here, the variance of $\hat{\beta}_j$ contains σ, an unknown parameter. Recall that, we estimate σ^2 by MSE, and so we can estimate the variance of $\hat{\beta}_j$ by $\hat{\text{Var}}\left[\hat{\beta}_j\right] = (X^\top X)^{-1}_{j,j} MSE$.

It can be shown that $\hat{\beta} \perp MSE$. Therefore, we have that

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{\widehat{var}(\hat{\beta}_j)}} \sim t_{dfE}.$$

Now that we know the distribution of $\hat{\beta}_j$, we can perform significance testing and compute confidence intervals.

If we want to test

$$H_0 \colon \beta_j = \beta_j^0 \qquad vs \qquad \beta_j \neq \beta_j^0$$

we can do the following.

The observed test statistic is $TS = \frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}}$. Note that, under the null hypothesis, we have that $\frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}} \sim t_{dfE}$. Thus, the corresponding p-value is obtained based on the t_{dfE} distribution. Specifically, we can compute the p-value $\Pr\left(-|TS| < Z\right) + \Pr\left(|TS| > Z\right) = 2 * \Pr\left(|TS| > Z\right)$, where $Z \sim t_{dfE}$.

The test proceeds as follows:

1. State the hypotheses

$$H_0 \colon \beta_j = \beta_j^0 \qquad vs \qquad H_1 : \ \beta_j \neq \beta_j^0.$$

2. Compute the test statistic $\frac{\hat{\beta}_j - \beta_j^0}{\sqrt{\widehat{var}(\hat{\beta}_j)}}$ and the p-value.
3. Interpret the p-value, and use it to decide whether you reject the null hypothesis.

Often, one may choose a threshold α, and reject the null hypothesis if the p-value falls below that threshold. Other times, we use the p-value as a description of evidence against the null. If it is larger than 0.05, but still small, then that still constitutes some evidence against the null hypothesis.

Let's now discuss one-sided hypotheses. First, consider:

$$H_0 \colon \beta_j \leq \beta_j^0 \qquad vs \qquad H_1 : \ \beta_j > \beta_j^0$$

Then, if the alternative hypothesis is true, we expect TS to be positive. The p-value is given by $\Pr(TS > Z)$, where $Z \sim t_{dfE}$. Notice that the p-value is measuring how extremely positive TS is. Using the threshold method, we can also check if $TS > t_{dfE,1-\alpha}$. Next, if we want to test

$$H_0: \beta_j \geq \beta_j^0 \qquad vs \qquad H_1: \ \beta_j < \beta_j^0,$$

then if the alternative hypothesis is true, we expect TS to be negative. The p-value is given by $\Pr(TS < Z)$, where $Z \sim t_{dfE}$. Notice that the p-value is measuring how extremely negative TS is. Using the threshold method, we can also check if $TS < t_{dfE,\alpha}$.

Note

We use $t_{k,p}$ to denote the pth quantile of the t distribution with k degrees of freedom. For $p = 0.025$ and large k, this is approximately equal to 2.

In Example 3.1, test if the coefficient for weight is equal to 1/3 vs. not equal to 1/3. Next, test if the coefficient for weight is less than or equal to 1/3 vs. greater than 1/3. Lastly, test if the coefficient for weight is non-zero.

First, we have that

$$\begin{aligned}
H_0: \beta_1 = 1/3 \qquad &vs \qquad H_1: \ \beta_1 \neq 1/3.\\
H_0: \beta_1 \leq 1/3 \qquad &vs \qquad H_1: \ \beta_1 > 1/3.\\
H_0: \beta_1 = 0 \qquad &vs \qquad H_1: \ \beta_1 \neq 0.
\end{aligned}$$

Now, let's execute the tests:

```
#changing matrix to scalar
MSE=c(MSE)
hvar_beta=solve(t(X)%*%X)*MSE

print(beta_hat)
```

```
            [,1]
[1,] -27.3762623
[2,]   0.2498741
```

```
TS=(beta_hat[2]-1/3)/sqrt(hvar_beta[2,2])

# not equal
# pt(x,df) is the CDF of a t distributed RV with df degrees of freedom at x.
p_val=2*(1-pt(abs(TS),dfe))
p_val
```

```
[1] 0.1857053
```

```r
# greater than
# pt(x,df) is the CDF of a t distributed RV with df degrees of freedom at x.
p_val=1-pt(TS,dfe)
p_val
```

```
[1] 0.9071474
```

```r
## testing equal to 0
TS=beta_hat[2]/sqrt(hvar_beta[2,2])

# not equal
# pt(x,df) is the CDF of a t distributed RV with df degrees of freedom at x.
p_val=2*(1-pt(abs(TS),dfe))
p_val
```

```
[1] 0.0006434484
```

```r
# We can also use the model object to test if it is not equal to 0:
# The test statistic and the pvalue are given in the t value and Pr(>|t|)  columns, respec
summary(model)
```

```
Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.5935  -5.7904   0.6536   5.2731  10.4004 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -27.37626   11.54743  -2.371 0.029119 *  
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
```

```
Multiple R-squared:  0.4853,	Adjusted R-squared:  0.4567 
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

Based on the concepts that you have learned in 1131, and what we have reviewed in previous lectures, it also follows from the above analysis that a $(1-\alpha)100$ confidence interval for β_j is

$$\hat{\beta}_j \pm t_{dfE,\alpha/2}\sqrt{\widehat{var}(\hat{\beta}_j)}.$$

Example 3.5. In Example 3.1, compute a 99% and a 95% confidence interval for the coefficient for weight. Which one is longer? Why? Interpret these intervals.

```r
# By hand
beta_hat[2]+c(-1,1)*qt(0.975,dfe)*sqrt(hvar_beta[2,2])
```

```
[1] 0.1224448 0.3773035
```

```r
beta_hat[2]+c(-1,1)*qt(0.995,dfe)*sqrt(hvar_beta[2,2])
```

```
[1] 0.07528522 0.42446306
```

```r
# Auto software/using lm:
confint(model,level=0.95)
```

```
                  2.5 %     97.5 %
(Intercept) -51.6365090 -3.1160157
Weight        0.1224448  0.3773035
```

```r
confint(model,level=0.99)
```

```
                   0.5 %    99.5 %
(Intercept) -60.61484736 5.8623227
Weight        0.07528522 0.4244631
```

If we took many samples of size 20 and computed a 95% (99%) confidence interval for each sample, then 95% (99%) of then would contain the true coefficient for the weight variable. We can conclude that with 95% (99%) confidence, the true coefficient for weight likely falls within (0.12, 0.38) ((0.08,0.42)).

Caution

The key to understanding a confidence interval is to realize that the end points of the interval depend on the sample, and are therefore, random. On the other hand, the population parameter is not random, it is fixed. Therefore, if we drew a different sample, the interval would move, and there is a $(1-\alpha)100\%$ chance that that interval catches the population parameter. Most of the time it will contain the parameter, but not always.

Recall that the point of computing a confidence interval is to report the uncertainty in our estimate that resulted from drawing a sample. We expect the true parameter to be somewhere in that range, and our best guess at the parameter is given by the center of the interval.

3.3.7 Inference for the mean response and prediction intervals

We may wish to estimate the average response at a specific set of the covariates x. Given x, the theoretical mean response is $x^\top\beta$. Given x, we can estimate the mean response as $x^\top\hat{\beta}$. For instance, what is the average body fat percentage at 160 pounds? How accurate is our estimate? We can use a confidence interval to answer this question.

Note that the expectation and variance of the estimate of the mean response are given by $\mathrm{E}\left[x^\top\hat{\beta}\right] = x^\top\beta$ and $\mathrm{Var}\left[x^\top\hat{\beta}\right] = x^\top(X^\top X)^{-1}x\sigma^2$. Again, we must estimate σ and we can write $\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right] = x^\top(X^\top X)^{-1}xMSE$.

Exercise 3.13. Under the assumptions of the normal linear regression model, show that for a fixed covariate vector $x \in \mathbb{R}^p$, $x^\top\hat{\beta}$ has a multivariate normal distribution and find it's mean and variance. Argue that $\frac{x^\top\hat{\beta}-x^\top\beta}{\sqrt{\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right]}} \sim t_{dfE}$.

It can be shown that a $(1-\alpha)100\%$ confidence interval for the mean response $\mathrm{E}\left[Y|X=x\right]$ is

$$x^\top\hat{\beta} \pm t_{dfE,\alpha/2}\sqrt{\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right]}.$$

Similarly, if we want to test

$$H_0\colon \mathrm{E}\left[Y|X=x\right] = \mu_0 \qquad vs \qquad \mathrm{E}\left[Y|X=x\right] \neq \mu_0$$

we can do the following:

The observed test statistic is $TS(x,\mu_0) = \frac{x^\top\hat{\beta}-\mu_0}{\sqrt{\hat{\mathrm{Var}}\left[x^\top\hat{\beta}\right]}}$. Observe that under the null hypothesis, we have that $TS(x,\mu_0) \sim t_{dfE}$. Therefore, the p-value is given by $2 * \Pr\left(|TS(x,\mu_0)| > Z\right)$.

Similar to the previous section, we can also perform one-sided tests:

- Right-sided test ($H_1 : x^\top\beta > \mu_0$): p-value $\Pr(TS(x, \mu_0) > Z)$.
- Left-sided test ($H_1 : x^\top\beta < \mu_0$): p-value $\Pr(TS(x, \mu_0) < Z)$.

We may also wish to predict what the response will be, given a new set of covariates. On top of that, we may again wish to quantify how much error there is in our prediction. For instance, what is the predicted body fat percentage of someone who is 160 pounds? Note that this differs from the previous section. In the previous section, we were interested in the average body fat percentage of someone who is 160 pounds. Here, we are interested in predicting the body fat percentage of a single, specific person, and not the average of the whole population.

Specifically, suppose that we have a subject whose covariates are given by z, but we do not know the value of the subjects response, which we can denote by Y_{new}. Then the true response is $(Y_{new}|Z = z) \; = z^\top\beta + \epsilon_{new}$.

Suppose we want to predict Y_{new} and give an idea of how much error is in our prediction. The predicted response is known, and is given by $\mathrm{E}[Y_{new}|Z = z] = z^\top\hat{\beta}$. We have $\mathrm{Var}[Y_{new}|Z = z] = \mathrm{Var}\left[z\hat{\beta}\right] + \mathrm{Var}[\epsilon_{new}] = z^\top(X^\top X)^{-1}z\sigma^2 + \sigma^2$. Therefore, the variation in a new response is the variation in our estimate of β plus the inherent population variation, σ^2. We have that this can be estimated with: $\widehat{\mathrm{Var}}[Y_{new}|Z = z] = z^\top(X^\top X)^{-1}zMSE + MSE$.

Exercise 3.14. Under the assumptions of the normal linear regression model, show that for a fixed covariate vector $z \in \mathbb{R}^p$, $Y_{new}|Z = z$ has a multivariate normal distribution and find it's mean and variance. Argue that given $Z = z$,

$$\frac{Y_{new} - z^\top\beta}{\sqrt{\widehat{\mathrm{Var}}[Y_{new}]}} \sim t_{dfE}.$$

Therefore, the $(1 - \alpha)100\%$ prediction interval for Y_{new} is given by:

$$z\hat{\beta} \pm t_{dfE,\alpha/2}\sqrt{z^\top(X^\top X)^{-1}zMSE + MSE}.$$

Note that the prediction interval is wider than that of the mean response interval for the same covariate vector z. That is because it is more difficult to predict the response for a specific person than it is to estimate a mean of a population. Furthermore, the interpretation of a prediction interval is different. A $(1 - \alpha)100\%$ prediction interval can be interpreted it as follows. Given a $(1 - \alpha)100\%$ prediction interval for $Y_{new}|Z = z$, say (a, b), we say that the probability Y_{new} is in (a, b) is $(1 - \alpha)100\%$. Note that this differs substantially from a confidence interval!

Example 3.6. In Example 3.1, execute the following: What is a 95% confidence interval for the mean of someone who weighs 165 pounds? What is a 95% prediction interval for the BF% of someone who weighs 165 pounds? Interpret these intervals.

```r
# Intervals are given as follows:

z <- data.frame(Weight=165)
predict(model, newdata = z, interval = 'confidence')
```

```
       fit      lwr      upr
1 13.85297 9.379675 18.32627
```

```r
predict(model, newdata = z, interval = 'prediction')
```

```
       fit       lwr      upr
1 13.85297 -1.617547 29.32349
```

We are 95% confident the mean body fat of a person who weighs 165 pounds is in 13.8529704, 9.3796749, 18.3262658. There is a 95% probability that the body fat of a person who weights 165 pounds is in 13.8529704, -1.6175473, 29.323488 . Note that the prediction interval is wider!

3.3.8 Homework stop 4

Exercise 3.15. What is the difference between a prediction interval and an interval for the mean response ?

Exercise 3.16. Code the confidence intervals for the mean response and prediction interval without using the predict function.

Exercise 3.17. Complete the chapter 3 practice problems from the problem list.

3.3.9 Partial testing

We may be interested in executing the following hypothesis test:

$$H_0 \colon (\beta_1, \dots, \beta_k) = 0 \qquad vs \qquad (\beta_1, \dots, \beta_k) \neq 0.$$

This amounts to testing whether the subset of variables $(\beta_1, \dots, \beta_k)$ adds anything to the model beyond $(\beta_{k+1}, \dots, \beta_p)$. For example, you may be interested in whether location related covariates affect the price of Airbnb. The overall idea is to compare the reduced (null) model with $p - k$ covariates to the complete (saturated, full) model (which contains all covariates).

Let's first review the F-test. We learned about the F test, which compares the following models:

$$Y|X = \beta^\top X + \epsilon \qquad vs \qquad Y|X = \beta_1 + \epsilon.$$

Here, the complete model is given by $Y|X = \beta^\top X + \epsilon$ and the reduced model is given by $Y|X = \beta_1 + \epsilon$. Recall that the test statistic is given by

$$\frac{SSM/dfM}{SSE/dfE} = \frac{(SST - SSE)/(dfT - dfE)}{SSE/dfE},$$

where the degrees of freedom are in terms of the full model (not the null model). We could then rewrite this test statistic as

$$\frac{SSM_C/dfM_C}{SSE_C/dfE_C} = \frac{(SST_C - SSE_C)/(dfT_C - dfE_C)}{SSE_C/dfE_C},$$

where C stands for the complete model. (All that has changed is the notation, we added a C subscript.)

Now, note that $SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$ has nothing to do with what covariates are in the model. In other words, SST is always the same, not matter what covariates are in the model. Therefore, $SST_C = SST_R = SST$, where SST_R stands for the "sum of squares total" in the reduced model. In our example of the F test, the least squares estimate of β_1 in the reduced model is $\hat{\beta}_1 = \bar{Y}$ and the associated residual vector is given by $\hat{\epsilon} = Y - \bar{Y}1_n$. But wait, observe that in this case, we have that $\hat{\epsilon}^\top \hat{\epsilon} = SST$! Therefore, putting everything together, in this example, we have that $SSM_C = SST_C - SSE_C = SSE_R - SSE_C$. That is, the model sum of squares for the complete model is the difference between the sum-squared error in the reduced model and the sum-squared error in the complete model. We can then rewrite the test statistic as

$$\frac{(SSE_R - SSE_C)/(dfT_C - dfE_C)}{SSE_C/dfE_C}.$$

The difference $SSE_R - SSE_C$ can be interpreted as the extra information gained from adding the covariates into the model OR total explained variations lost by going from the full model to the reduced model.

This idea can be generalized to develop a general method for testing hypotheses of the type:

$$H_0\colon (\beta_2, \ldots, \beta_k) = 0 \qquad vs \qquad (\beta_2, \ldots, \beta_k) \neq 0.$$

We complete the test as follows. Given a full model (which contains $\beta_1, \ldots, \beta_p$) and reduced model (which contains $\{\beta_1, \beta_{k+1}, \ldots, \beta_p\}$), define:

- $SSE_R - SSE_C = SSdrop$
- $dfE_R - dfE_C = dfdrop$
- $MSdrop = SSdrop/dfdrop$

Then the test statistic and p-value are given by: $TS = MSdrop/MSE_C$ and $\Pr\left(F_{dfdrop,dfE_C} \geq TS\right)$, respectively.

We can interpret $SSE_R - SSE_C$ as the extra info gained from adding the extra covariates into the model OR total explained variations lost by going from the full model to the reduced model. In addition, $dfE_R - dfE_C = k - 1$, or the number of covariates dropped from the full model to obtain the reduced model.

> **Note**
>
> If you take $k = 1$, then this is equivalent to the t-test!

3.3.10 Partial coefficient of determination

We can define the **partial coefficient of determination** as follows:

$$
\begin{aligned}
R^2(X_1, \dots, X_{k-1}|X_k, \dots, X_p) &= (SSE_R - SSE_C)/SSE_R \\
&= SSdrop/SSE_R.
\end{aligned}
$$

You might also see the partial correlation coefficient:

$$R(X_1, \dots, X_{k-1}|X_k, \dots, X_p) = \sqrt{R^2(X_1, \dots, X_{k-1}|X_k, \dots, X_p)}.$$

This quantity is the extra proportion of variation explained from adding the covariates $X_1, \dots, X_{k-1}$ to the model which already contains $X_k, \dots, X_p$.

Example 3.7. A researcher ran an experiment to see if YouTube, Facebook and newspaper ads would improve sales. Run the partial F test to see how online advertising affects sales. Compute and interpret the following quantities:

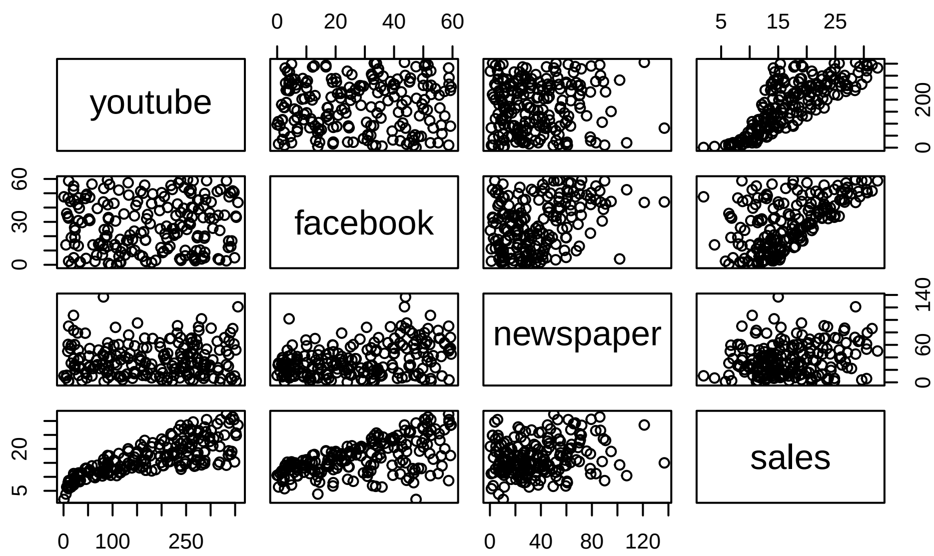
- $SSE_R - SSE_C = SSdrop$
- $dfE_R - dfE_C = dfdrop$
- $MSdrop = SSdrop/dfdrop$
- Test stat: $TS = MSdrop/MSE_C$
- p-value: $\Pr(F_{dfdrop,dfE_C} \geq TS)$
- Partial coefficient of determination

```r
# install.packages('datarium')
data("marketing", package = "datarium")
#printing out first few rows
```

```r
head(marketing, 4)
```

```
  youtube facebook newspaper sales
1  276.12    45.36     83.04 26.52
2   53.40    47.16     54.12 12.48
3   20.64    55.08     83.16 11.16
4  181.80    49.56     70.20 22.20
```

```r
plot(marketing)
```




```r
#setting n to be a variable (sample size)
n=nrow(marketing)


# Estimation: How to get an estimate $\hat\beta$ of $\beta$?
# lm( sales~    , data= marketing)
full_model<- lm(sales ~ youtube+facebook+newspaper, data = marketing)
summary(full_model)
```

```
Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5932  -1.0690   0.2902   1.4272   3.3951 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,	Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
summ=summary(full_model)

full_model$coefficients
```

```
 (Intercept)      youtube     facebook    newspaper 
 3.526667243  0.045764645  0.188530017 -0.001037493 
```

```r
MSE=summ$sigma^2

SSE_C=sum(summ$residuals^2)



# Inference: What is the error of $\hat\beta$? Is $f$ degenerate? I.e., is $\beta=0$?

#regular ANOVA
summary(full_model)
```

```
Call:
```

```
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max 
-10.5932  -1.0690   0.2902   1.4272   3.3951 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
youtube      0.045765   0.001395  32.809   <2e-16 ***
facebook     0.188530   0.008611  21.893   <2e-16 ***
newspaper   -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
#confidence intervals for beta coefficients
# confint.lm(full_model)

#Partial F Test
model_red=lm(sales ~ newspaper, data = marketing)
sum_reduced=summary(model_red)
MSER=sum_reduced$sigma^2
SSE_R=sum(sum_reduced$residuals^2)

SSdrop=SSE_R-SSE_C

MSEdrop=SSdrop/2
Fstat=MSEdrop/MSE

1-pf(Fstat,2,196)
```

```
[1] 0
```

```r
part_test=anova(model_red,full_model); part_test
```

```
Analysis of Variance Table
```

```
Model 1: sales ~ newspaper
Model 2: sales ~ youtube + facebook + newspaper
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    198 7394.1
2    196  801.8  2    6592.3 805.71 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
partial_c_det=SSdrop/SSE_R



SSER=sum(model_red$residuals*model_red$residuals); SSER
```

```
[1] 7394.119
```

```r
dfer=model_red$df.residual; dfer
```

```
[1] 198
```

```r
SSEC=sum(full_model$residuals*full_model$residuals); SSEC
```

```
[1] 801.8284
```

```r
dfeC=full_model$df.residual; dfeC
```

```
[1] 196
```

```r
SSdrop=SSER-SSEC; SSdrop
```

```
[1] 6592.29
```

```r
dfdrop=dfer-dfeC

MSdrop=SSdrop/dfdrop; MSdrop
```

```
[1] 3296.145
```

```r
R_online=SSdrop/SSER; R_online
```

```
[1] 0.8915586
```

```r
part_test$F
```

```
[1]       NA 805.7141
```

```r
part_test$`Pr(>F)`
```

```
[1]           NA 2.812622e-95
```

```r
# Prediction: Predict any values if necessary.
# What if we have a 300$ budget and we only can pick one advertising method?
new_data=marketing[1:3,1:3]
new_data[1:3,]=diag(300,3)
predict(full_model,new_data)
```

```
        1         2         3 
17.256061 60.085672  3.215419 
```

```r
# It's best to put our money in FB... meta?

# What about intervals?

predict(full_model,new_data, interval = 'confidence')
```

```
        fit         lwr       upr
1 17.256061 16.56191879 17.950203
2 60.085672 55.25061022 64.920734
3  3.215419 -0.09445737  6.525296
```

3.3.11 Another example

It's a good time to stop and do another example to review the topics covered so far.

Example 3.8. In the dataset `mtcars` we have the following variables:

- mpg: Miles/(US) gallon
- cyl: Number of cylinders
- disp: Displacement (cu.in.)
- hp: Gross horsepower
- drat: Rear axle ratio
- wt: Weight (1000 lbs)
- qsec: 1/4 mile time
- vs: V/S
- am: Transmission (0 = automatic, 1 = manual)
- gear: Number of forward gears
- carb: Number of carburetors

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). Overall, we would like to investigate the relationship between `mpg` and the following variables: `cyl, disp, hp, drat, wt, qsec, gear, carb.` Let's investigate the following questions:

1. Assume the normal MLR model. Store the covariate matrix and response in a variable. Fit a normal MLR model to the data. – That is use `lm()` to fit the model.
2. What are the least squares estimates? What is the MSE?
3. Generate the ANOVA table. Is the model significant?
4. Test if `drat` contributes anything to the model, adjusting for the other covariates. Test if `drat` is related to `mpg`, without adjusting for the other covariates.
5. Test if the subset of variables `gear, carb` contribute to the model jointly, adjusting for the remaining covariates. What is the partial coefficient of determination? Interpret the partial coefficient of determination. Test if the subset of variables `gear, carb` contribute to the model jointly, without adjusting for the remaining covariates.
6. Compute a confidence interval for the mean `mpg` of cars with the following set of covariate values 6.6, 176, 121, 4.29, 2.882, 18.106, 0, 1.1, 4.4, 4.4. Compute a prediction interval for the `mpg` of a car with the above set of covariate values.
7. Compute a confidence interval for the coefficient for `disp`.
8. Compute and interpret the coefficient of determination.

```r
data("mtcars")
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
dim(mtcars)
```

```
[1] 32 11
```

```r
# 1.
# response~all variables minus the two variables we will not include
model=lm(mpg~.-vs-am,data=mtcars)
summ=summary(model)
summ
```

```
Call:
lm(formula = mpg ~ . - vs - am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0230 -1.6874 -0.4109  0.9640  5.4400 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 17.88964   17.81996   1.004   0.3259  
cyl         -0.41460    0.95765  -0.433   0.6691  
disp         0.01293    0.01758   0.736   0.4694  
hp          -0.02085    0.02072  -1.006   0.3248  
drat         1.10110    1.59806   0.689   0.4977  
wt          -3.92065    1.86174  -2.106   0.0463 *
qsec         0.54146    0.62122   0.872   0.3924  
gear         1.23321    1.40238   0.879   0.3883  
carb        -0.25510    0.81563  -0.313   0.7573  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.622 on 23 degrees of freedom
Multiple R-squared:  0.8596,    Adjusted R-squared:  0.8107
F-statistic:  17.6 on 8 and 23 DF,  p-value: 4.226e-08
```

```r
X=model.matrix(model)
Y=mtcars$mpg
X[1:5,]
```

```
                  (Intercept) cyl disp  hp drat    wt  qsec gear carb
Mazda RX4                   1   6  160 110 3.90 2.620 16.46    4    4
Mazda RX4 Wag               1   6  160 110 3.90 2.875 17.02    4    4
Datsun 710                  1   4  108  93 3.85 2.320 18.61    4    1
Hornet 4 Drive              1   6  258 110 3.08 3.215 19.44    3    1
Hornet Sportabout           1   8  360 175 3.15 3.440 17.02    3    2
```

```r
Y
```

```
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
[31] 15.0 21.4
```

```r
# 2.
LSE=coef(model)
LSE
```

```
(Intercept)         cyl        disp          hp        drat          wt 
17.88963741 -0.41459575  0.01293240 -0.02084886  1.10109551 -3.92064847 
       qsec        gear        carb 
 0.54145693  1.23321026 -0.25509911 
```

```r
MSE=summ$sigma^2
MSE
```

```
[1] 6.874941
```

```r
# 3.
null_model=lm(mpg~1,data=mtcars)
anova(null_model,model)
```

```
Analysis of Variance Table

Model 1: mpg ~ 1
Model 2: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     31 1126.05
2     23  158.12  8    967.92 17.599 4.226e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 4.
# Notice the p value is 0.5 , not sign.
summ$coefficients['drat',]
```

```
  Estimate Std. Error    t value   Pr(>|t|)
 1.1010955  1.5980601  0.6890201  0.4977032
```

```r
drat=lm(mpg~drat,,data=mtcars)
# Notice the p value is 1.78e-05 , sig! explain this difference!
summary(drat)
```

```
Call:
lm(formula = mpg ~ drat, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0775 -2.6803 -0.2095  2.2976  9.0225

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.525      5.477  -1.374     0.18
drat           7.678      1.507   5.096 1.78e-05 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.485 on 30 degrees of freedom
Multiple R-squared:  0.464, Adjusted R-squared:  0.4461
F-statistic: 25.97 on 1 and 30 DF,  p-value: 1.776e-05
```

```
# 5.
red_model=lm(mpg~.-vs-am-gear-carb,data=mtcars)
anova(red_model,model)
```

```
Analysis of Variance Table

Model 1: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am - gear - carb
Model 2: mpg ~ (cyl + disp + hp + drat + wt + qsec + vs + am + gear +
    carb) - vs - am
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     25 163.48
2     23 158.12  2    5.3532 0.3893 0.6819
```

```
ob=anova(red_model,model)
ob$`Sum of Sq`[2]/ob$RSS[1]
```

```
[1] 0.0327457
```

```
# 3.7% of the variation in mpg is explained from adding the covariate gear and carb to the

# 6.
new_ob=c(6.6,176,121,4.29,2.882,18.106,0,1.1,4.4,4.4)
new_ob=matrix(new_ob,nrow=1,ncol=length(new_ob))
colnames(new_ob)=names(mtcars[1,-1])
new_ob=data.frame(new_ob)
predict(model,new_ob, interval = 'confidence')
```

```
       fit      lwr      upr
1 22.43839 18.49468 26.38211
```

```
predict(model,new_ob, interval = 'prediction')
```

```
       fit     lwr      upr
1 22.43839 15.7322 29.14459
```

```
# 7.
confint(model)
```

```
                   2.5 %      97.5 %
(Intercept) -18.97375462 54.75302945
cyl          -2.39565252  1.56646102
disp         -0.02343129  0.04929609
hp           -0.06371601  0.02201829
drat         -2.20474377  4.40693480
wt           -7.77195651 -0.06934042
qsec         -0.74362628  1.82654014
gear         -1.66782660  4.13424711
carb         -1.94235037  1.43215215
```

```
#8.
summ$r.squared
```

```
[1] 0.8595764
```

```
# 85% of the variation in mpg is explained by cyl, disp, hp, drat, wt, qsec, gear and carb
```

Exercise 3.18. Interpret all of the above quantites.

3.4 Checking model assumptions

We learned how to test significance of one or multiple variables, compute confidence intervals for the estimated coefficients, mean response, and predicted response. All the methods rely on the assumptions! Recall that we assume 1. The relationship is linear $Y|X = X\beta + \epsilon$, 2. $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ 3. $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.

We now briefly discuss how to use the data to check if these assumptions are appropriate. We will cover this in more detail in the next chapter.

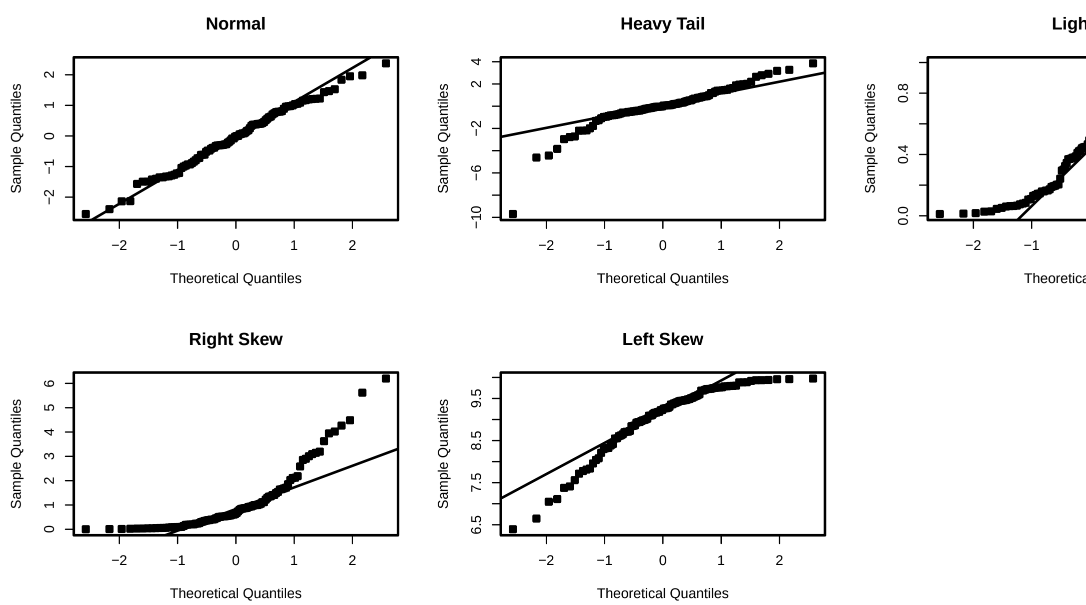
3.4.1 Checking normality

We do not know ϵ, however, we do know $\hat{\epsilon}$, which is our best proxy for the true random error vector ϵ. To check if the true random error vector is normally distributed we can use histograms and quantile-quantile plots. More specifically, if the histogram of the residuals looks more or less bell-shaped, with tails similar to the normal PDF, then the assumption of normality is valid.

Recall that a qq-plot compares the quantiles of the sample to the quantiles of the theoretical normal distribution. The x-axis represents the theoretical quantiles. The y-axis represents the sample quantiles. If the sample follows a normal distribution, the points in the qq-plot will approximately lie on a line.

Interpretation:

- Straight Line: If the points lie on or near the straight line, the sample appears normal.
- Heavy Tails: Points deviating upwards or downwards at the ends suggest the sample has heavier or lighter tails than the normal distribution.
- S-Shape: Points forming an S-shape indicate the sample has lighter tails and a heavier center than the normal distribution.



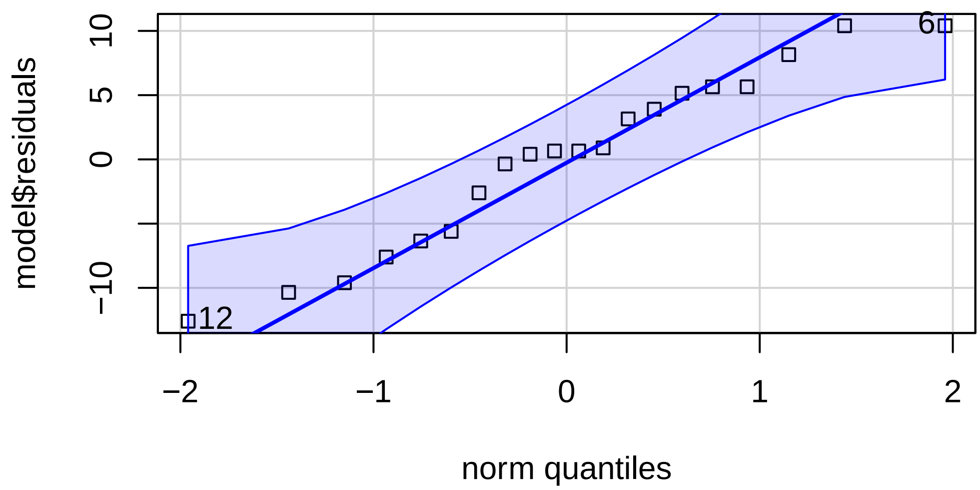
See below for an example:
Note that you will always have some deviation at the ends of the line in the qq-plot.

Example 3.9. In examples Example 3.1 and Example 3.7, check that the normality assumption is valid.

```r
# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))
model= lm(BodyFat ~Weight, data = df)
summary(model)
```

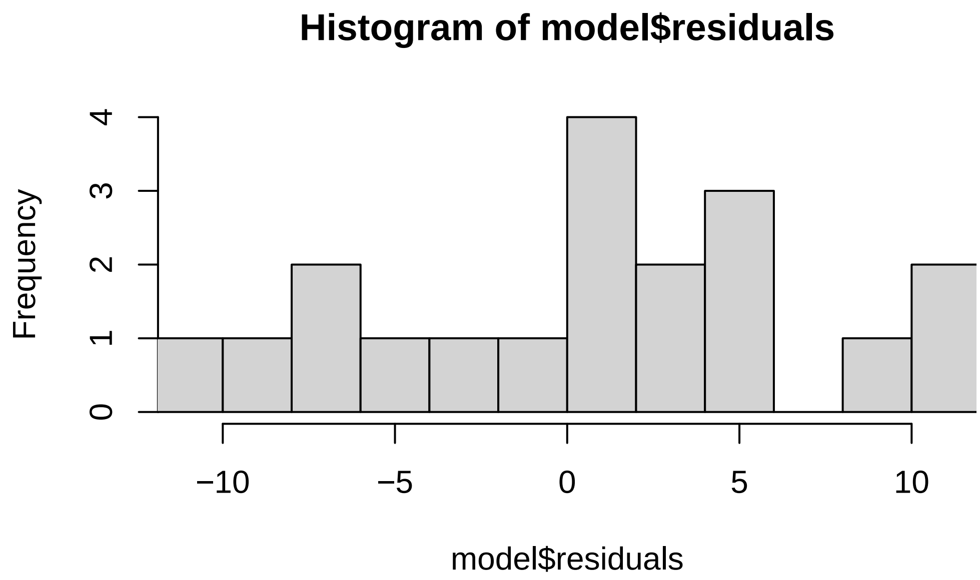
```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5935  -5.7904   0.6536   5.2731  10.4004

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.37626   11.54743  -2.371 0.029119 *
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,    Adjusted R-squared:  0.4567
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```r
car::qqPlot(model$residuals,pch=22)
```




```
[1] 12  6
```

```r
hist(model$residuals,breaks=10,xlim=c(-11,11))
```

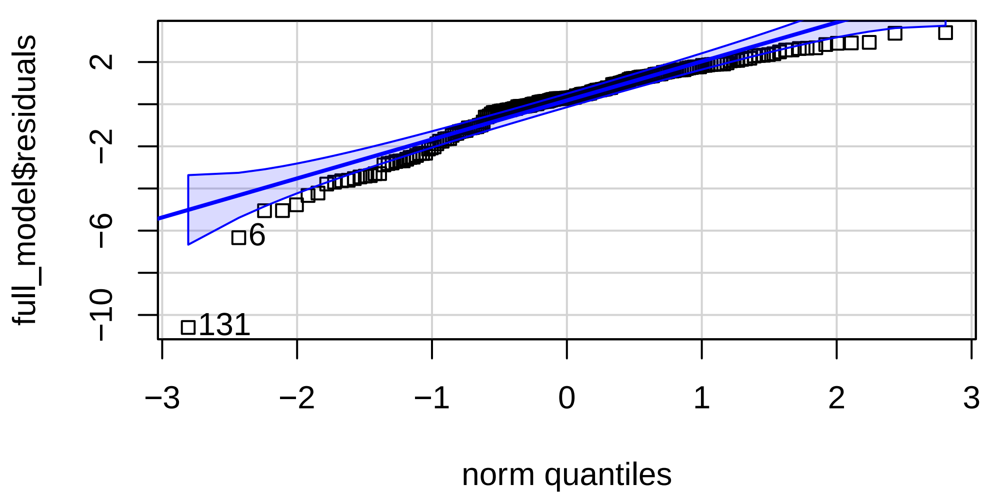



```r
# This appears okay!

# Let's do the next example
# install.packages('datarium')
data("marketing", package = "datarium")


# lm( sales~    , data= marketing)
full_model<- lm(sales ~ youtube+facebook+newspaper, data = marketing)
summary(full_model)
```

```
Call:
lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max
-10.5932  -1.0690   0.2902   1.4272   3.3951

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.526667   0.374290   9.422   <2e-16 ***
```

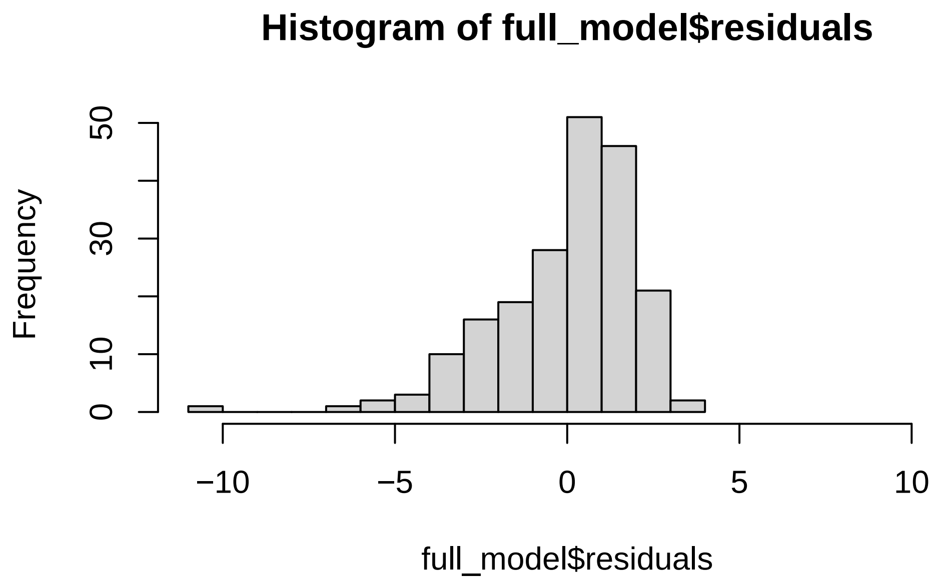
```
youtube       0.045765   0.001395  32.809   <2e-16 ***
facebook      0.188530   0.008611  21.893   <2e-16 ***
newspaper    -0.001037   0.005871  -0.177     0.86
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.023 on 196 degrees of freedom
Multiple R-squared:  0.8972,	Adjusted R-squared:  0.8956
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

```r
# Not great.
car::qqPlot(full_model$residuals,pch=22)
```




```
[1] 131   6
```

```r
hist(full_model$residuals,breaks=10,xlim=c(-11,11))
```



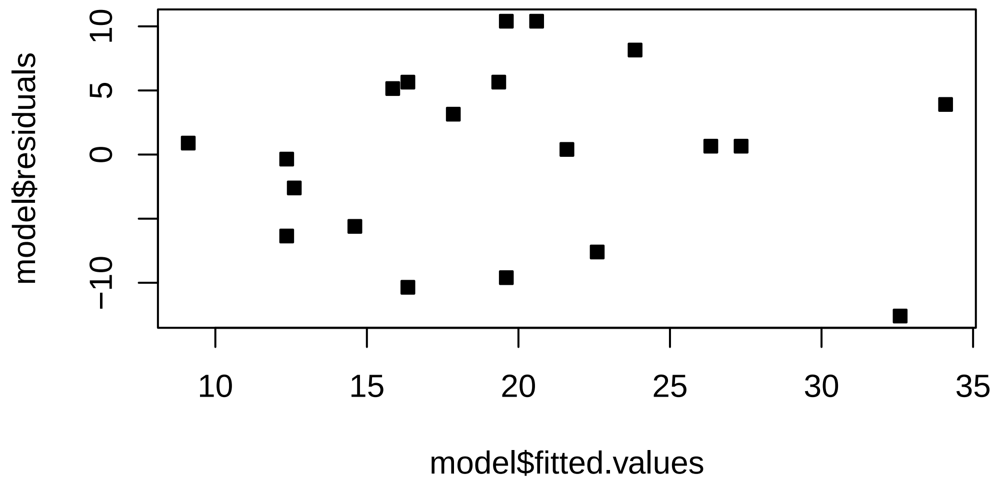
3.4.2 Checking the other assumptions

To check the remaining assumptions (constant variance, independence of residuals, zero mean and linear relationship), we can use some other diagnostic plots.

One plot is that of the fitted values $\widehat{Y}$ (x-axis) against the residuals $\hat{\epsilon}$ (y-axis). If the error depends on $\hat{y}$, then the identically distributed assumption on the errors is probably not valid. If the assumptions are valid, we should observe on the plots that at all levels of the response, the mean of the residuals is 0 and the variance remains the same. Thus, we should see a horizontal band centered at 0 containing the observations.

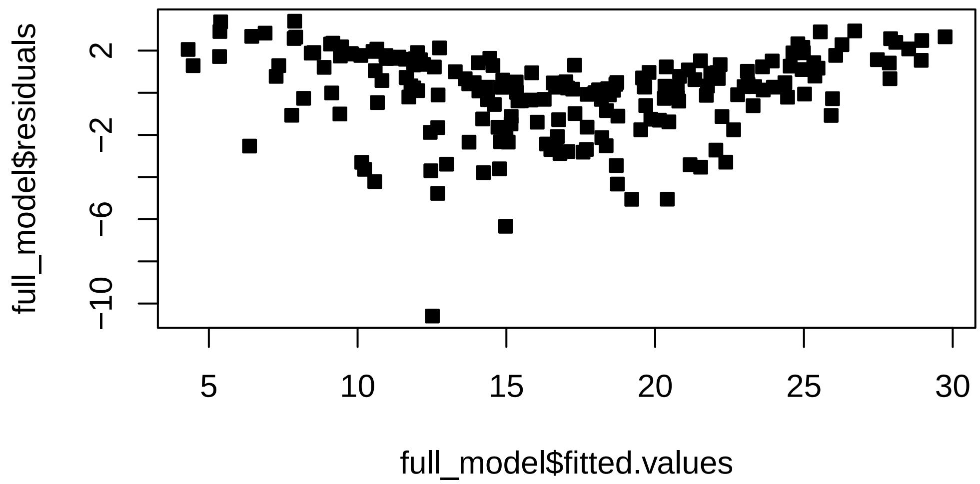
This appears to be the case in the body fat example:

```
plot(model$fitted.values,model$residuals,pch=22,bg=1)
```



Observe that in the marketing example, the residuals admit a pattern. This usually indicates either a non-linear relationship with the covariates, or an important covariate is missing. In this case, we would say the assumption of identically distributed errors is violated.

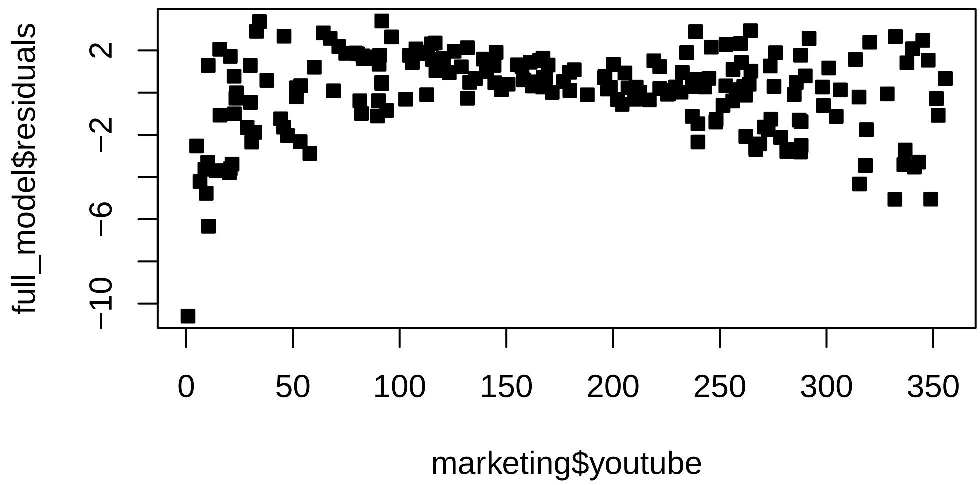
```
plot(full_model$fitted.values,full_model$residuals,pch=22,bg=1)
```



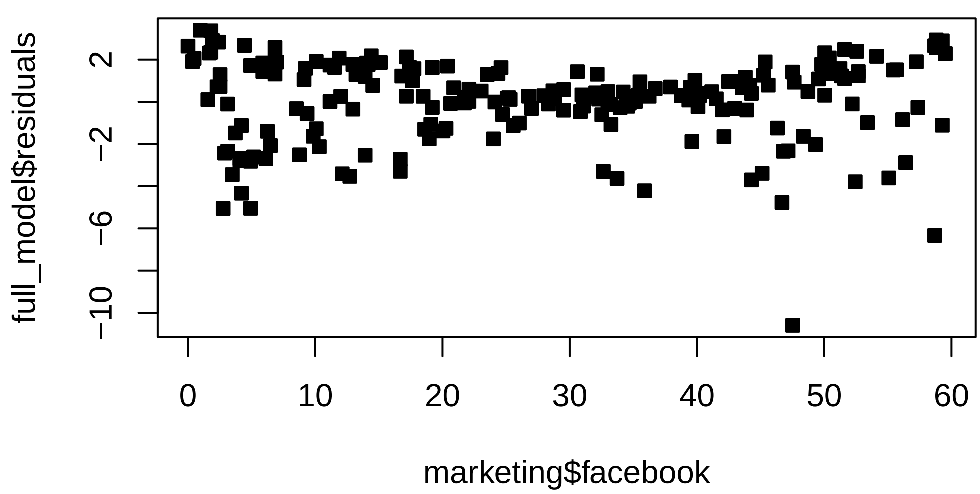
Plotting the residuals against the covariates can reveal dependence between the errors. For instance, if time is a covariate, you can plot the residuals over time to see if they have any relationship with time. If there appears to be dependence among the residuals, then the assumptions of the model are violated. That is, in these plots we should also see a horizontal band centered at 0 containing the observations. If not, then the residuals have a relationship with the given covariate.

Be VERY careful about the scale of your plot, as it can affect your interpretation. Zooming out or in too much can make everything look fine. In addition, the y-axis not being centered at 0 can cause you to misinterpret the plot.

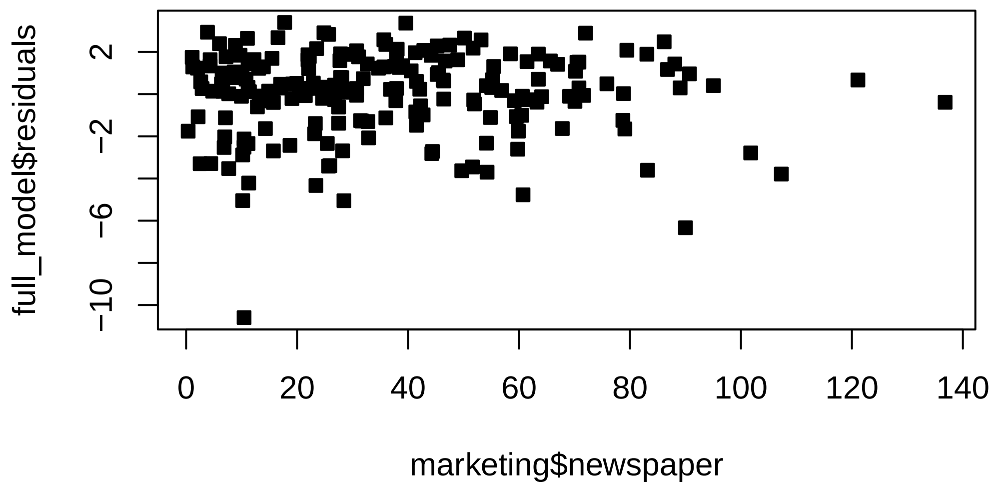
```
plot(marketing$youtube,full_model$residuals,pch=22,bg=1)
```



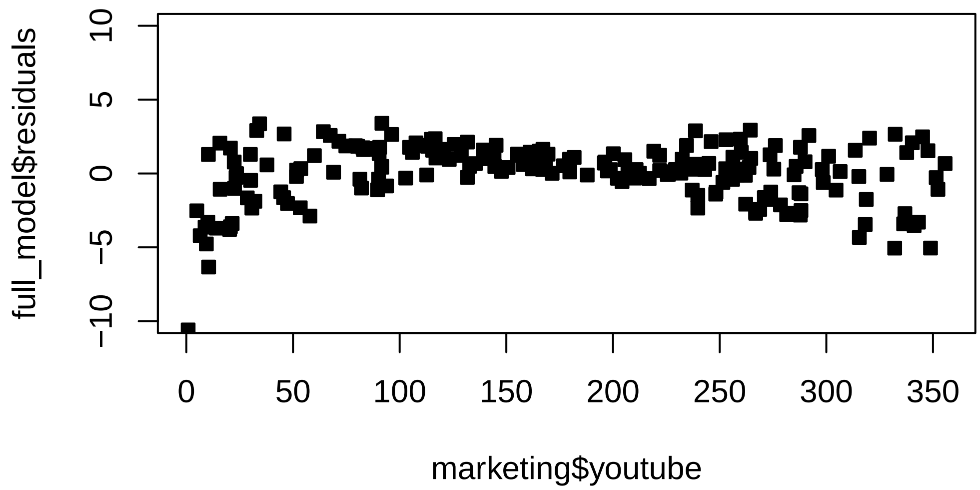

```
plot(marketing$facebook,full_model$residuals,pch=22,bg=1)
```



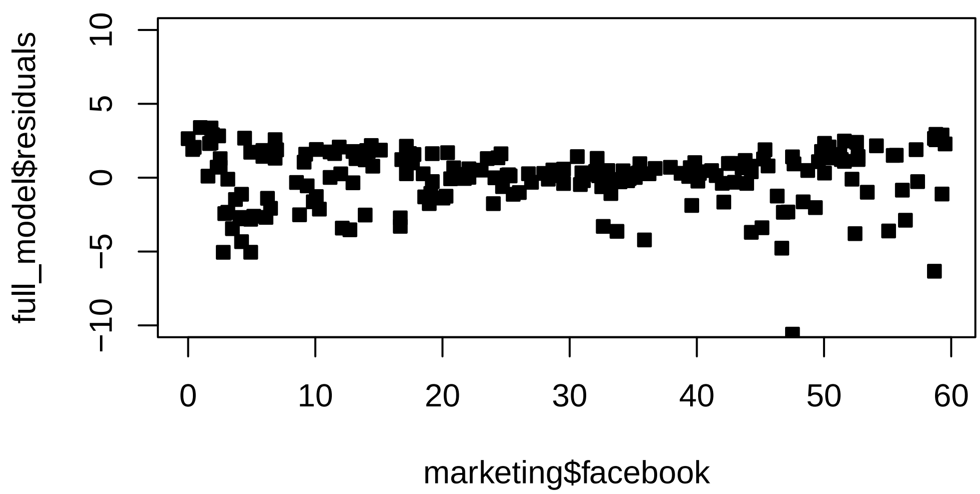
```r
plot(marketing$newspaper,full_model$residuals,pch=22,bg=1)
```



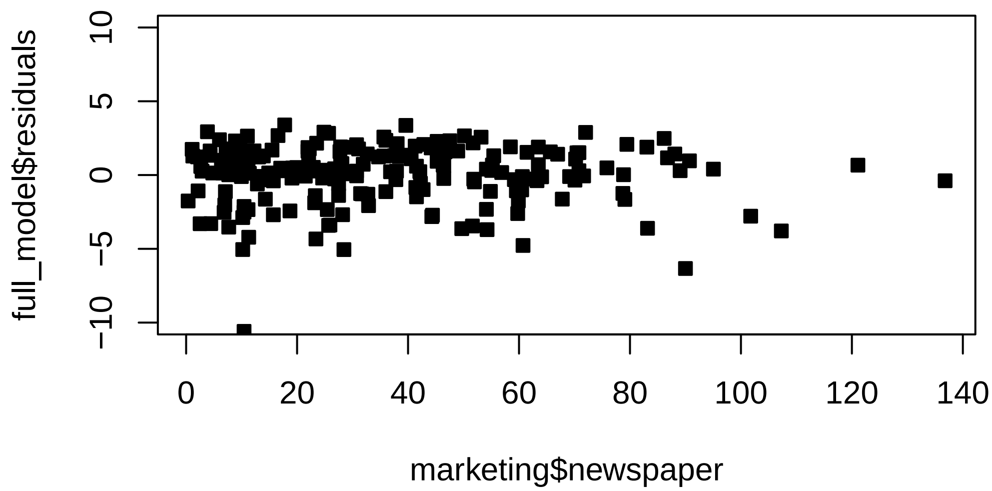

```r
plot(marketing$youtube,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```




```
plot(marketing$facebook,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```



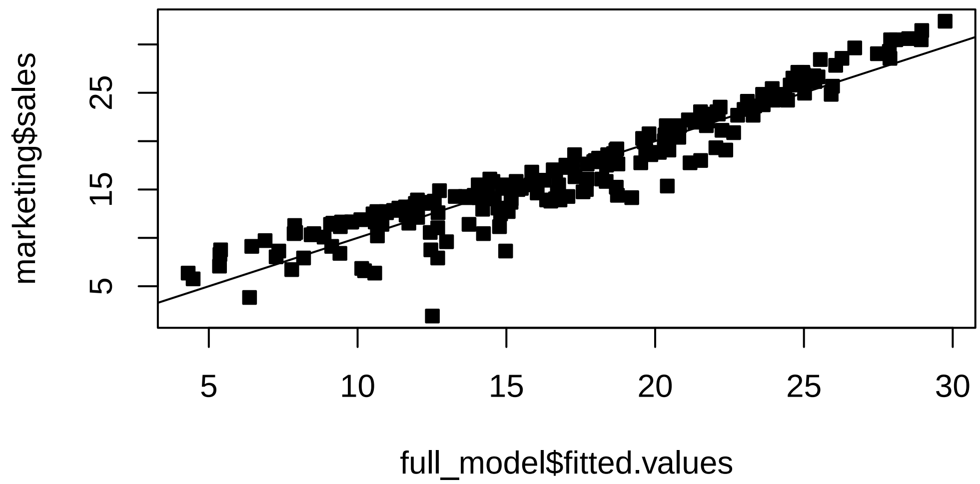
```r
plot(marketing$newspaper,full_model$residuals,pch=22,bg=1,ylim=c(-10,10))
```



Notice how the newspaper plot changes with the new axis limits. It appears that the variance of the error is changing with the value of the Facebook and Youtube budgets.

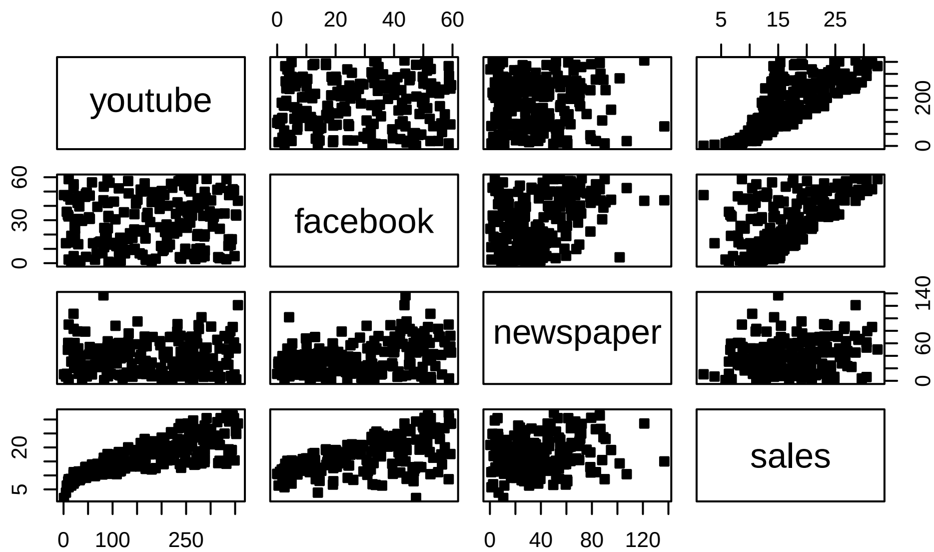
Another plot is that of the fitted values against the response. This gives an idea of the overall fit of the model. We should observe the points scatters around the line $y = x$.

```r
plot(full_model$fitted.values,marketing$sales,pch=22,bg=1)
abline(0,1)
```



Notice that the line is slightly curved above the line at the ends. This means that at high and low values, the actual sales are empirically greater than as predicted by the model. Let's plot the actual data.

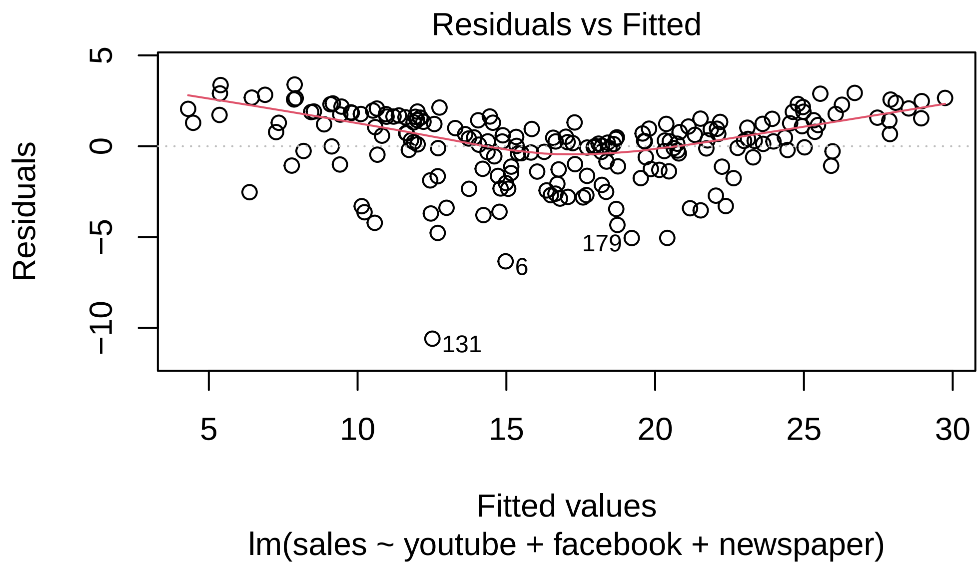
```
plot(marketing,pch=22,bg=1)
```

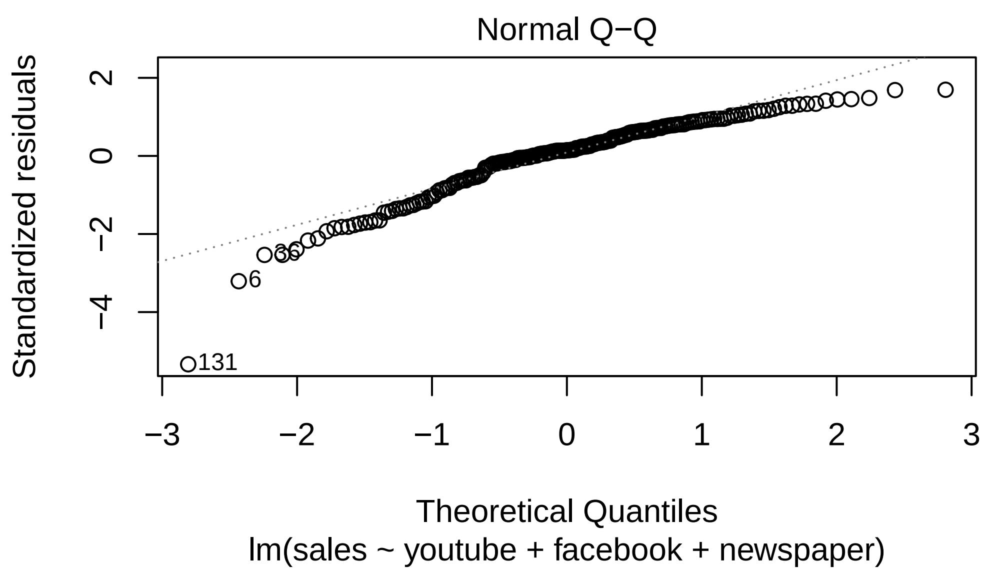


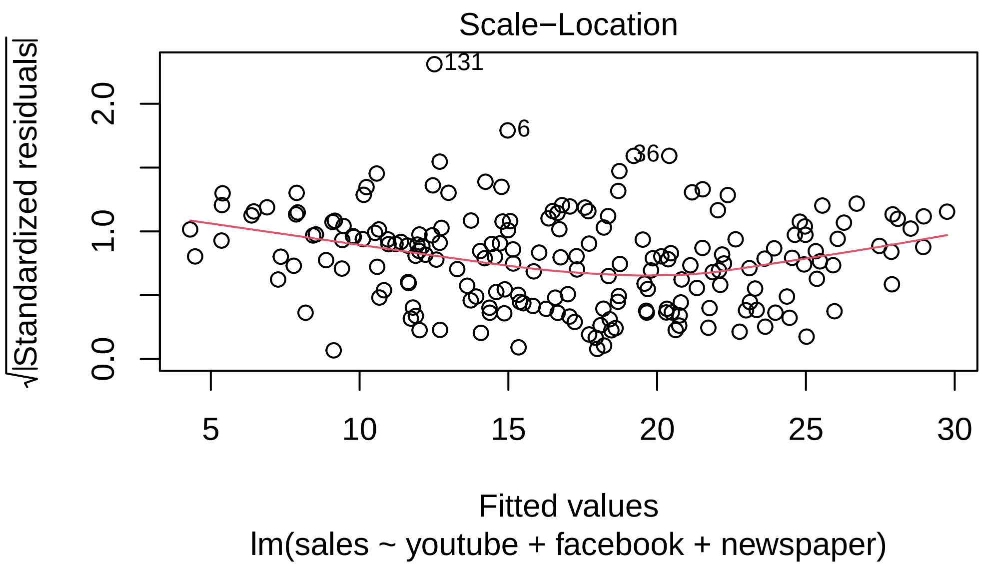
In this case, Youtube and Facebook spending seems to have a nonlinear relationship with sales. We will see how to remedy this in later chapters.

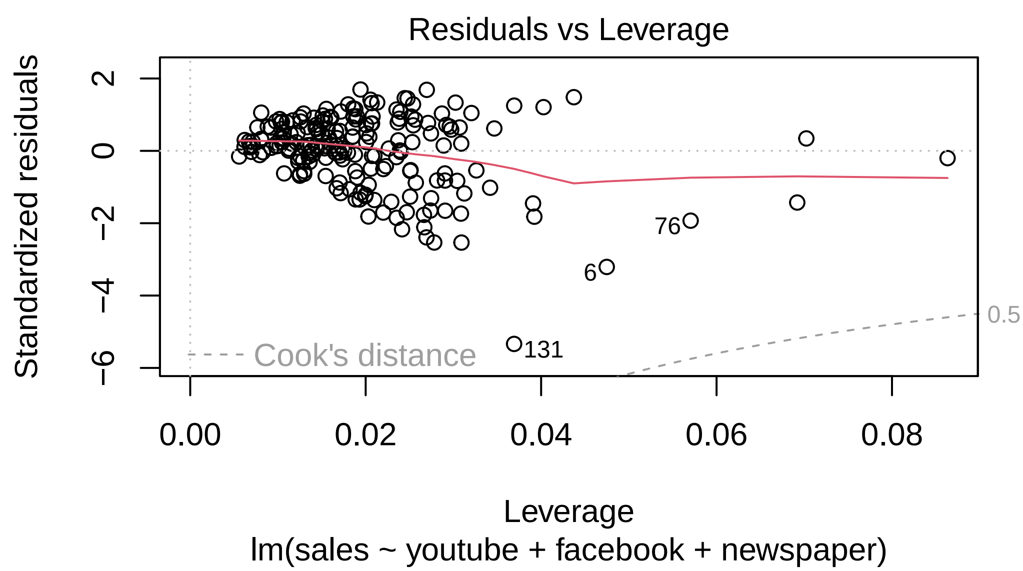
As a final note, observe that we can put the `model` object in the `plot()` function to obtain the diagnostic plots.

```
plot(full_model)
```


Residuals vs Fitted
Residuals
5
0
−5
−10
179
6
131
5
10
15
20
25
30
Fitted values
lm(sales ~ youtube + facebook + newspaper)


Normal Q−Q
Standardized residuals
2
0
−2
−4
6
131
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(sales ~ youtube + facebook + newspaper)





We will learn in later chapters how to check the assumptions more thoroughly and how to remedy violations of the assumptions.

3.4.3 Homework stop 5

Complete the assigned textbook problems for Chapter 4.

Exercise 3.19. List the assumptions for the normal MLR model and the MLR model. Write down how you would check each assumption.

3.5 Simple linear regression

A special case of the multiple linear regression is **simple linear regression**. A simple linear regression model is a regression model with **one explanatory variable**: $Y_i = \beta_0+\beta_1 X_i+\epsilon_i$.

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

3.5.1 Estimated Coefficients

In this case, following some matrix manipulations (verify this for homework), we have

$$X^\top X = \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix}, \ X^\top y = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix}.$$

Now, recall if

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$A^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

From MATH 1131 (or simple algebraic manipulation), we know

$$\sum_{i=1}^n (x_i-\bar{x})(y_i-\bar{y}) = \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} = \sum_{i=1}^n x_i y_i - n^{-1}\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n y_i\right)$$

$$\sum_{i=1}^n (x_i-\bar{x})^2 = \sum_{i=1}^n x_i^2 - n\bar{x}^2 = \sum_{i=1}^n x_i^2 - n^{-1}\left(\sum_{i=1}^n x_i\right)^2.$$

Therefore

$$
\begin{aligned}
(X^\top X)^{-1} &= \frac{1}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix} \\
&= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix}.
\end{aligned}
$$

To summarize:

$$
\begin{aligned}
X^\top X &= \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \\
X^\top y &= \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix} \\
(X^\top X)^{-1} &= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}
\end{aligned}
$$

Now, we have that

$$
\begin{aligned}
\hat{\beta} &= \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = (X^\top X)^{-1} X^\top y \\
&= \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i \\ -\sum_{i=1}^n x_i \sum_{i=1}^n y_i + n\sum_{i=1}^n x_i y_i \end{pmatrix}.
\end{aligned}
$$

Now,

$$
\hat{\beta}_1 = \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \left(-\sum_{i=1}^n x_i \sum_{i=1}^n y_i + n\sum_{i=1}^n x_i y_i \right).
$$

Exercise 3.20. Let's show that

$$
\hat{\beta}_1 = \frac{n\sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}.
$$

Does this look familiar? We see that,

$$
\hat{\beta}_1 = \widehat{\mathrm{cov}}(X, Y) \frac{\hat{\sigma}_y}{\hat{\sigma}_x},
$$

where $\widehat{\mathrm{cov}}(X, Y)$ is the estimated correlation between X and Y. Let's interpret this:

1. If $\widehat{\mathrm{cov}}(X, Y) \approx 0$ then $\hat{\beta}_1 \approx 0$ - low correlation implies an estimated slope close to 0.
2. The estimated coefficient $\hat{\beta}_1$ is the product of the estimated correlation between X and Y and the ratio of the estimated standard deviation of Y to that of X.

Now, looking at the intercept term, we have

$$\hat{\beta}_0 = \frac{1}{n\sum_{i=1}^n (x_i - \bar{x})^2} \left(\sum_{i=1}^n x_i^2 \ \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \ \sum_{i=1}^n x_i y_i \right).$$

Exercise 3.21. Show that

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

Observe that the intercept is the mean of Y minus the mean of X times the estimated slope. In essence, it tells us that the intercept $(\hat{\beta}_0)$ represents the value of (X) when (X) is at its mean value $((\bar{X}))$ and that $(\bar{X})$ is adjusted by subtracting the contribution of $(\hat{\beta}_1 \bar{X})$.

This adjustment ensures that the regression line passes through the point $((\bar{X}, \bar{X}))$, which is the point of averages for the data.

3.5.2 Inference in SLR

We can also simplify the values used for inference in the SLR model. Recall that $\text{Var}\left[\hat{\beta}\right] = (X^\top X)^{-1}\sigma^2$, and so we have

$$\begin{aligned}
\text{Var}\left[\hat{\beta}_0\right] &= \frac{\sum_{i=1}^n x_i^2}{n\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 = \frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2 + n\bar{x}^2}{n\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 \\
&= \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right]\sigma^2 \\
\text{Var}\left[\hat{\beta}_1\right] &= \frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2 \\
\text{cov}(\hat{\beta}_0, \hat{\beta}_1) &= -\frac{\bar{x}}{\sum_{i=1}^n (x_i - \bar{x})^2}\sigma^2.
\end{aligned}$$

We know from previous sections that a $(1-\alpha)100$ confidence interval of β_i, where $i = 0, 1$, is

$$\hat{\beta}_i \pm t_{df_E, \alpha/2}\sqrt{\widehat{var}(\hat{\beta}_i)}.$$

Similarly, let β_i^0 be a hypothesized value of β_i, for $i = 0, 1$. If we want to test whether $\beta_i = \beta_i^0$, then the observed test statistic is given by

$$\frac{\hat{\beta}_i - \beta_i^0}{\sqrt{\widehat{var}(\hat{\beta}_i)}},$$

and the corresponding p-value is obtained via the t_{dfE} distribution as usual.

> **Note**
>
> Similarly, inference for the mean response and predictions can be obtained. We can also simplify the $ANOVA$ table, R^2, etc. For instance, the R^2 is the square of the sample correlation coefficient between X and Y.

3.5.3 Inference for the correlation coefficient

If we are interested in doing a hypothesis test, or constructing confidence intervals for the correlation between two variables, say X and Y, we can use the simple linear regression model.

We have already derived the relationship between the estimated correlation coefficient and the estimated slope of the simple linear regression model. More specifically, if the estimated correlation coefficient is 0, then the estimated slope of the simple linear regression is 0. One can show that the same relationship holds at the population level: $\beta_1 = \rho\sigma_y/\sigma_x$, where $\rho = \text{corr}\,[X, Y]$.

Now, suppose that we want to test if $H_0 : \rho = 0$ versus $H_a : \rho \neq 0$. Using the fact that $\beta_1 = \rho\sigma_y/\sigma_x$, the above test is equivalent to the statement $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 \neq 0$. Therefore, we can just test if the slope parameter in the model $Y|X = \beta_0 + \beta_1 X + \epsilon$ is 0.

Letting $\hat{\rho} = \widehat{\text{corr}}(X, Y)$ The observed test statistic is then:

$$\frac{\hat{\beta}_1}{\sqrt{\widehat{var}(\hat{\beta}_1)}} = \frac{\hat{\rho}\sqrt{n-2}}{\sqrt{1-\hat{\rho}^2}},$$

and the corresponding p-value is obtained based on the t_{dfE} distribution.

However, when the hypothesized value for ρ is non-zero, the problem becomes very complicated. The exact distribution of $\hat{\rho}$ is extremely difficult to obtain under the null hypothesis. The following procedure gives an approximation of the distribution of a function of $\hat{\rho}$ under the null hypothesis. In particular, Fisher suggested the transformation for $\rho \in (0, 1)$,

$$\theta = \frac{1}{2}\log\frac{1+\rho}{1-\rho}.$$

Then

$$\hat{\theta} = \frac{1}{2}\log\frac{1+\hat{\rho}}{1-\hat{\rho}},$$

is an estimate of θ, where $\hat{\theta}$ is approximately distributed as normal with mean θ and variance $\frac{1}{n-3}$. Hence, an approximate $(1-\alpha)100$ confidence interval of θ is

$$\hat{\theta} \pm z_{\alpha/2}\sqrt{\frac{1}{n-3}},$$

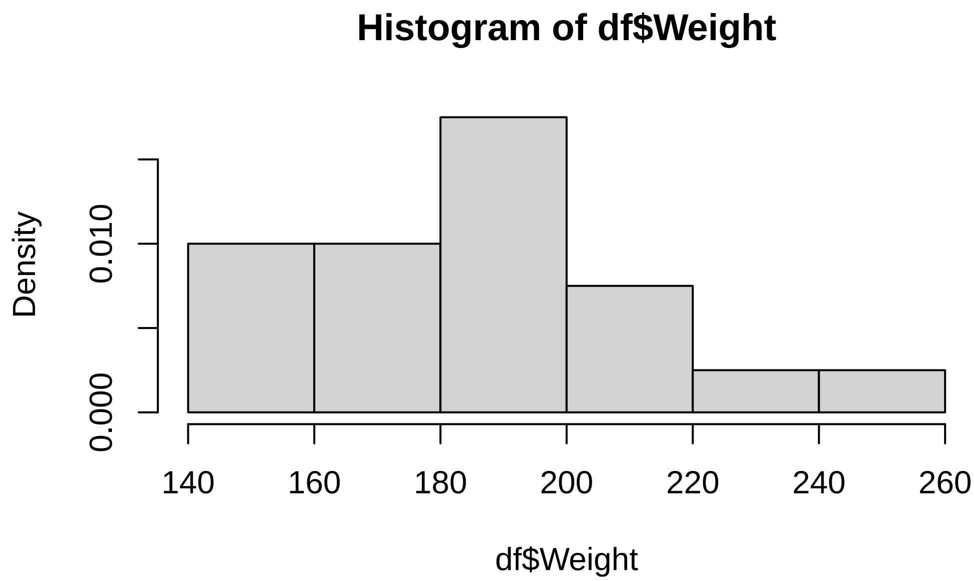
and the corresponding confidence interval of ρ can be obtained by the inverse transformation. Similarly, if the hypothesized value of ρ is ρ_0, then the hypothesized value of θ is $\theta_0 = \frac{1}{2}\log\frac{1+\rho_0}{1-\rho_0}$. The observed test statistic can be obtained and the corresponding p-value can be obtained based on the standard normal distribution.

Example 3.10. In Example 3.1 test if the correlation between body fat and weight is 0. Next, test if the correlation is greater than 1/2. Construct a 95% CI for ρ.

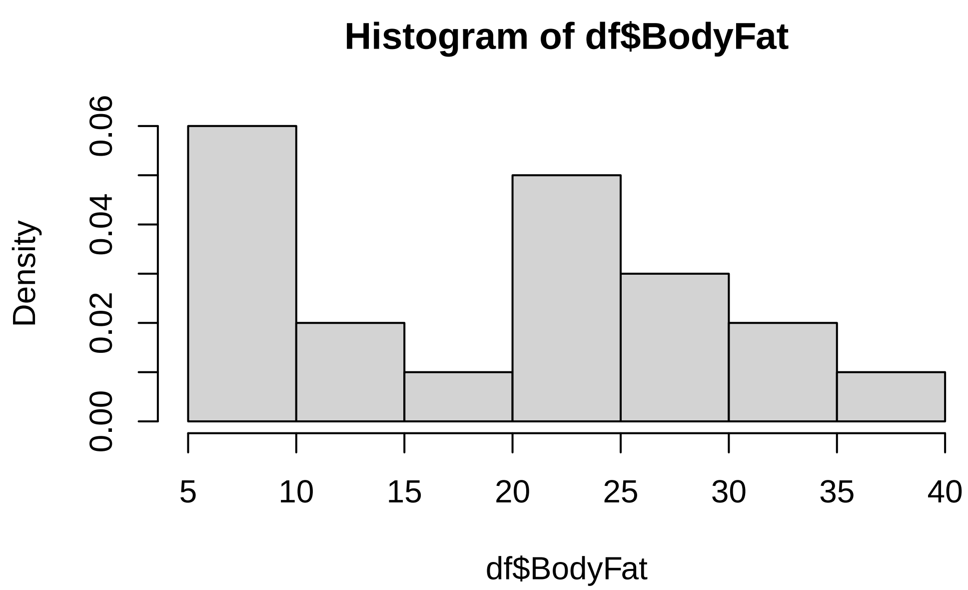
```
############################ Exploratory
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 ,
         173 , 187 , 188 , 188 , 240 , 175 , 168 ,
         246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 ,
           30 , 10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )

df=data.frame(cbind(Weight=Weight,BodyFat=BodyFat))


cor(df)
```

```
           Weight   BodyFat
Weight  1.0000000 0.6966328
BodyFat 0.6966328 1.0000000
```

```
hist(df$Weight,freq=F)
```




```
hist(df$BodyFat,freq=F)
```



```r
summary(df)
```

```
     Weight          BodyFat     
 Min.   :146.0   Min.   : 6.00  
 1st Qu.:171.8   1st Qu.:10.00  
 Median :187.5   Median :21.00  
 Mean   :188.6   Mean   :19.75  
 3rd Qu.:201.2   3rd Qu.:27.25  
 Max.   :246.0   Max.   :38.00  
```

```r
cor(df)[1,2]
```

```
[1] 0.6966328
```

```r
X=cbind(rep(1,nrow(df)), df$Weight)
Y=df$BodyFat

beta_hat= solve(t(X)%*%X)%*%t(X)%*%Y
beta_hat
```

```
            [,1]
[1,] -27.3762623
[2,]   0.2498741
```

```r
model=lm(BodyFat~ Weight,df)
model
```

```

Call:
lm(formula = BodyFat ~ Weight, data = df)

Coefficients:
(Intercept)       Weight  
   -27.3763       0.2499  
```

```r
summary(model)
```

```
Call:
lm(formula = BodyFat ~ Weight, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-12.5935  -5.7904   0.6536   5.2731  10.4004

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -27.37626   11.54743  -2.371 0.029119 *
Weight        0.24987    0.06065   4.120 0.000643 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.049 on 18 degrees of freedom
Multiple R-squared:  0.4853,    Adjusted R-squared:  0.4567
F-statistic: 16.97 on 1 and 18 DF,  p-value: 0.0006434
```

```r
cor(df)[1,2]^2
```

```
[1] 0.4852972
```

```r
cor(df)[1,2]
```

```
[1] 0.6966328
```

```r
a=function(x){
  (exp(2*x)-1)/(exp(2*x)+1)
}
a(1.36)
```

```
[1] 0.8763931
```

3.5.4 Homework stop 6

- Complete the Chapter 2 questions in the textbook.

Exercise 3.22. For

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

- Compute $\hat{\beta}$, Var $\left[\hat{\beta}_1\right]$, Var $\left[\hat{\beta}_0\right]$, cov $[(] \hat{\beta}_0, \hat{\beta}_1)$
- Show $\hat{\beta}_1 = r\frac{\hat{\sigma}_y}{\hat{\sigma}_x}$

3.6 Additional concepts & examples

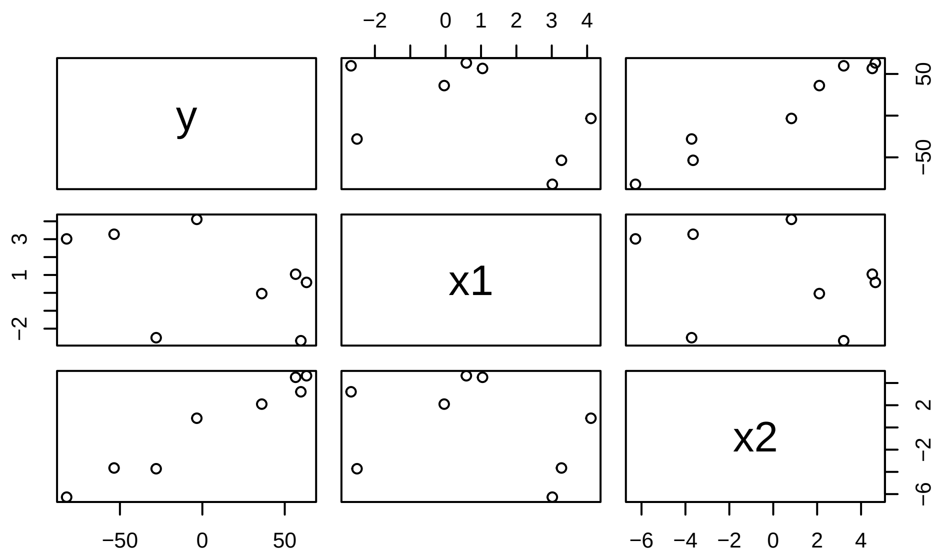
Here we touch on a few important examples and notes about the MLR.

3.6.1 Beware scatter plots in MLR

Sometimes, scatter plots are misleading for determining the relationship between Y and a collection of p covariates. In the following example, it appears that $X1$ and Y do not have a relationship, when in fact they do. Generally, this phenomena goes away with higher sample sizes.

```
# Scatter diagram beware?
# x1=c(2,3,4,1,5,6,7,8)
# x2=c(2,3,4,1,5,6,7,8)
# x=c(2,3,4,1,5,6,7,8)

# x1=1:8
# x2=c(2,1:6,4)
# y=8-5*x1+12*x2+rnorm(8,0,2)

set.seed(445)




n=8
x1=rnorm(n,5,5)
x2=rnorm(n,3,5)
y=8-5*x1+12*x2+rnorm(n,0,2)

df=data.frame(cbind(y,x1,x2))
```

```r
plot(df)
```



Next, we do an example from the textbook, which uses the NFL data. Specifically, we try to evaluate the relationship between number of wins and several explanatory variables.

Example 3.11. Using the following NFL data, complete 3.1-3.4, 4.1 and 4.2 in the textbook.

```r
################## NFL example #################
# This gives you the data sets used in the textbook
# install.packages('MPV')
df=MPV::table.b1
# Note for more information, run ?MPV::table.b1

head(df)
```

```
   y   x1   x2   x3   x4 x5  x6   x7   x8   x9
1 10 2113 1985 38.9 64.7  4 868 59.7 2205 1917
2 11 2003 2855 38.8 61.3  3 615 55.0 2096 1575
3 11 2957 1737 40.1 60.0 14 914 65.6 1847 2175
4 13 2285 2905 41.6 45.3 -4 957 61.4 1903 2476
5 10 2971 1666 39.2 53.8 15 836 66.1 1457 1866
```

```
6 11 2309 2927 39.7 74.1  8 786 61.0 1848 2339
```

```r
# names too long
names(df)
```

```
 [1] "y"  "x1" "x2" "x3" "x4" "x5" "x6" "x7" "x8" "x9"
```

```r
# rename to make it easier
names(df)=c("Wins","RushY","PassY","PuntA","FGP","TurnD","PenY","PerR","ORY","OPY")
names(df)
```

```
 [1] "Wins"  "RushY" "PassY" "PuntA" "FGP"   "TurnD" "PenY"  "PerR"  "ORY"
[10] "OPY"
```

```r
# Wins~ beta_1+beta_2Passing yrds+beta_3per_rush+beta_4ORY+epsilon
# summary(df)
# plot(df)
# run the model
regression_model=lm(  Wins ~ PassY+PerR+ORY  ,data= df )
summary(regression_model)
```

```
Call:
lm(formula = Wins ~ PassY + PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0370 -0.7129 -0.2043  1.1101  3.7049

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.808372   7.900859  -0.229 0.820899
PassY        0.003598   0.000695   5.177 2.66e-05 ***
PerR         0.193960   0.088233   2.198 0.037815 *
ORY         -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.706 on 24 degrees of freedom
```

```
Multiple R-squared:  0.7863,    Adjusted R-squared:  0.7596 
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

```
# model=lm(Wins~PassY+PerR+ORY,data=df)
# get the confidence intervals.
confint(regression_model)
```

```
                   2.5 %       97.5 %
(Intercept) -18.114944410 14.498200293
PassY         0.002163664  0.005032477
PerR          0.011855322  0.376065098
ORY          -0.007451027 -0.002179961
```

What conclusions can you make from this output? - All variables seem important! For isntance, we sdee that for every 1% increase in percentage rushing, there is a 0.193960 increase in number of wins, on average, holding passing yards and opponent rushing yards constant.

```
#### CI
# mean response of z'\beta , z=(2000,60,1900)'
new_data=data.frame(  matrix(c(2000,60,1900),ncol=3) )
names(new_data)
```

```
[1] "X1" "X2" "X3"
```

```
names(new_data)=c( 'PassY','PerR','ORY' )

predict(regression_model, new_data  ,   interval = 'confidence')
```

```
       fit      lwr      upr
1 7.875942 7.072672 8.679213
```

```
predict(regression_model, new_data  ,   interval = 'predict')
```

```
       fit      lwr     upr
1 7.875942 4.263986 11.4879
```

```r
## ANOVA

regression_model_reduced=lm(  Wins ~ 1  ,data= df )
summary(regression_model_reduced)
```

```
Call:
lm(formula = Wins ~ 1, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9643 -2.9643 -0.4643  3.0357  6.0357 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.9643     0.6576   10.59 4.09e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.48 on 27 degrees of freedom
```

```r
anova(regression_model_reduced,regression_model)
```

```
Analysis of Variance Table

Model 1: Wins ~ 1
Model 2: Wins ~ PassY + PerR + ORY
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1     27 326.96                                  
2     24  69.87  3    257.09 29.437 3.273e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# subset test

regression_model_reduced=lm(  Wins ~ PassY ,data= df )
anova(regression_model_reduced,regression_model)
```

```
Analysis of Variance Table

Model 1: Wins ~ PassY
Model 2: Wins ~ PassY + PerR + ORY
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     26 250.77
2     24  69.87  2     180.9 31.069 2.189e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# summary(df)

summary(regression_model)
```

```
Call:
lm(formula = Wins ~ PassY + PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0370 -0.7129 -0.2043  1.1101  3.7049

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.808372   7.900859  -0.229 0.820899
PassY        0.003598   0.000695   5.177 2.66e-05 ***
PerR         0.193960   0.088233   2.198 0.037815 *
ORY         -0.004816   0.001277  -3.771 0.000938 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.706 on 24 degrees of freedom
Multiple R-squared:  0.7863,    Adjusted R-squared:  0.7596
F-statistic: 29.44 on 3 and 24 DF,  p-value: 3.273e-08
```

```r
# anova(regression_model)
summ=summary(regression_model)
summ$r.squared
```

```
[1] 0.7863069
```

```r
summ$adj.r.squared
```

```
[1] 0.7595953
```

```r
regression_model2=lm(Wins~PassY+ORY,data=df)

SSER=sum(regression_model2$residuals*regression_model2$residuals); SSER
```

```
[1] 83.9382
```

```r
dfer=regression_model2$df.residual; dfer
```

```
[1] 25
```

```r
SSEC=sum(regression_model$residuals*regression_model$residuals); SSEC
```

```
[1] 69.87
```

```r
dfeC=regression_model$df.residual; dfeC
```

```
[1] 24
```

```r
SSdrop=SSER-SSEC; SSdrop
```

```
[1] 14.06819
```

```r
dfdrop=dfer-dfeC

MSdrop=SSdrop/dfdrop; MSdrop
```

```
[1] 14.06819
```

```
R_prp=SSdrop/SSER; R_prp
```

```
[1] 0.1676018
```

```
MSdrop
```

```
[1] 14.06819
```

```
1-pf(MSdrop,dfdrop,dfeC)
```

```
[1] 0.000986662
```

```
cor(regression_model$fitted.values , df$Wins)^2
```

```
[1] 0.7863069
```

```
confint(regression_model2)
```

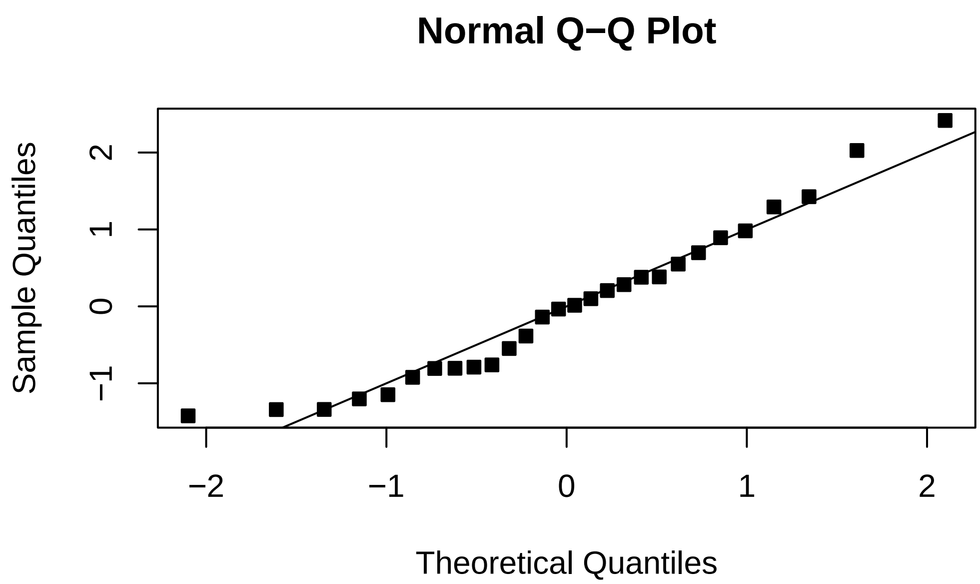
```
                   2.5 %       97.5 %
(Intercept)  9.321778092 20.103571885
PassY        0.001654121  0.004568143
ORY         -0.008797465 -0.004819085
```

```
new_data=df[1,c(3,8,9)]
new_data[1,]=c(2300 , 56 , 2100)
predict(regression_model2,new_data,interval = 'confidence')
```

```
     fit      lwr      upr
1 7.5709 6.814662 8.327138
```

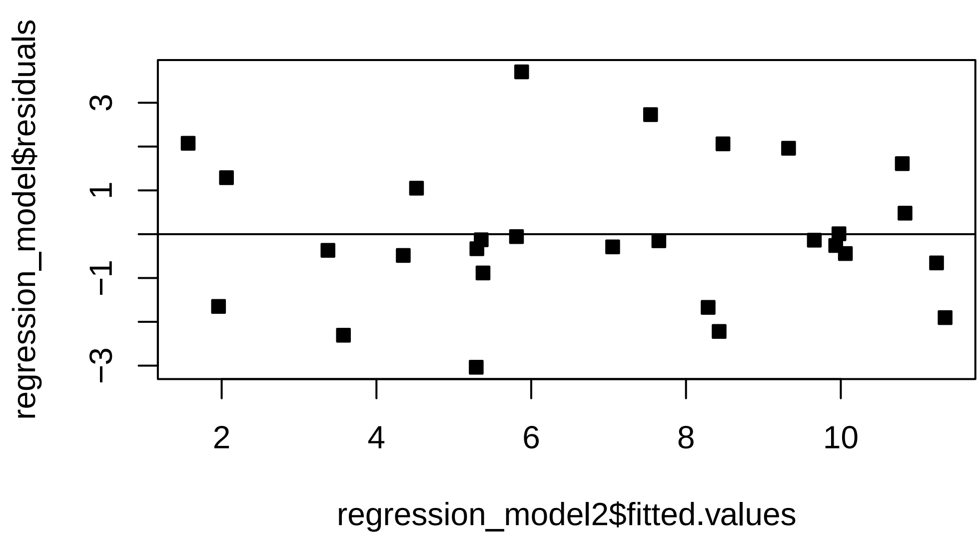
```
#########  check the fit #######
MSE=summ$sigma^2
qqnorm(regression_model2$residuals/summ$sigma,pch=22,bg=1)
```

```r
abline(0,1)
```




```r
hist(regression_model2$residuals,breaks=6)
```

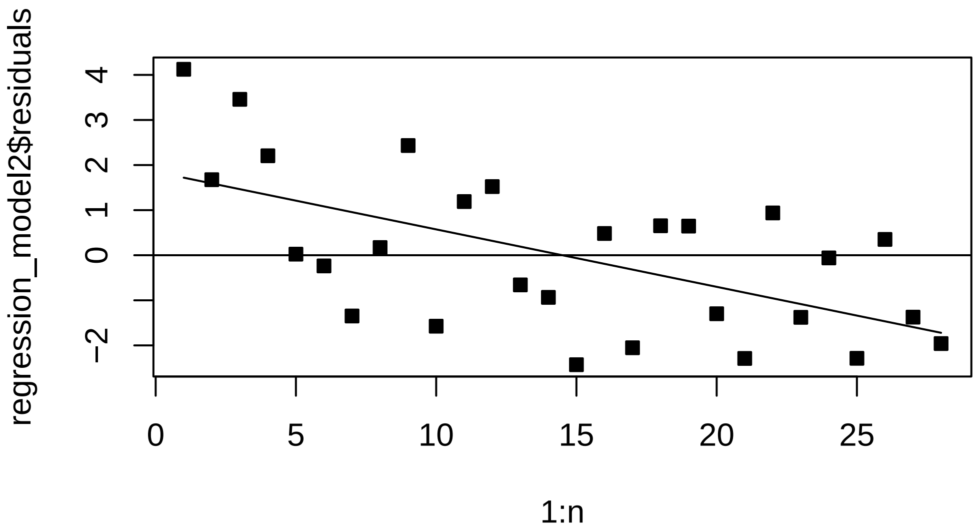

```
plot(regression_model2$fitted.values,regression_model$residuals,pch=22,bg=1)
abline(h=0)
```



```r
n=nrow(df)
plot(1:n,regression_model2$residuals,pch=22,bg=1)
abline(h=0)

time=(1:n)
res=lm(regression_model2$residuals~time)
summary(res)
```

```

Call:
lm(formula = regression_model2$residuals ~ time)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.36425 -1.04520 -0.07845  1.16457  2.40353 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.8479     0.5610   3.294 0.002852 ** 
time         -0.1274     0.0338  -3.771 0.000848 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.445 on 26 degrees of freedom
Multiple R-squared:  0.3535,	Adjusted R-squared:  0.3286 
F-statistic: 14.22 on 1 and 26 DF,  p-value: 0.0008481
```

```r
lines(time,res$fitted.values)
```




```r
regression_model3=lm(Wins~PerR+ORY,data=df)
summ3=summary(regression_model3)
summ3
```

```

Call:
lm(formula = Wins ~ PerR + ORY, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7985 -1.5166 -0.5792  1.9927  4.5248 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 17.944319   9.862484   1.819  0.08084 . 
PerR         0.048371   0.119219   0.406  0.68839   
ORY         -0.006537   0.001758  -3.719  0.00102 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.432 on 25 degrees of freedom
Multiple R-squared:  0.5477,	Adjusted R-squared:  0.5115 
```

```
F-statistic: 15.13 on 2 and 25 DF,  p-value: 4.935e-05
```

```r
summ3$r.squared
```

```
[1] 0.5476628
```

```r
summ3$adj.r.squared
```

```
[1] 0.5114759
```

```r
confint(regression_model)
```

```
                   2.5 %       97.5 %
(Intercept) -18.114944410 14.498200293
PassY         0.002163664  0.005032477
PerR          0.011855322  0.376065098
ORY          -0.007451027 -0.002179961
```

```r
confint(regression_model3)
```

```
                 2.5 %       97.5 %
(Intercept) -2.36784828 38.256485319
PerR        -0.19716429  0.293906022
ORY         -0.01015637 -0.002916818
```

```r
new_data3=df[1,c(8,9)]
new_data3[1,]=c(56 , 2100)
predict(regression_model3,new_data3,interval = 'confidence')
```

```
       fit      lwr      upr
1 6.926243 5.828643 8.023842
```

```r
predict(regression_model2,new_data,interval = 'confidence')
```

```
      fit      lwr      upr
1 7.5709 6.814662 8.327138
```

Be careful about extrapolating beyond the region containing the original observations. It is very possible that a model that fits well in the region of the original data will perform poorly outside that region. It is easy to inadvertently extrapolate, since the levels of the regressors jointly define a region containing the data which is impossible to visualize in its entirety beyond 2 dimensions. Ideally, we want to make inferences which lie inside the convex hull of the regressors.

We can use the diagonal of the hat matrix $H = X(X^\top X)^{-1}X^\top$. In general, the point that has the largest value of h_{ii}, say h_{max}, will lie on the boundary of the convex hull in a region of the x-space where the density of the observations is relatively low. Points that lie in the set $\{x^\top(X^\top X)^{-1}x \leq h_{max}\}$ enclose the convex hull. Thus, for a value we are interested in predicting, say y, we can check if we are extrapolating with $y^\top(X^\top X)^{-1}y \leq h_{max}$.

A serious problem that may dramatically impact the usefulness of a regression model is multicollinearity, or near - linear dependence among the regression variables. Multicollinearity implies near - linear dependence among the regressors. The regressors are the columns of the X matrix, so clearly an exact linear dependence would result in a singular $X^\top X$. This will impact our ability to estimate β.

We can check for this dependence with the **variance inflation factor** (VIF). The variance inflation factor can be written as $(1 - R_j^2)^{-1}$, where R_j^2 is the coefficient of determination obtained from regressing X_j on the other regressor variables. If VIF is large, say > 3, then you will likely need to make some changes to your regression model.

Sometimes, you may observe that regression coefficients have the a sign that is unexpected, or contradicts nature. This is likely due to one of the following:

- The range of some of the regressors is too small – if the range of some of the regressors is too small, then the variance of $\hat{\beta}$ is high.
- Important regressors have not been included in the model.
- Multicollinearity is present.
- Computational errors have been made.

We close this Chapter with the following statement. Recall the modelling overview from Chapter 1:

- Posit the model: What is the linear regression model – what are all the assumptions of the linear regression model?
- Estimation: How can we estimate parameters of the linear regression model?
- Inference: How can we compute confidence intervals and run hypothesis tests associated with the linear regression model?

- Fit: Does our fitted line match up with the data? What about the normality assumption? Do the errors appear normal? Do the errors seem independent? Is the variance constant? How much variability is explained by our model?
- Prediction: How can we predict a new Y? What is the error of this prediction

If you have learned the concepts of this chapter, you should be able to complete all of these steps! In the following chapters, we will discuss different problems that can arise in regression modelling and how to remedy them.

3.6.2 Homework questions

Exercise 3.23. Show $\text{Var}\left[\hat{Y}|X\right] = \sigma^2 H$.

Exercise 3.24. Check for multicolinearity in our past examples.

Exercise 3.25. Complete the problem sets from Chapter's 2, 3 and 4!

4 Residual analysis

Recall that we want to study the normal MLR:

$$Y|X = X\beta + \epsilon,$$

where

- $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.
- $\beta \in \mathbb{R}^{p \times 1}$ is the unknown, population coefficient vector
- $X \in \mathbb{R}^{n \times p}$ is a covariate matrix

We assume that: - The relationship is linear $Y|X = X\beta + \epsilon$, - $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. - $\epsilon_i \perp \epsilon_j$ for $i \neq j,\ i, j \in [n]$.

We have seen some methods for checking if these are appropriate, we will dive deeper now.

Recall that the residuals are defined as:

$$\hat{\epsilon} = Y - \hat{Y} = Y - X\hat{\beta}.$$

Given that a residual may be viewed as the deviation between the data and the fit, it is also a measure of the variability in the response variable not explained by the regression model. It is also convenient to think of the residuals as the realized or observed values of the model errors. Thus, it's reasonable to conclude that departures from the assumptions on the errors should show up in the residuals. Analysis of the residuals is an effective way to discover several types of model inadequacies.

4.1 Properties of residuals

The following are some properties of the residual vector. First, the sample mean of the residuals is zero: $\sum_{i=1}^{n} \hat{\epsilon}_i / n = \hat{\epsilon} 1_n \cdot 1/n = 0$. We also have that $\mathrm{E}\left[\hat{\epsilon}\right] = 0$. Next, the sample variance of the residual vector is approximately the MSE: $\frac{1}{n-1} \sum_{i=1}^{n} \hat{\epsilon}_i^2 = \frac{n-p}{n-1} MSE$. Lastly, unlike the random error ϵ_i, the residuals **are not** independent. Sometimes we say that they are "approximately independent" if $p << n$, which we will touch on later.

4.2 Types of residuals

We will refer to $\hat{\epsilon}_i$ as simply the residuals, or ordinary residuals when we need to be extra clear.

The standardized residual is given by

$$d_i = \hat{\epsilon}_i / \sqrt{MSE}.$$

This is an approximate Z-score for the residuals, since the residuals have 0 mean, the MSE is approximately the variance of the random error and the residuals approximate the random error. We say that large d_i (> 3) indicates an outlier, though, we may want to use a more robust measure of the variance. We will generally prefer to use a different type of residual, which we now present.

We now introduce the hat matrix: $H = X(X^\top X)^{-1}X^\top$. Note that H is symmetric and idempotent. The hat matrix appears often in regression analysis, and you should remember this quantity. It is called the hat matrix because $\hat{Y} = HY$.

Note that the eigenvalues of H, and any idempotent matrix A are either 0 or 1:

$$\lambda x = Ax = A^2 x = A\lambda x = \lambda^2 x,$$

which implies that $\lambda \in \{0, 1\}$.

Exercise 4.1. Verify that H is symmetric and idempotent and that $\hat{Y} = HY$, where one recalls that a matrix A is idempotent if $AA = A$.

Now, note that:

$$\hat{\epsilon} = (I - H)Y = (I - H)\epsilon.$$

Exercise 4.2. Verify that $\hat{\epsilon} = (I - H)Y = (I - H)\epsilon.$.

Using this identity, we have that $\text{Var}\left[\hat{\epsilon}\right] = (I - H)\epsilon(I - H)^\top = (I - H)\sigma^2$.

The fact that H is symmetric and idempotent implies that its diagonal elements are between 0 and 1. It follows that the elements on the diagonal of $(I - H)$ are also between 0 and 1. Therefore, the MSE overestimates the variance of the residuals: the variance of residual i is given by $(1 - h_{ii})\sigma^2 < \sigma^2 \approx MSE$. Here, h_{ii} denotes the ith diagonal element of the matrix H.

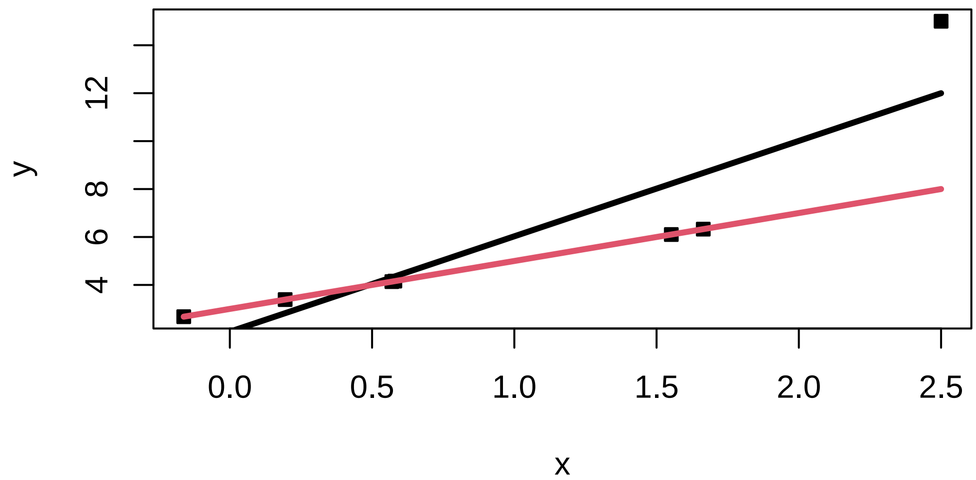
It can be shown that h_{ii} is a measure of how outlying x_i is, relative to the other observed covariate vectors. Therefore, $\text{Var}\left[\hat{\epsilon}_i\right]$ depends on how outlying x_i is. Covariate vectors that are central, relative to the other observed covariate vectors, have larger variance than residuals at more remote locations. What do you think about this?

Now, intuitively, some remote points in our data may differ from the population and severely impact our model. However, the variance of the ordinary residuals is lower at these points. Therefore, these impacts may be hard to detect from solely the ordinary residuals because the ordinary residuals of remote points will usually be smaller.

Therefore, we will call points that are outlying in the x-space **leverage points**. We will refer to **influence points** are not only remote in terms of the specific values for the regressors, but the observed response is not consistent with the values that would be predicted based on only the other data points.

In the example below, observe that the right-most point is both a leverage and influential point.

```
##########
# Simulate some data
set.seed(330)
x=c(rnorm(6),2.5)
y=x*2+3
y[7]=y[7]+7

# Plot the data and fitted lines
plot(x,y,pch=22,bg=1)
a=lm(y~x)
curve(a$coefficients[1]+x*a$coefficients[2],add=T,lwd=3)
curve(x*2+3,add=T,col=2,lwd=3)
```




```
a$coefficients
```

```
(Intercept)           x 
   2.048937    3.979977 
```

We now demonstrate mathematically why leverage points can be dangerous. Let $\hat{y}_n^*$ be the estimate of Y_n based on the other data and let $\delta_n = y_n - \hat{y}_n^*$. Note that one can show that $\hat{y}_n = \hat{Y}_n^* + h_{nn}\delta_n$. Next, we know that if x_n is outlying, i.e., $||x_n||$ is large, then $h_{nn} \approx 1$. This implies that $\hat{y}_n \approx y_n$, which means that the regression line is dragged to pass through (x_n, y_n).

To detect these types of outlying points, it makes sense to then define the **studentized residuals**:

$$r_i = \frac{\hat{\epsilon}_i}{\sqrt{MSE(1-h_{ii})}}.$$

The studentized residuals in the simple linear regression model reduce to

$$r_i = \frac{\hat{\epsilon}_i}{\sqrt{MSE}\left[1 - \left(\frac{1}{n} + \frac{(X_i - \bar{X})^2}{\sum (X - \bar{X})^2}\right)\right]}.$$

Observe that as X_i grows large, we have that $\frac{(X_i - \bar{X})^2}{\sum(X - \bar{X})^2} \to 1$, which implies that $\left[1 - \left(\frac{1}{n} + \frac{(X_i - \bar{X})^2}{\sum(X - \bar{X})^2}\right)\right] \to 0$ and $r_i \to \infty$. On the other hand, as $\hat{\epsilon}_i$ grows large, we have that r_i grows large. Therefore, the studentized residual will be large for observations with large ordinary residuals, and for leverage observations.

Earlier, we presented δ_i, the difference between the response of the ith observation and the predicted response based on the observations with the ith points removed. These are known as the **PRESS residuals**. This seems hard computationally, but one can show that

$$\delta_i = \frac{\hat{\epsilon}_i}{1 - h_{ii}}.$$

Note that when h_{ii} is large, this indicates a highly influential point. Observe that a large PRESS residual δ_i, but small ordinary residual $\hat{\epsilon}_i$, indicates that the model fit without (X_i, Y_i) predicts Y_i poorly.

Exercise 4.3. Show that standardizing the PRESS residual, that is, dividing the PRESS residual by its standard deviation, results in $\hat{\epsilon}_i / \sqrt{\sigma^2(1 - h_{ii})}$. Compare this to the studentized residual.

Lastly, if we believe that (X_i, Y_i) is outlying, then we can also leave (X_i, Y_i) out in the MSE calculation. This results in the **R-studentized residuals**:

$$\tilde{r}_i = \frac{\hat{\epsilon}_i}{\sqrt{\widetilde{MSE}_i(1 - h_{ii})}},$$

where $\widetilde{MSE}_i$ is the mean squared error computed from the regression model with (X_i, Y_i) excluded:

$$\widetilde{MSE}_i = \frac{(n - p + 1)MSE - \hat{\epsilon}_i^2 / (1 - h_{ii})}{n - p}.$$

4.3 Revisiting checking model assumptions

Recall from Checking model assumptions that we plot the residuals to check various assumptions. In this case, we can now use our upgraded residuals to make these plots. In general, any of the residuals that incorporate the values h_{ii} are acceptable. We will generally use the studentized residuals.

Recall that we may want to plot:

- QQplot of the studentized residuals
- Histogram of the studentized residuals

- Plot of studentized residuals against the fitted values
- Studentized residuals against the covariates
- Studentized residuals against covariates that are not currently in the model
- Studentized residuals against time in some contexts

4.3.1 Example

Example 4.1. Here, this data contains delivery times, the number of products in the delivery and the distance of the delivery. Perform a residual analysis on the model which regresses delivery times against the number of products in the delivery and the distance of the delivery. Compute all the different types of residuals.

```
########## Delivery Time


# Load and inspect the data
data(delivery, package="robustbase")
df=delivery
n=nrow(df)
head(df)
```

```
  n.prod distance delTime
1      7      560   16.68
2      3      220   11.50
3      3      340   12.03
4      4       80   14.88
5      6      150   13.75
6      7      330   18.11
```

```
# Fit the model
model=lm(delTime~.,data=df)
s=summary(model); s
```

```

Call:
lm(formula = delTime ~ ., data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.7880 -0.6629  0.4364  1.1566  7.4197 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.341231   1.096730   2.135 0.044170 *
n.prod      1.615907   0.170735   9.464 3.25e-09 ***
distance    0.014385   0.003613   3.981 0.000631 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.259 on 22 degrees of freedom
Multiple R-squared:  0.9596,    Adjusted R-squared:  0.9559
F-statistic: 261.2 on 2 and 22 DF,  p-value: 4.687e-16
```

```r
# X matrix
X=model.matrix(model)

# Hat matrix
hat=X%*%solve(t(X)%*%X)%*%t(X)

# Compute h_ii
hii=diag(hat)
hii
```

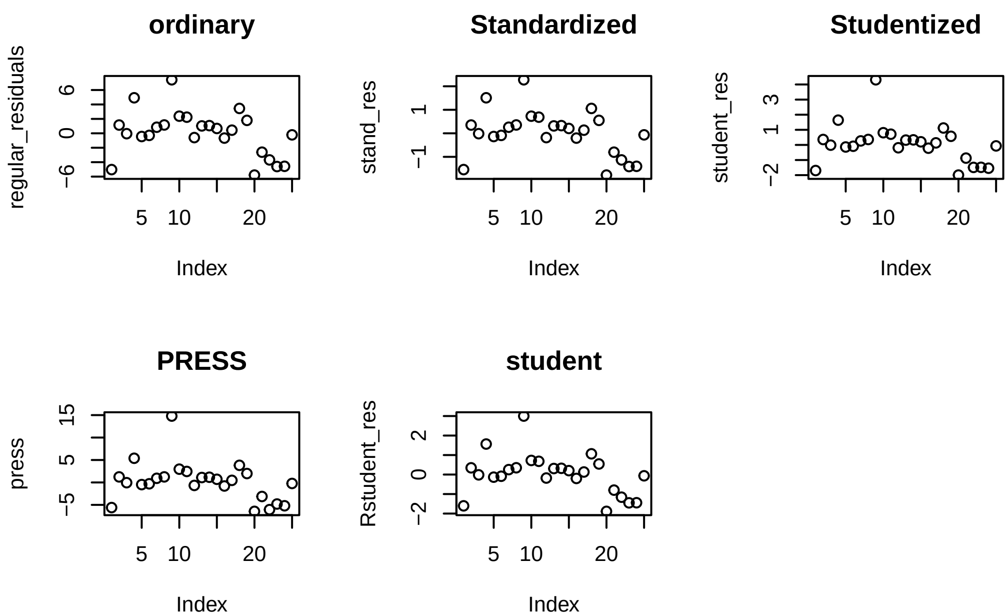
```
         1          2          3          4          5          6          7
0.10180178 0.07070164 0.09873476 0.08537479 0.07501050 0.04286693 0.08179867
         8          9         10         11         12         13         14
0.06372559 0.49829216 0.19629595 0.08613260 0.11365570 0.06112463 0.07824332
        15         16         17         18         19         20         21
0.04111077 0.16594043 0.05943202 0.09626046 0.09644857 0.10168486 0.16527689
        22         23         24         25
0.39157522 0.04126005 0.12060826 0.06664345
```

```r
max(hii)
```

```
[1] 0.4982922
```

```r
# Notice 9 is large

###### ordinary residuals
```

```r
regular_residuals=model$residuals
# or


# standardized residuals
stand_res=model$residuals/s$sigma

# studentized residuals
student_res=rstudent(model)


#PRESS residuals
press=model$residuals/(1-hii)

# Get the MSE_is
MSE_i=((n-2)*(s$sigma)^2-regular_residuals^2/(1-hii))/(n-3)

#r studentized residuals
Rstudent_res=model$residuals/sqrt(MSE_i)

# Plot them all and compare
par(mfrow=c(2,3))
plot(regular_residuals,main="ordinary")
plot(stand_res,main="Standardized")
plot(student_res,main="Studentized")
plot(press,main="PRESS")
plot(Rstudent_res,main="student")

# Notice 9 is much more outlying in the last 3 graphs.

# Reset plotting
par(mfrow=c(1,1))
```




```
# 9 is largest
which.max(student_res)
```

```
9
9
```

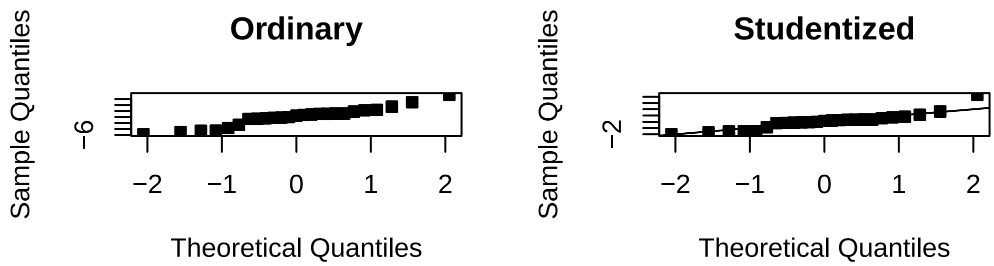
```
# Notice the standardized is half as large as the studentized.
student_res[9]
```

```
       9
4.31078
```

```
stand_res[9]
```

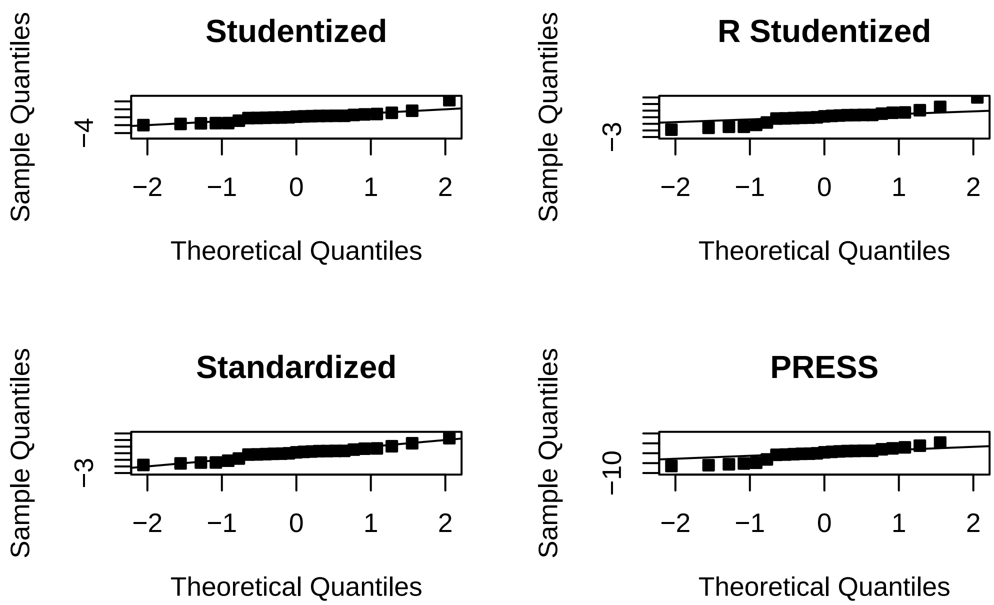
```
        9
2.276351
```

```r
par(mfrow=c(2,2))

# Notice the difference !!!
qqnorm(regular_residuals,pch=22,bg=1,main="Ordinary")

qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)

# Compare all

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

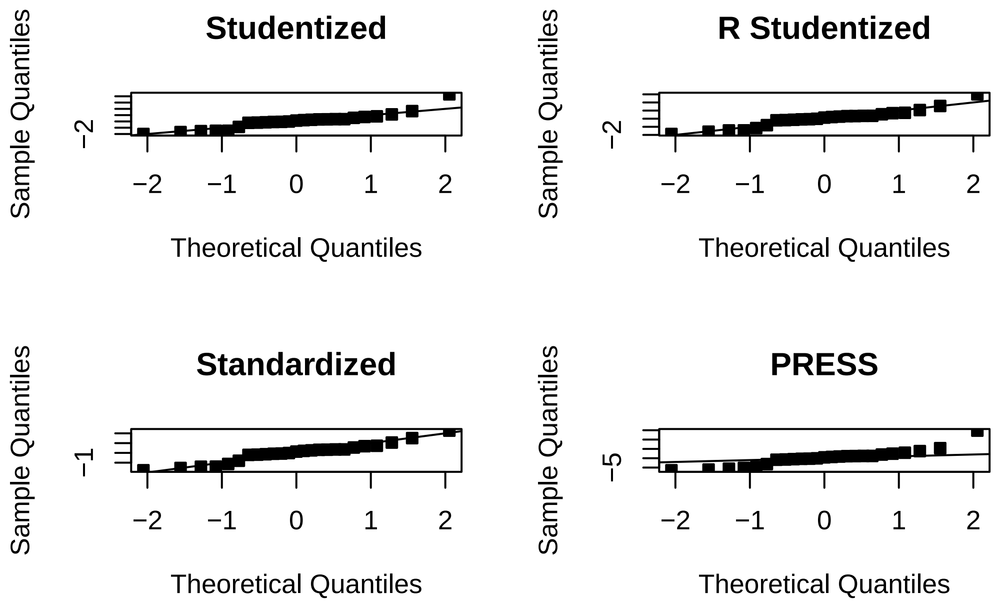



```r
qqnorm(student_res,pch=22,bg=1,ylim=c(-5,5),main="Studentized")
abline(0,1)
# hist(student_res)

qqnorm(Rstudent_res,pch=22,bg=1,ylim=c(-3,3),main="R Studentized")
qqline(Rstudent_res,pch=22,bg=1,ylim=c(-10,10))
# abline(0,1)

qqnorm(stand_res,pch=22,bg=1,ylim=c(-3,3),main="Standardized")
```

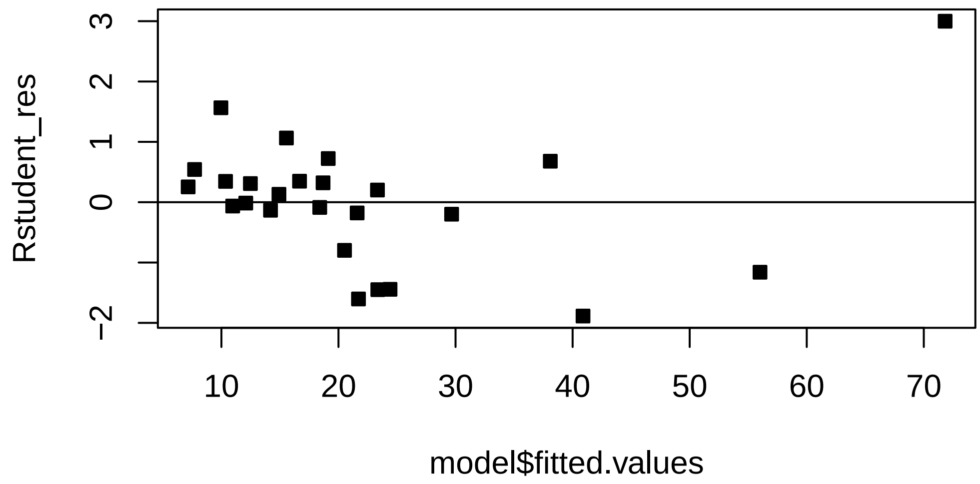
```
abline(0,1)

qqnorm(press,pch=22,bg=1,ylim=c(-10,10),main="PRESS")
qqline(press,pch=22,bg=1,ylim=c(-10,10))
#careful of the scale!

par(mfrow=c(3,2))
qqline(model$residuals,pch=22,bg=1,main="Ordinary")
```



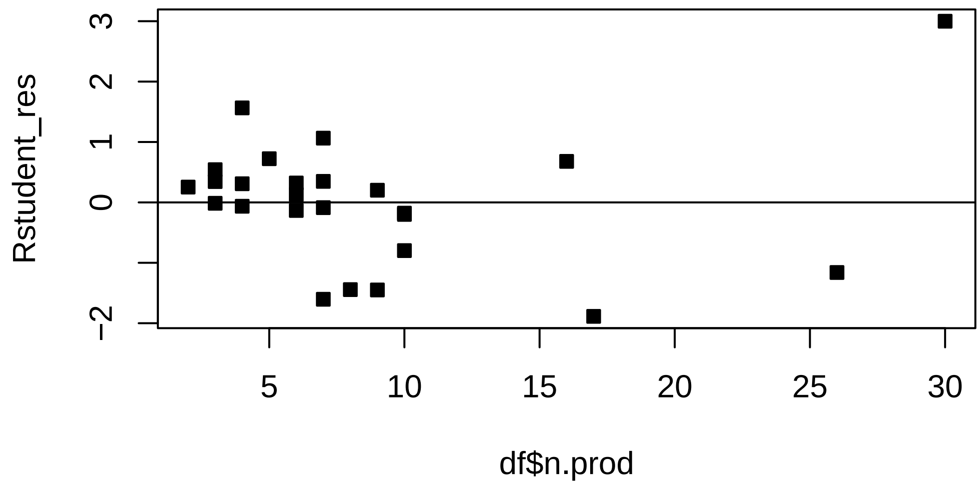

```
par(mfrow=c(2,2))
qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)
qqnorm(Rstudent_res,pch=22,bg=1,main="R Studentized")
abline(0,1)
qqnorm(stand_res,pch=22,bg=1,main="Standardized")
abline(0,1)
qqnorm(press,pch=22,bg=1,main="PRESS")
abline(0,1)
```



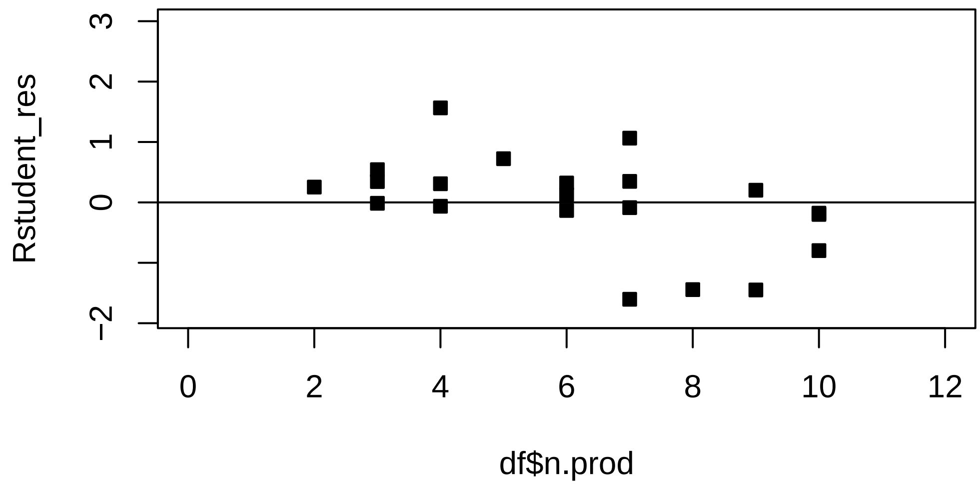

```
# Now we plot the fitted values against the R studentized residuals
par(mfrow=c(1,1),pch=22)
plot(model$fitted.values,Rstudent_res,bg=1)
abline(h=0)
```



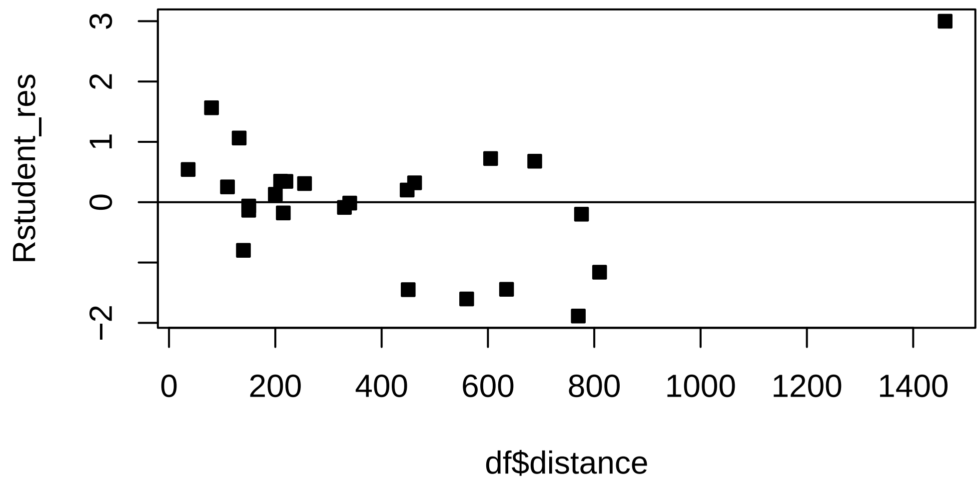

```
# Now we plot the number of products against the R studentized residuals
# There is one moderately large delivery!
plot(df$n.prod,Rstudent_res,bg=1)
abline(h=0)
```



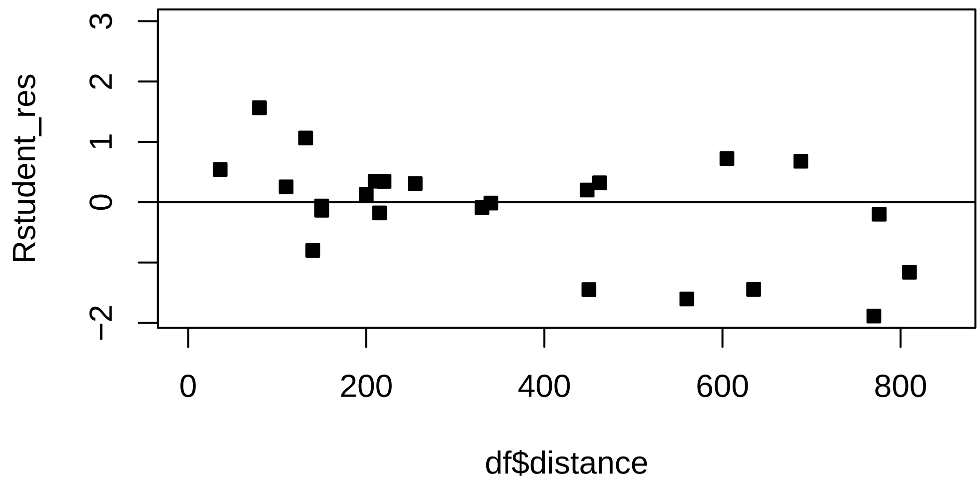

```r
# Care for the scale
plot(df$n.prod,Rstudent_res,bg=1,xlim=c(0,12))
abline(h=0)
```




```r
# There is one very far delivery!
plot(df$distance,Rstudent_res,bg=1)
abline(h=0)
```




```
# Care for the scale
plot(df$distance,Rstudent_res,bg=1,xlim=c(0,850))
abline(h=0)
```




```r
# What happens to the model when we remove this outlying observation (the far distance del
df2=df[-which.max(df$distance),]


# refit the model
model=lm(delTime~.,data=df2)
s=summary(model); s
```

```

Call:
lm(formula = delTime ~ ., data = df2)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0325 -1.2331  0.0199  1.4730  4.8167 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.447238   0.952469   4.669 0.000131 ***
n.prod      1.497691   0.130207  11.502 1.58e-10 ***
distance    0.010324   0.002854   3.618 0.001614 ** 
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.43 on 21 degrees of freedom
Multiple R-squared:  0.9487,    Adjusted R-squared:  0.9438
F-statistic: 194.2 on 2 and 21 DF,  p-value: 2.859e-14
```

```
# X matrix
X=model.matrix(model)

# Hat matrix
hat=X%*%solve(t(X)%*%X)%*%t(X)

# Compute h_ii
hii=diag(hat)
hii
```

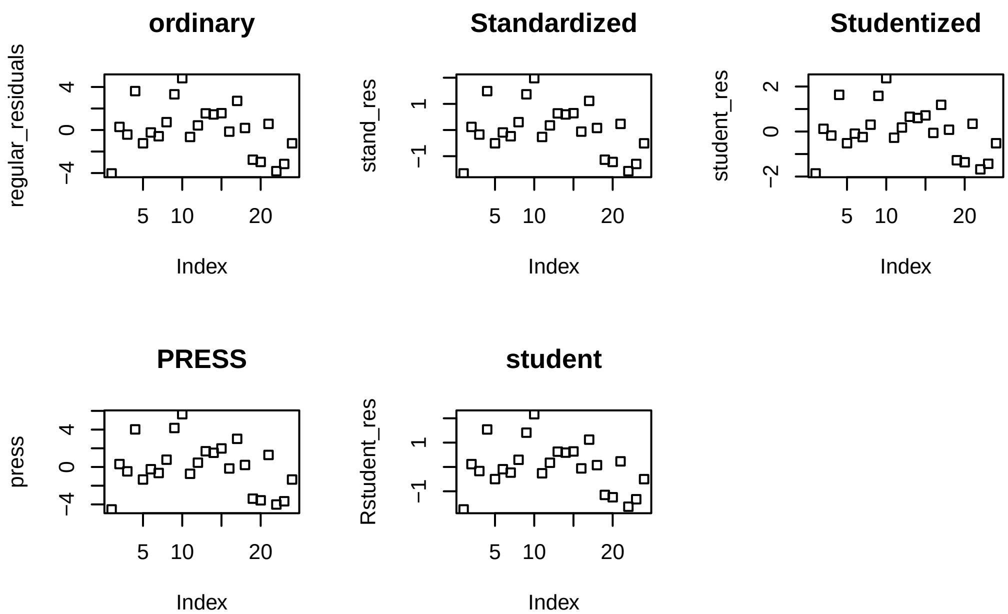
```
         1          2          3          4          5          6          7
0.11083391 0.07741039 0.09998709 0.10097319 0.08066357 0.04290146 0.10024969
         8         10         11         12         13         14         15
0.06537738 0.20438000 0.14675966 0.11367920 0.06437975 0.08033747 0.04661503
        16         17         18         19         20         21         22
0.21115081 0.06254612 0.10128434 0.11992977 0.18537865 0.16642759 0.55671434
        23         24         25
0.04687996 0.13894064 0.07620000
```

```
max(hii)
```

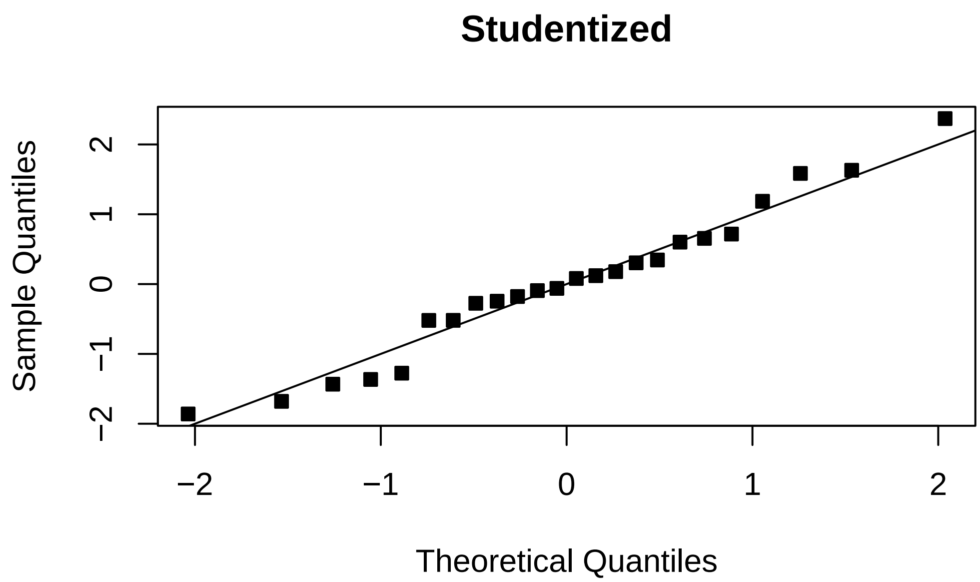
```
[1] 0.5567143
```

```
###### ordinary residuals
regular_residuals=model$residuals
# or


# standardized residuals
stand_res=model$residuals/s$sigma

# studentized residuals
student_res=rstudent(model)
```

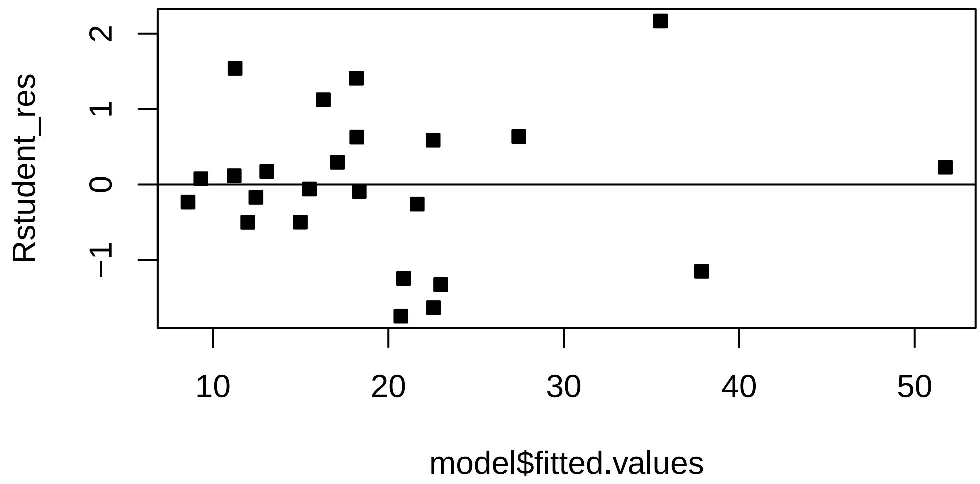
```
#PRESS residuals
press=model$residuals/(1-hii)

# Get the MSE_is
MSE_i=((n-2)*(s$sigma)^2-regular_residuals^2/(1-hii))/(n-3)

#r studentized residuals
Rstudent_res=model$residuals/sqrt(MSE_i)

# Plot them all and compare - much better
par(mfrow=c(2,3))
plot(regular_residuals,main="ordinary")
plot(stand_res,main="Standardized")
plot(student_res,main="Studentized")
plot(press,main="PRESS")
plot(Rstudent_res,main="student")


# Notice that these graphs are fine now...
par(mfrow=c(1,1))
```



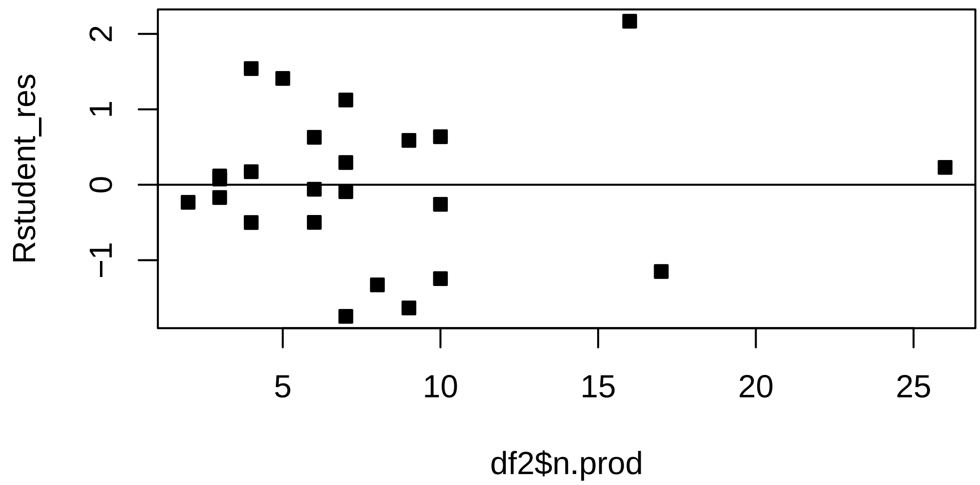
```
qqnorm(student_res,pch=22,bg=1,main="Studentized")
abline(0,1)
```



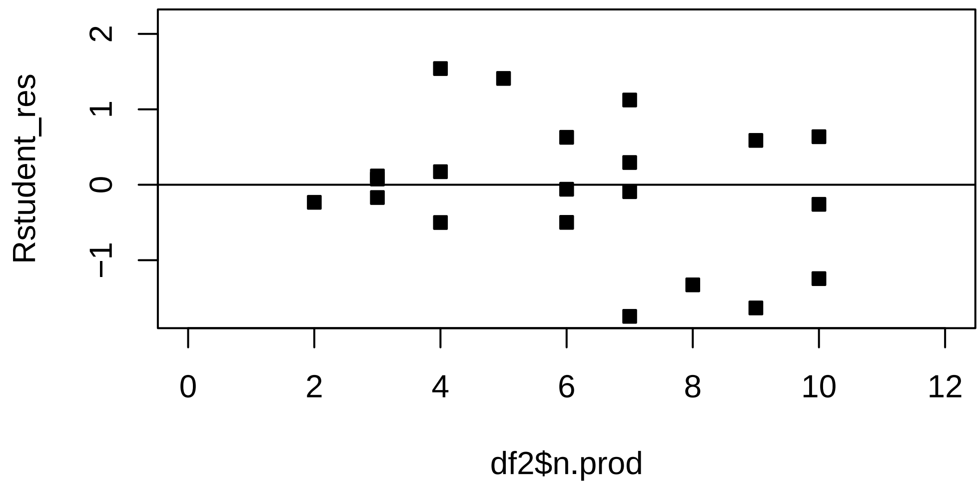

```
# Now we plot the fitted values against the R studentized residuals
par(mfrow=c(1,1),pch=22)
plot(model$fitted.values,Rstudent_res,bg=1)
abline(h=0)
```



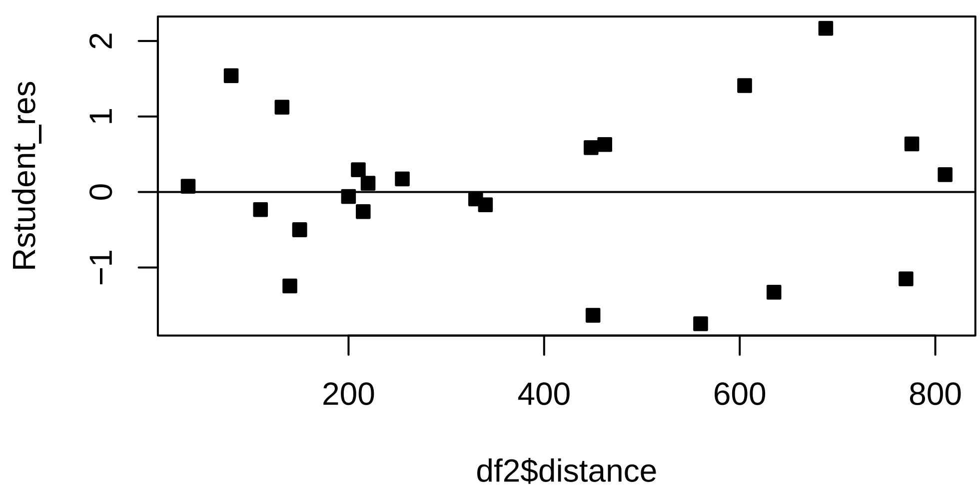

```
# Now we plot the number of products against the R studentized residuals
plot(df2$n.prod,Rstudent_res,bg=1)
abline(h=0)
```



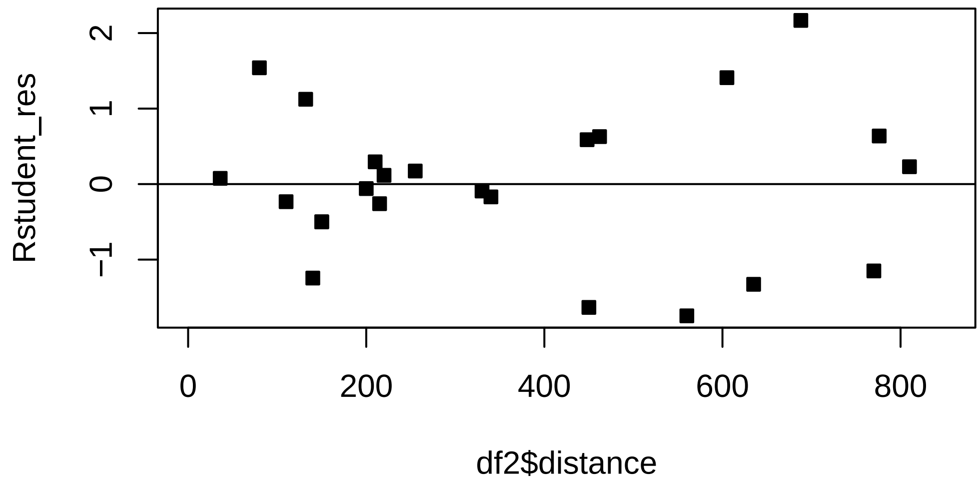

```
# Care for the scale
plot(df2$n.prod,Rstudent_res,bg=1,xlim=c(0,12))
abline(h=0)
```




```
plot(df2$distance,Rstudent_res,bg=1)
abline(h=0)
```



```
# Care for the scale
plot(df2$distance,Rstudent_res,bg=1,xlim=c(0,850))
abline(h=0)
```



Now, we have introduced different types of residuals and the appropriate graphs to examine when checking for violations of the assumptions. When we observe violations of the assumptions - what do we do? That will be the topic of the next section.

Some of these remedies include: - Transformations of the response - Transformations of certain regressors - Robust methods/outlier removal - Inclusion of new regressors

4.4 Homework stop

Do the Chapter 4 questions from the textbook.

Exercise 4.4. In the context of a regression model, do you think a point whose covariates are outlying is more problematic than a point whose response, given the covariates, is outlying?

Exercise 4.5. Make a table describing the differences between each type of residual.

Exercise 4.6. Perform a residual analysis on the marketing data from Example Example 3.7.

Exercise 4.7. Perform a residual analysis on the data from Example Example 3.8.

5 Transformations

5.1 Variance-stabilizing transformations

Recall that we assume that $\forall i \in [n]$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. A common reason for a violation of this assumption is for Y to have a distribution in which the variance is related to its mean. For example, if the response Y is a Poisson random variable, i.e.,

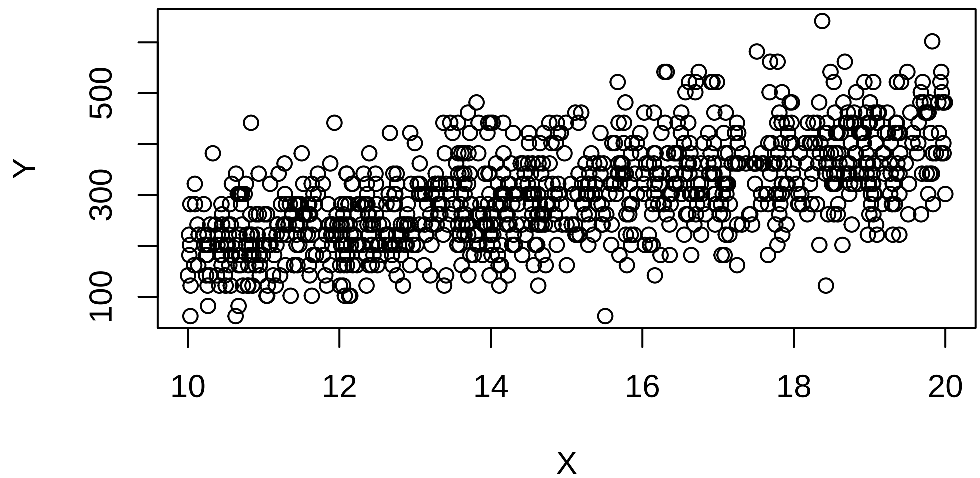
$$Y|X \sim Pois(X\beta),$$

then we have that $\mathrm{E}[Y] = X\beta$, then $\mathrm{E}[Y] = \mathrm{Var}[Y] = X\beta$. In this case, the simple linear regression assumptions are violated. In particular, the variance is not the same for each observation. Here, it happens that taking the response to be roughly $\sqrt{Y}$ fixes the problem. That is, performing the regression analysis with $\sqrt{Y}$ as the response variable instead of Y, ensures that the regression assumptions are (approximately) satisfied. This example gives rise to the idea of **transformations**. If our data do not satisfy the assumptions for the MLR or the normal MLR, we might ask if there is some transformation of either the response, some of the covariates, or both that make the data suitable for a MLR analysis. Note that the assumptions are important. For instance, if the variance is not homogeneous, the OLS estimator will still be unbiased, but they will no longer have BLUE property. That means that some other estimator will work better for such data!

Which transformation should we choose? Sometimes, we can use prior experience or theoretical considerations to guide us in selecting an appropriate transformation. Other times, we must choose it empirically, i.e., based on the data. Often, the square root and the logarithm are popular choices. If you response is between 0 and 1, and the data appear to be "football shaped", then you may like to take the $\arcsin(\sqrt{Y})$.

We now demonstrate what one of these relationships looks like in simple linear regression. We now simulate a dataset where $\sigma^2 \propto \mathrm{E}[Y|X]$, and plot X against Y. We use the Poisson example discussed previously.

```
set.seed(2352)
n=1000
X=runif(n,5,10)*2
Y=20*rpois(n,X)+2
plot(X,Y)
```

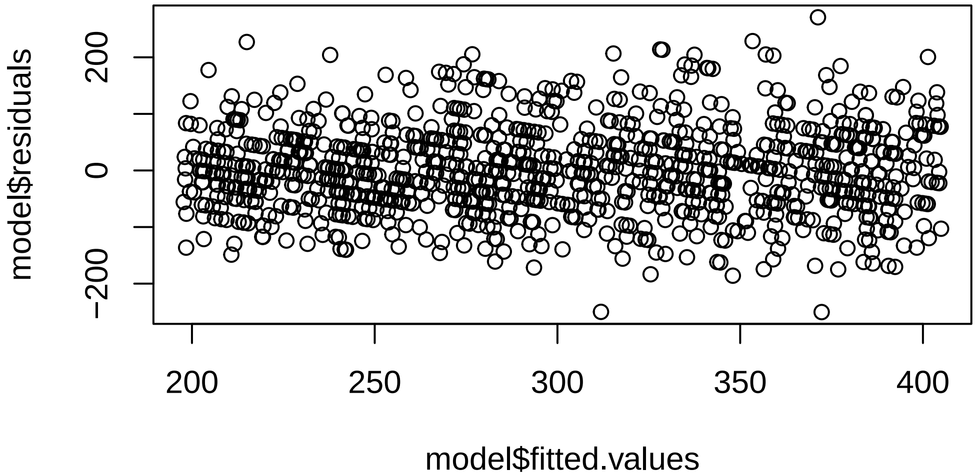



```
# Performing a regression analysis yields:
model=lm(Y~X)
# Notice the intercept is poorly estimated!
summary(model)
```

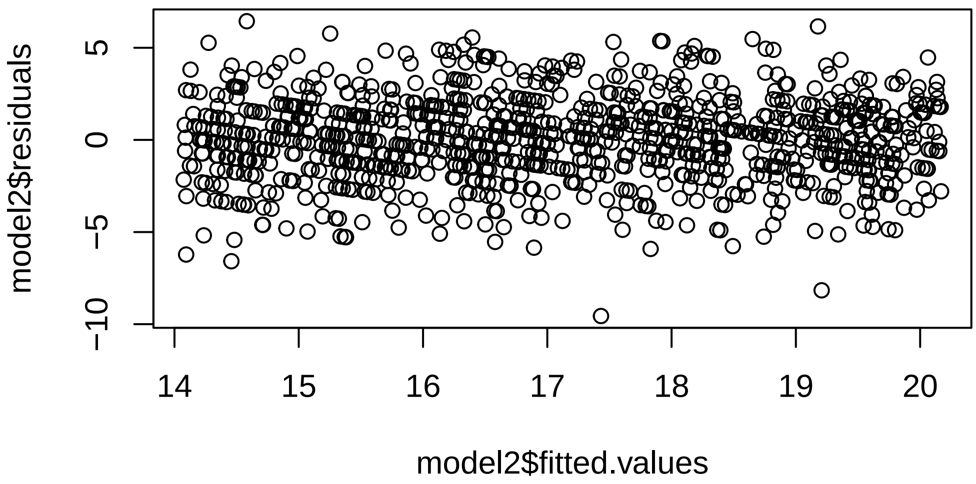
```
Call:
lm(formula = Y ~ X)

Residuals:
     Min       1Q   Median       3Q      Max
-250.289  -53.392   -2.127   48.795  270.716

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -9.5199    13.0735  -0.728    0.467
X            20.7241     0.8599  24.101   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 77.61 on 998 degrees of freedom
Multiple R-squared:  0.3679,    Adjusted R-squared:  0.3673
```

```
F-statistic: 580.9 on 1 and 998 DF,  p-value: < 2.2e-16
```

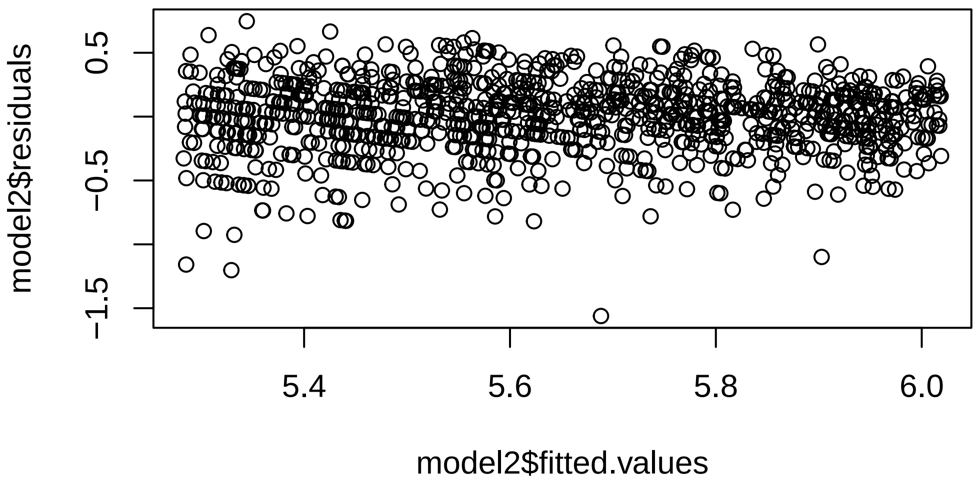
```
# Notice the fan shape in the residuals against the fitted values?
plot(model$fitted.values,model$residuals)
```




```
# Let's perform the transformations

model2=lm(sqrt(Y)~X)

plot(model2$fitted.values,model2$residuals)
```

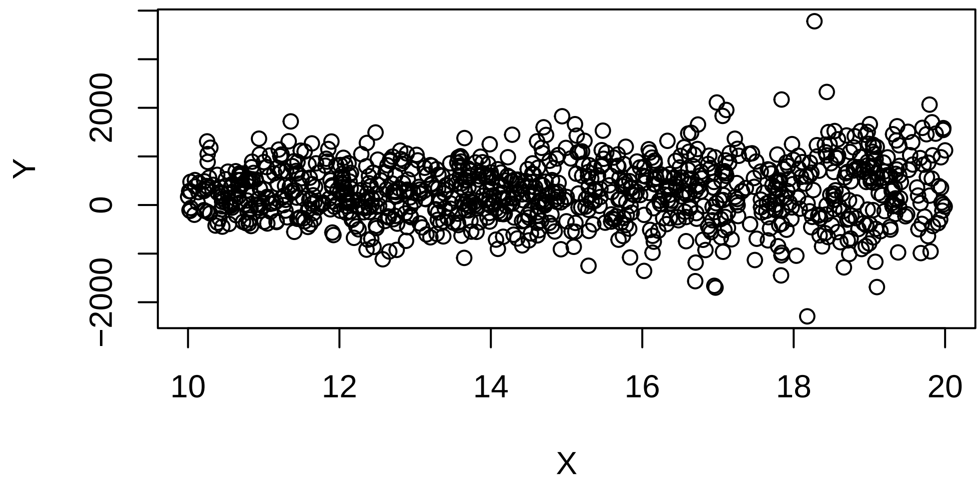



```
model2=lm(log(Y)~X)

plot(model2$fitted.values,model2$residuals)
```



However, these transformations do not always work. Suppose we have that $Y \sim \mathcal{N}(X, 4 * X^2)$. We then have that $\sigma = 2*X = 2*\mathrm{E}[Y|X]$. Notice how the spread of the points is increasing with X? This is a symptom of non-homogeneous variance. However, the proposed transformations do not work.

```
set.seed(2352)
# \sigma^2\propto \E{Y}
Y=20*rnorm(n,X,X*2)+2

plot(X,Y)
```

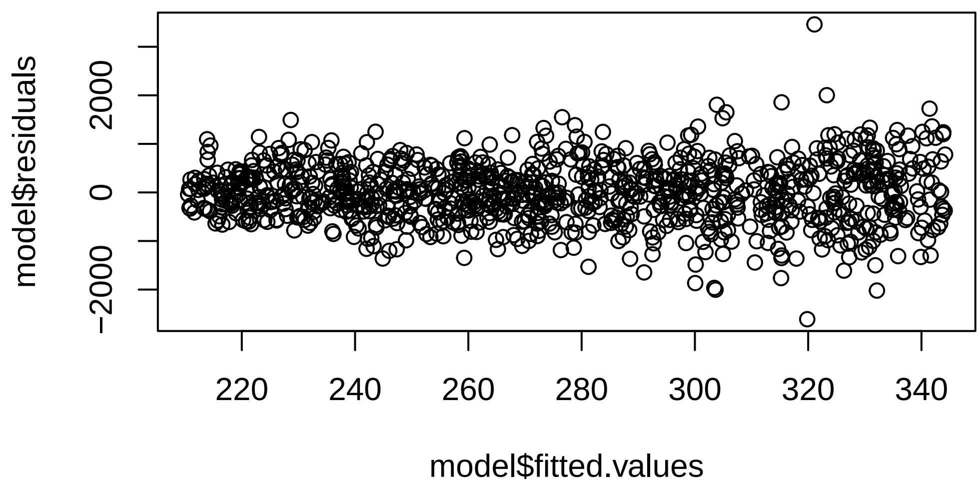



```r
# Performing a regression analysis yields:
model=lm(Y~X)

# notice the intercept is poorly estimated.
summary(model)
```

```
Call:
lm(formula = Y ~ X)

Residuals:
    Min      1Q  Median      3Q     Max 
-2610.8  -375.3    -3.5   384.8  3460.5 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   76.844    103.345   0.744   0.4573  
X             13.367      6.797   1.966   0.0495 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 613.5 on 998 degrees of freedom
```

```
Multiple R-squared:  0.00386,   Adjusted R-squared:  0.002861
F-statistic: 3.867 on 1 and 998 DF,  p-value: 0.04953
```

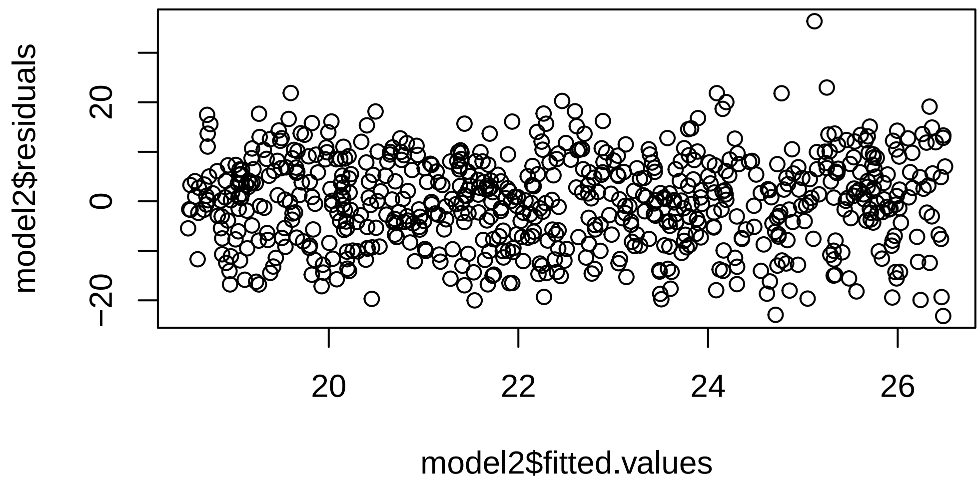
```r
# Notice the fan shape in the residuals against the fitted values?
plot(model$fitted.values,model$residuals)
```



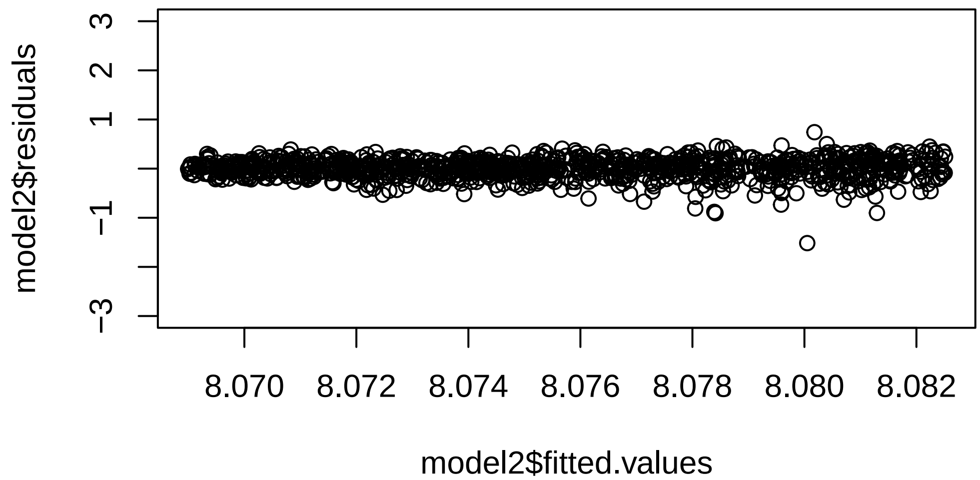

```r
# Let's perform the transformation

# Performing a regression analysis yields:
model2=lm(sqrt(Y)~X)
```

```
Warning in sqrt(Y): NaNs produced
```

```r
# Notice the fan shape in the residuals against the fitted values?
plot(model2$fitted.values,model2$residuals)
```




```
# Performing a regression analysis yields:
model2=lm(log(Y+3000)~X)

# Notice the fan shape in the residuals against the fitted values?
plot(model2$fitted.values,model2$residuals,ylim=c(-3,3))
```



In general, a good transformation to correct violated assumptions can improve estimates and test accuracy.

> **Caution**
>
> It is often necessary to convert any predicted values back to the original units. Applying the inverse transformation to predicted values gives an estimate of the median of the distribution of the (untransformed) response – instead of the mean. This implies that predictions are generally biased. Prediction and confidence intervals do not suffer this illness. They can be converted back to the original units via the inverse transformation and the interpretation will remain the same.

5.1.1 Inverse transformations

Let's expand on this. It is a good time to recall that in general, for a real function f, we have that $\mathrm{E}\left[f(X)\right] \neq f(\mathrm{E}\left[X\right])$. For instance, for many random variables Z, we would have that $\mathrm{E}\left[Z^2\right] \neq \mathrm{E}\left[Z\right]^2$, $\mathrm{E}\left[\log Z\right] \neq \log \mathrm{E}\left[Z\right]$ etc. .

In a transformed regression model, we fit the following model:

$$f(Y) = X\beta + \epsilon.$$

If we are interested in predicting the value of Y given z, then it seems natural to take the predictions for $f(Y)$ given z, which are given by $\beta^\top z$ and apply the inverse transformation f^{-1}. For instance, to predict $Y|Z = z$, we may compute: $f^{-1}(\beta^\top z)$. It turns out, this prediction is biased, and we should use a different method instead.

To see why it's biased, observe that the predictions from the model $f(Y) = X\beta + \epsilon$ for a new set of covariates z are given by $\widehat{f}(Y) = \widehat{\beta}^\top z \approx \mathrm{E}\left[f(Y)|Z = z\right]$. Now, we have that

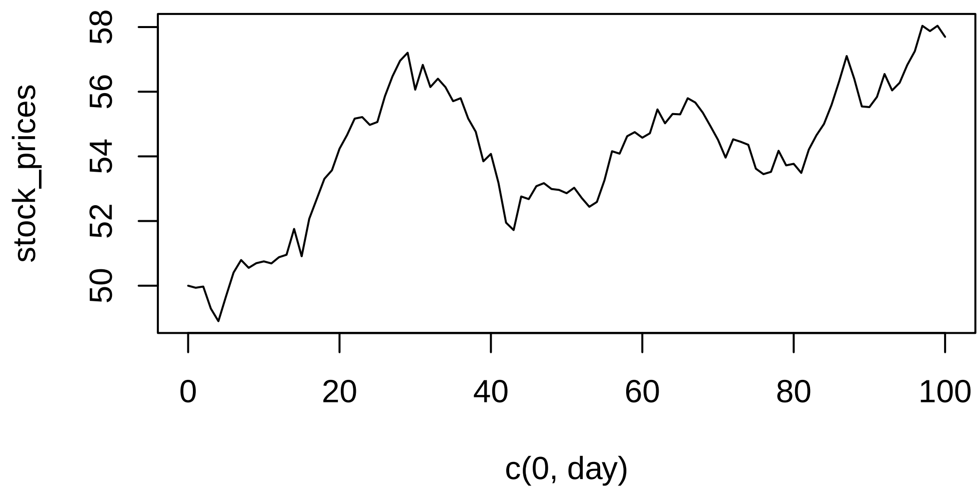
$$f^{-1}(\widehat{\beta}^\top z) \approx f^{-1}(\mathrm{E}\left[f(Y)|Z = z\right]) \neq \mathrm{E}\left[f^{-1}(f(Y))|Z = z\right] = \mathrm{E}\left[\widehat{f}(Y)|Z = z\right].$$

The solution to this problem is to adjust for the bias. For the log transform, we can multiply the resulting inverse transformed predictions by $\exp(\widehat{\sigma}^2/2)$. For the square root transformation, we add $\widehat{\sigma}^2$ to the resulting inverse transformed predictions. See (Miller 1984) for more information.

One can also use confidence and prediction intervals to predict the value of Y given z. Confidence or prediction intervals may be directly converted from one metric to another – such interval estimates are percentiles of a distribution which are unaffected by the transformation. They can be converted back to the original units via the inverse transformation and the interpretation will remain the same. Optimal intervals are intervals with the shortest average interval length for a given confidence level, under a given set of assumptions. However, it may be that the resulting intervals may not be "optimal". One way to get a prediction in the original units, is to apply the inverse transformation to the prediction interval computed from the transformed model and take the midpoint of that interval. This does not always work well - and should be checked against the original data.

Example 5.1. Let's simulate what happens when, given the day $t \in [100]$, we try to estimate the mean stock price P_t for some stock (maybe ?Gamestop?) in a model which regresses the logged rate of return against the day. Note that the logged returns at time t are given by: $L = \log\left(\frac{P_t}{P_{t-1}}\right)$.

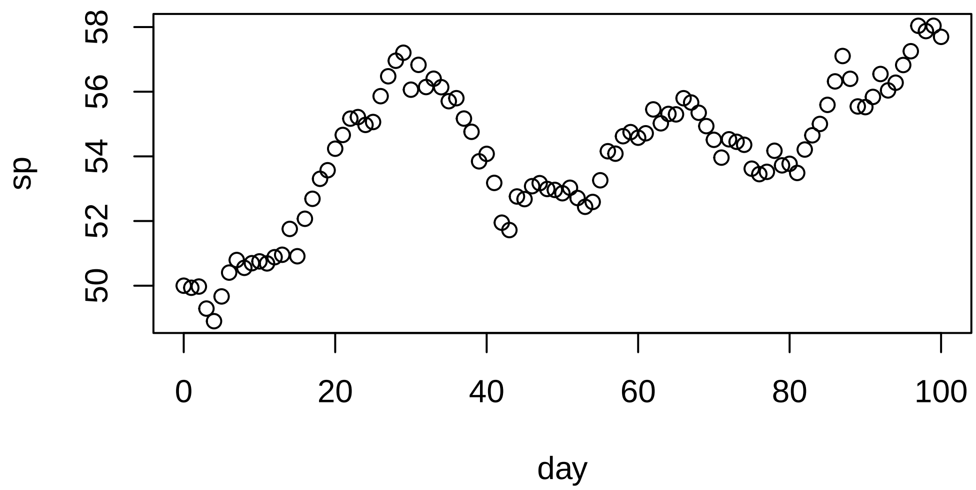
```
set.seed(2352)
# Simulate the stock prices
n=100
day=seq(1:n)
log_return=rnorm(n,0.000001+0.000005*day,0.01)
# log_return=rnorm(n,0.000001+0.000005*day,0.0005)
stock_prices=c(50,50*exp(cumsum(log_return)))
# exp(log_return)[1:10]

plot(c(0,day), stock_prices,type='l')
```




```
df=data.frame(cbind("day"=c(0,day),"sp"=stock_prices))


# Now, suppose this is our starting dataset
head(df)
```

```
  day       sp
1   0 50.00000
2   1 49.93624
3   2 49.97369
4   3 49.29229
5   4 48.90210
6   5 49.66743
```

```
# Notice that the pattern is not great... but we can regress on the transformed response
plot(df)
```




```r
# Fitting the model

# Compute the log returns
df$lr=NA
df$lr[2:(n+1)]=log(df$sp[-1]/df$sp[-(n+1)])

# Sanity Check
# log_return[1:5]
# df$lr[2:6]

model=lm(lr~day,df)
summary(model)
```

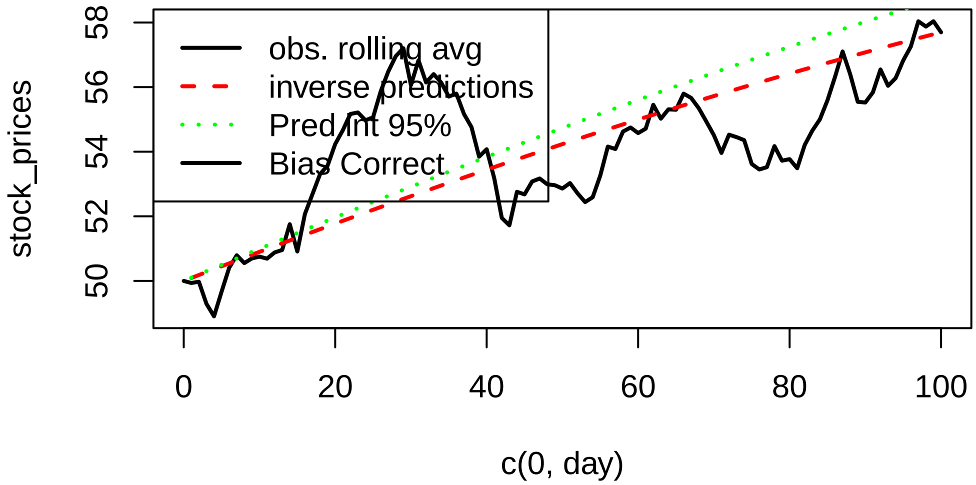
```
Call:
lm(formula = lr ~ day, data = df)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0249054 -0.0064851  0.0000518  0.0075839  0.0208044

Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.824e-03  1.926e-03   0.947    0.346
day         -7.771e-06  3.312e-05  -0.235    0.815

Residual standard error: 0.00956 on 98 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.0005615, Adjusted R-squared:  -0.009637
F-statistic: 0.05506 on 1 and 98 DF,  p-value: 0.815
```

```
plot(c(0,day), stock_prices,type='l',lwd=2)
lines(50*exp(cumsum(fitted.values(model))),col='red',lty=2,lwd=2)

# Intervals for the mean at each time point
intervals=predict(model,interval ='prediction')[,2:3]
```

```
Warning in predict.lm(model, interval = "prediction"): predictions on current data refer to _
```

```
midpoint=ret=rep(0,n)
for(i in 1:n){
  if(i==1){
    midpoint[i]=50*exp(intervals[i,1])/2+50*exp(intervals[i,2])/2
  }
  else
    midpoint[i]=midpoint[i-1]*(exp(intervals[i,1])+exp(intervals[i,2]))/2
}
lines(midpoint,col="green",lty=3,lwd=2)

legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Bias Co
```



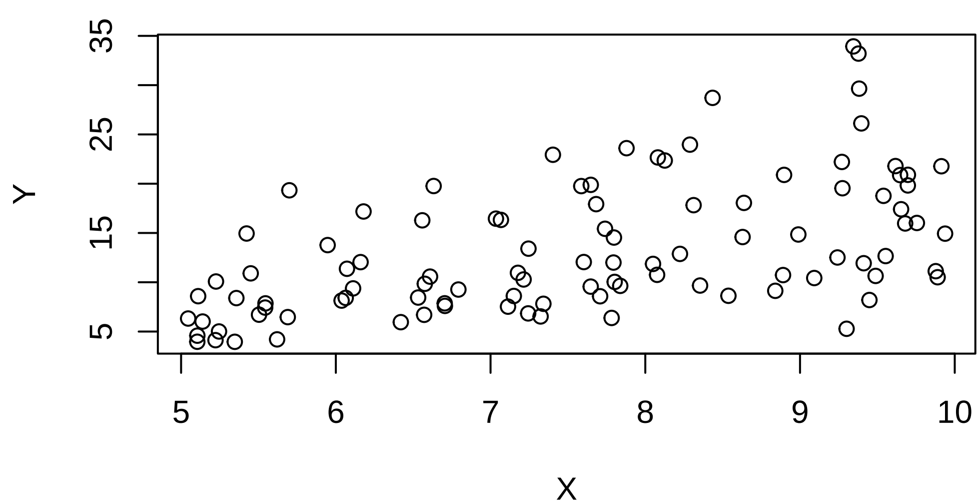
A second example...

```
set.seed(2352)
# Simulate data
n=100
X=runif(n,5,10)
logs=rnorm(n,1+0.2*X,0.5)
Y=exp(logs)
plot(X,Y)


df=data.frame(cbind("X"=X,"Y"=Y))
df=df[order(X),]

# Fitting the model
model=lm(log(Y)~X,data=df)
summary(model)
```

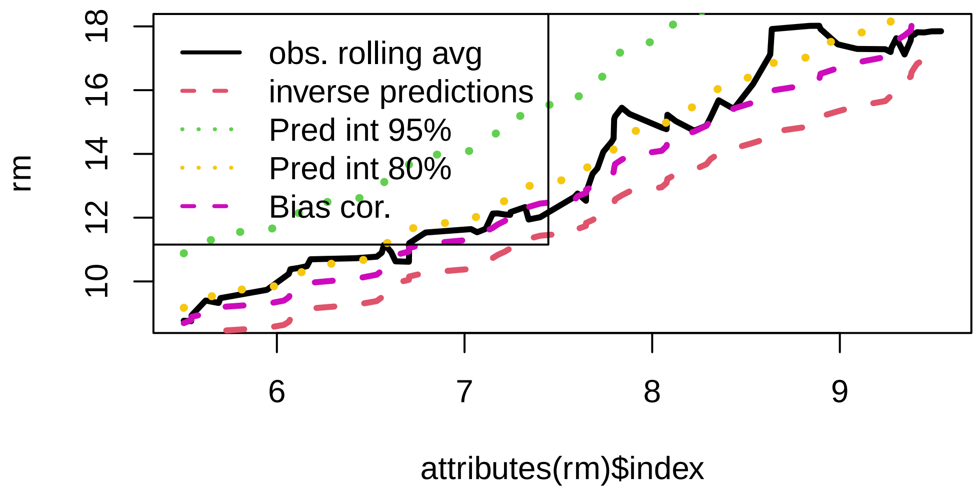
```
Call:
lm(formula = log(Y) ~ X, data = df)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.13484 -0.31721 -0.02877  0.29765  0.85929

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.00060    0.21048   4.754 6.85e-06 ***
X            0.19333    0.02721   7.106 1.94e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4113 on 98 degrees of freedom
Multiple R-squared:  0.3401,	Adjusted R-squared:  0.3333
F-statistic:  50.5 on 1 and 98 DF,  p-value: 1.937e-10
```

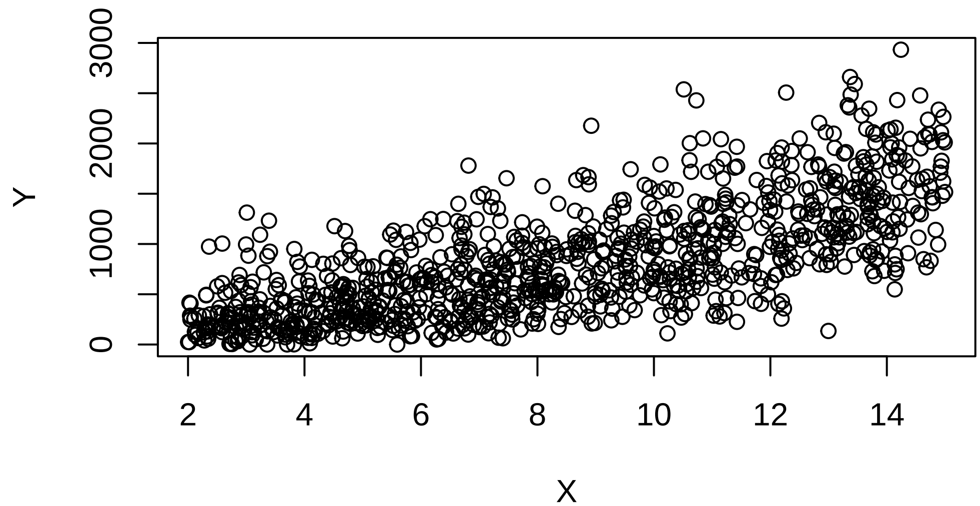
```
s=summary(model)$sigma
# Rolling average



zb=zoo::zoo(x=df$Y,df$X)
```



```
rm=zoo::rollmean(zb,25)

plot(attributes(rm)$index,rm,lty=1,lwd=3,type='l')
# plot(X,Y)

zb=zoo::zoo(x=exp(fitted.values(model)),df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=2,lty=2,lwd=3)

lines(attributes(rm)$index,rm*exp(s^2/2),col=6,lty=2,lwd=3)

# Intervals for the mean at each time point
nd=data.frame("X"=df$X)
ivs=predict(model, newdata = nd,interval = 'prediction')[,2:3]
intervals=rowMeans(exp(ivs))
zb=zoo::zoo(x=intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=3,lty=3,lwd=4)


# Intervals for the mean at each time point - notice when we lower the level the performan
nd=data.frame("X"=df$X)
ivs=predict(model, newdata = nd,interval = 'prediction', level = 0.8)[,2:3]
intervals=rowMeans(exp(ivs))
zb=zoo::zoo(x=intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=7,lty=3,lwd=4)




# Intervals for the mean at each time point using confidence intervals
# ivs=predict(model, newdata = nd,interval = 'confidence', level = 0.8)[,2:3]
# intervals=rowMeans(exp(ivs))
# zb=zoo::zoo(x=intervals,df$X)
# rm=zoo::rollmean(zb,25)
# lines(attributes(rm)$index,rm,col=6,lty=3,lwd=3)

legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Pred in
```




```r
set.seed(2352)

# Simulate data

n=1000
X=runif(n,2,15)
sq=rnorm(n,10+2*X,7)
Y=sq^2
plot(X,Y)
```




```r
df=data.frame(cbind("X"=X,"Y"=Y))
df=df[order(X),]

# Fitting the model
model=lm(sqrt(Y)~X,data=df)
summary(model)
```

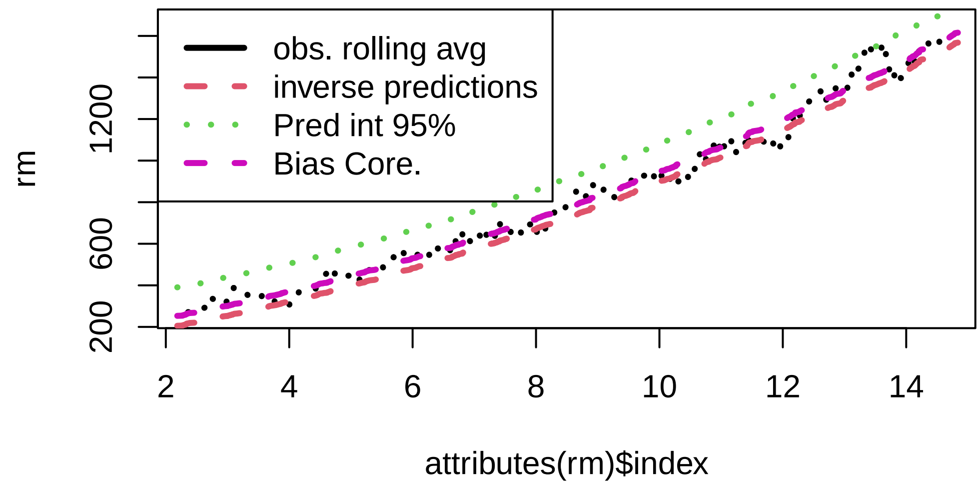
```
Call:
lm(formula = sqrt(Y) ~ X, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-24.2975  -4.5612  -0.0315   4.7083  20.2670 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.94439    0.54223   18.34   <2e-16 ***
X            1.99810    0.05897   33.88   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.919 on 998 degrees of freedom
Multiple R-squared:  0.535, Adjusted R-squared:  0.5345
F-statistic:  1148 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
s=summary(model)$sigma
# plot(X,Y)
# Rolling average



zb=zoo::zoo(x=df$Y,df$X)
rm=zoo::rollmean(zb,50)

plot(attributes(rm)$index,rm,col=1,lty=3,lwd=3,type='l')
zb=zoo::zoo(fitted.values(model)^2,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=2,lty=2,lwd=3)
lines(attributes(rm)$index,rm+s^2,col=6,lty=2,lwd=3)


# Intervals for the mean at each time point
intervals=rowMeans(predict.lm(model,interval = 'prediction')[,2:3]^2)
```

```
Warning in predict.lm(model, interval = "prediction"): predictions on current data refer to _
```

```
zb=zoo::zoo(intervals,df$X)
rm=zoo::rollmean(zb,25)
lines(attributes(rm)$index,rm,col=3,lty=3,lwd=3)
legend("topleft",legend=c("obs. rolling avg","inverse predictions","Pred int 95%","Bias Co
```



We see that the bias correction is the best performing method. However, this involves working out the bias for each transformation. For a complicated transformation, this may be quite difficult. For common transformations, this has already been completed for us.

Let's do an example with some real data. The following example is taken from the textbook:

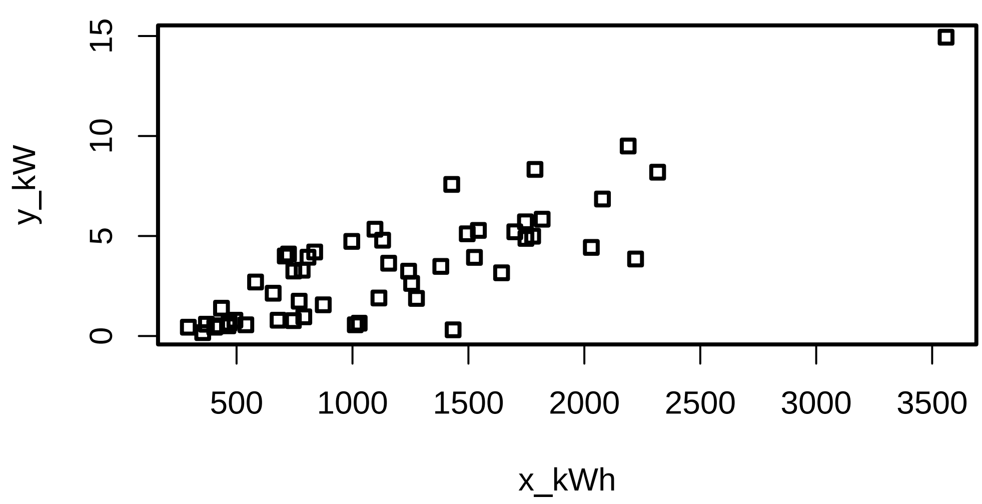
Example 5.2. An electric utility is interested in developing a model relating peak - hour demand Y to total energy usage during the month X. This is an important planning problem because while most customers pay directly for energy usage (in kilowatt - hours), the generation system must be large enough to meet the maximum demand imposed. Data for 53 residential customers for the month of August is given below.

```
# Electric Utility Data

df<- data.frame(
    Customer = c(1:53),
    x_kWh = c(679, 292, 1012, 493, 582, 1156, 997, 2189, 1097, 2078, 1818, 1700, 747, 2030
    y_kW = c(0.79, 0.44, 0.56, 0.79, 2.70, 3.64, 4.73, 9.50, 5.34, 6.85, 5.84, 5.21, 3.25,
  )



df
```

	Customer	x_kWh	y_kW
1	1	679	0.79
2	2	292	0.44
3	3	1012	0.56
4	4	493	0.79
5	5	582	2.70
6	6	1156	3.64
7	7	997	4.73
8	8	2189	9.50
9	9	1097	5.34
10	10	2078	6.85
11	11	1818	5.84
12	12	1700	5.21
13	13	747	3.25
14	14	2030	4.43
15	15	1643	3.16
16	16	414	0.50
17	17	354	0.17
18	18	1276	1.88
19	19	745	0.77
20	20	435	1.39
21	21	540	0.56
22	22	874	1.56
23	23	1543	5.28
24	24	1029	0.64
25	25	710	4.00
26	26	1434	0.31
27	27	837	4.20
28	28	1748	4.88
29	29	1381	3.48
30	30	1428	7.58
31	31	1255	2.63
32	32	1777	4.99
33	33	370	0.59
34	34	2316	8.19
35	35	1130	4.79
36	36	463	0.51
37	37	770	1.74
38	38	724	4.10
39	39	808	3.94
40	40	790	0.96
41	41	783	3.29
42	42	406	0.44

```
43         43  1242   3.24
44         44   658   2.14
45         45  1746   5.71
46         46   468   0.64
47         47  1114   1.90
48         48   413   0.51
49         49  1787   8.33
50         50  3560  14.94
51         51  1495   5.11
52         52  2221   3.85
53         53  1526   3.93
```

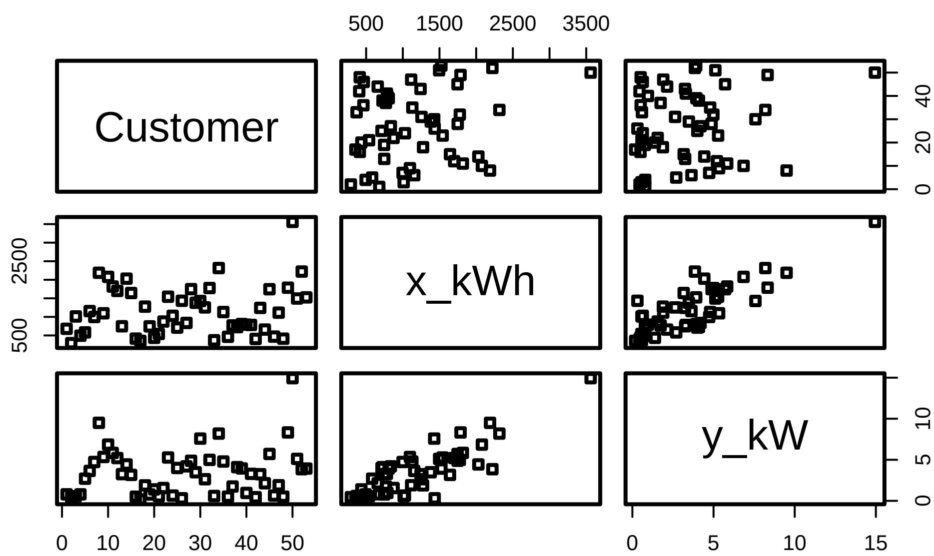
```
#  changing the plot aesthetics
par(pch=22,lwd=2)


# Explore

plot(df[,2:3])
```




```
summary(df)
```

```
    Customer        x_kWh           y_kW       
 Min.   : 1   Min.   : 292   Min.   : 0.170  
 1st Qu.:14   1st Qu.: 679   1st Qu.: 0.790  
 Median :27   Median :1029   Median : 3.250  
 Mean   :27   Mean   :1153   Mean   : 3.413  
 3rd Qu.:40   3rd Qu.:1543   3rd Qu.: 4.880  
 Max.   :53   Max.   :3560   Max.   :14.940  
```

```r
plot(df)
```



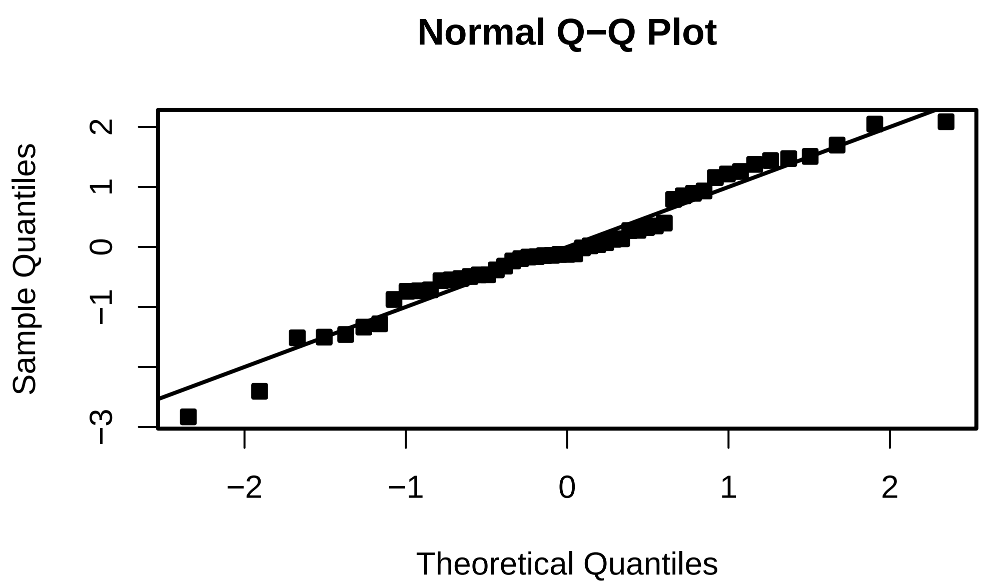

```r
# Model

model=lm(y_kW~x_kWh, df); model
```

```
Call:
lm(formula = y_kW ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh  
  -0.831304     0.003683  
```

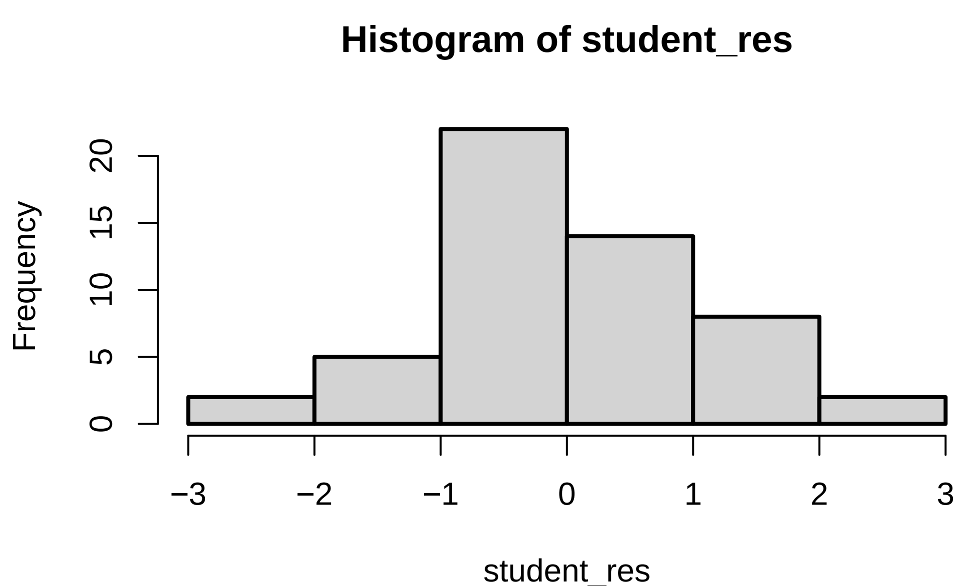
```r
summ=summary(model); summ
```

```

Call:
lm(formula = y_kW ~ x_kWh, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.1399 -0.8275 -0.1934  1.2376  3.1522 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.8313037  0.4416121  -1.882   0.0655 .  
x_kWh        0.0036828  0.0003339  11.030 4.11e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.577 on 51 degrees of freedom
Multiple R-squared:  0.7046,	Adjusted R-squared:  0.6988 
F-statistic: 121.7 on 1 and 51 DF,  p-value: 4.106e-15
```

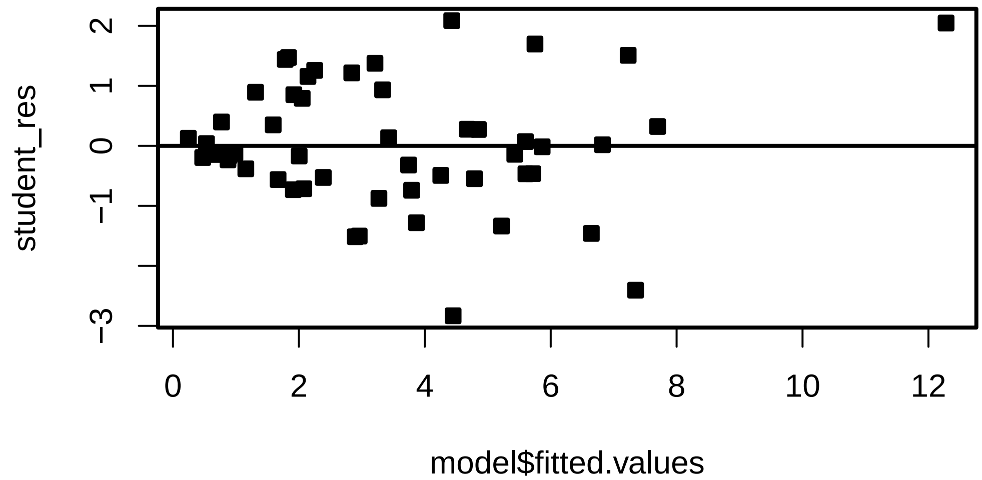
```r
# Now do the residual analysis
# Studentized residuals
student_res=rstudent(model)
MSE=summ$sigma^2

qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```




```r
hist(student_res,breaks=6)
```



```r
plot(model$fitted.values,student_res,pch=22,bg=1)
abline(h=0)
```



We see that the residual variance increases with the mean of Y. This is easily seen by the fan shape of the residuals in the plot of the residuals against the fitted values.

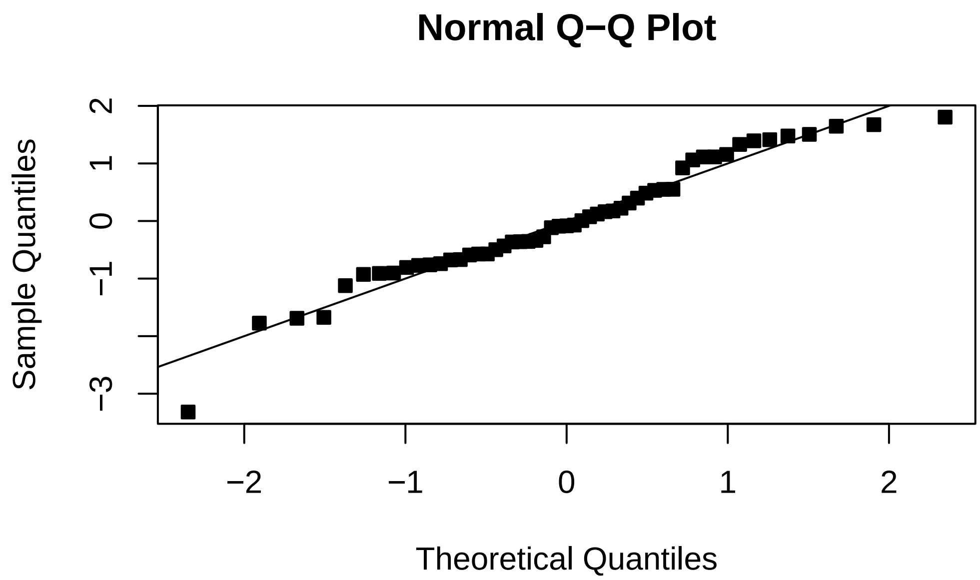
```r
######## Let's try the sqrt transformation

model2=lm(sqrt(y_kW)~x_kWh, df)
model2
```

```

Call:
lm(formula = sqrt(y_kW) ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh
  0.5822259    0.0009529
```

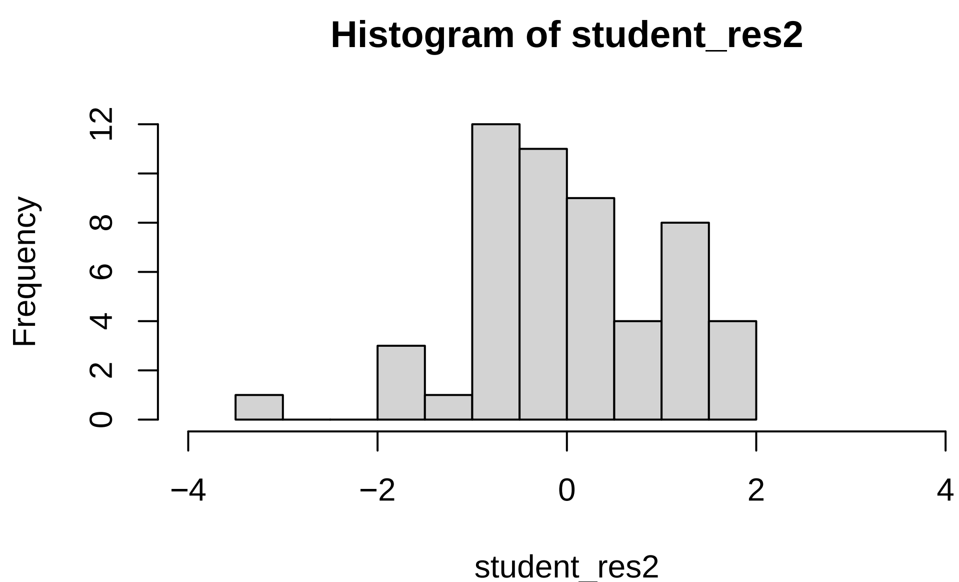
```r
summ2=summary(model2); summ2
```

```
Call:
lm(formula = sqrt(y_kW) ~ x_kWh, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.39185 -0.30576 -0.03875  0.25378  0.81027 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.822e-01  1.299e-01   4.481 4.22e-05 ***
x_kWh       9.529e-04  9.824e-05   9.699 3.61e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.464 on 51 degrees of freedom
Multiple R-squared:  0.6485,    Adjusted R-squared:  0.6416 
F-statistic: 94.08 on 1 and 51 DF,  p-value: 3.614e-13
```

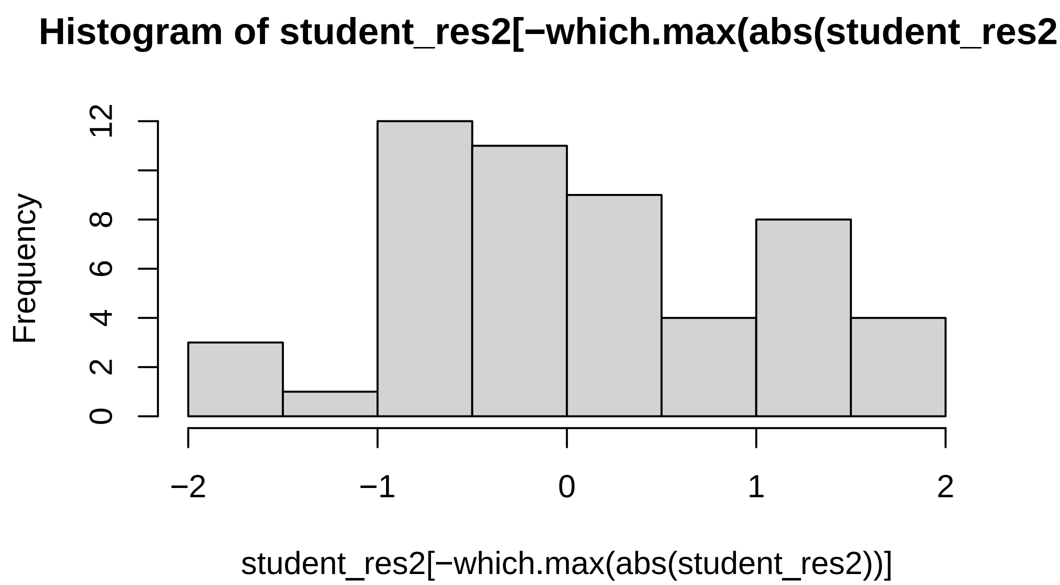
```r
student_res2=rstudent(model2)
MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



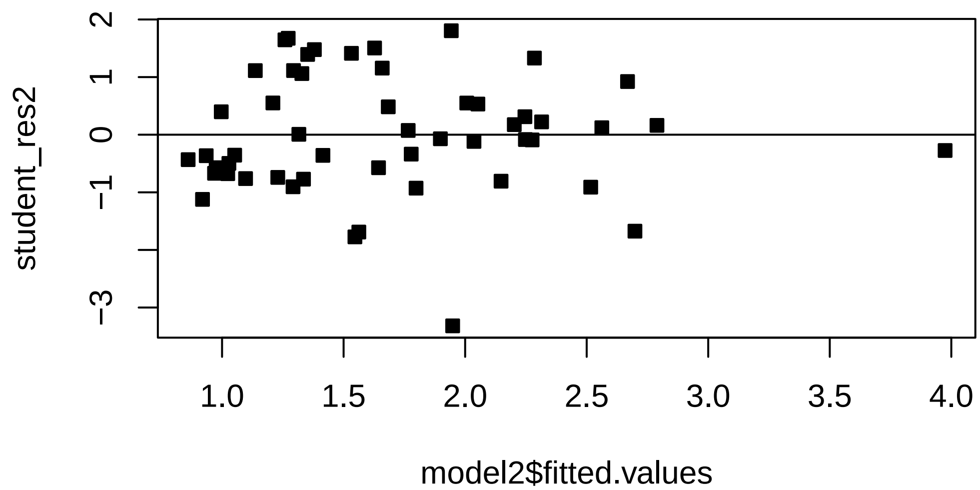

```r
hist(student_res2,breaks=10,xlim=c(-4,4))
```



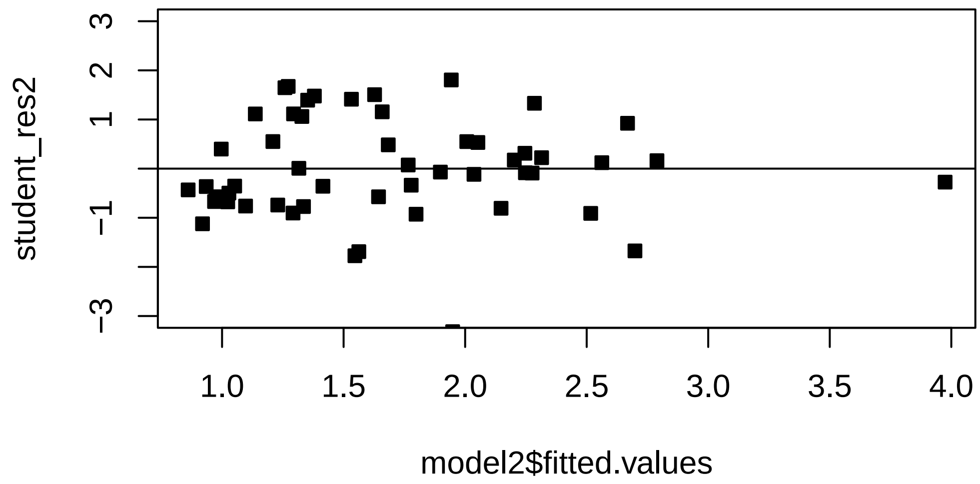
```
hist(student_res2[-which.max(abs(student_res2))],breaks=10,xlim=c(-2,2))
```



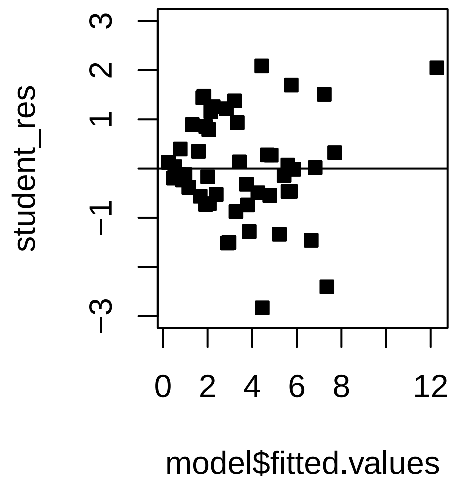

```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```

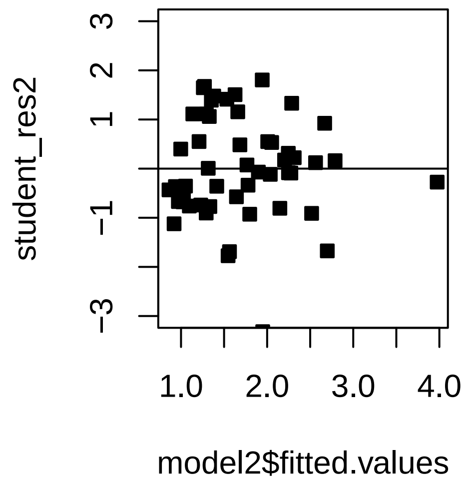



```
# There is one large outlier skewing the previous plot. Let's rescale and remove.
plot(model2$fitted.values,student_res2,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```




```
# Compare!
par(mfrow=c(1,2))
plot(model$fitted.values,student_res,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
plot(model2$fitted.values,student_res2,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```





We see that the transformation has solved the problem. Note that sometimes, even though the square-root transformation may be more suitable, the analyst may opt for the logarithm transform. This is because the log transformation gives a nicer interpretation to the coefficients. In this case, that is not working well, see below:

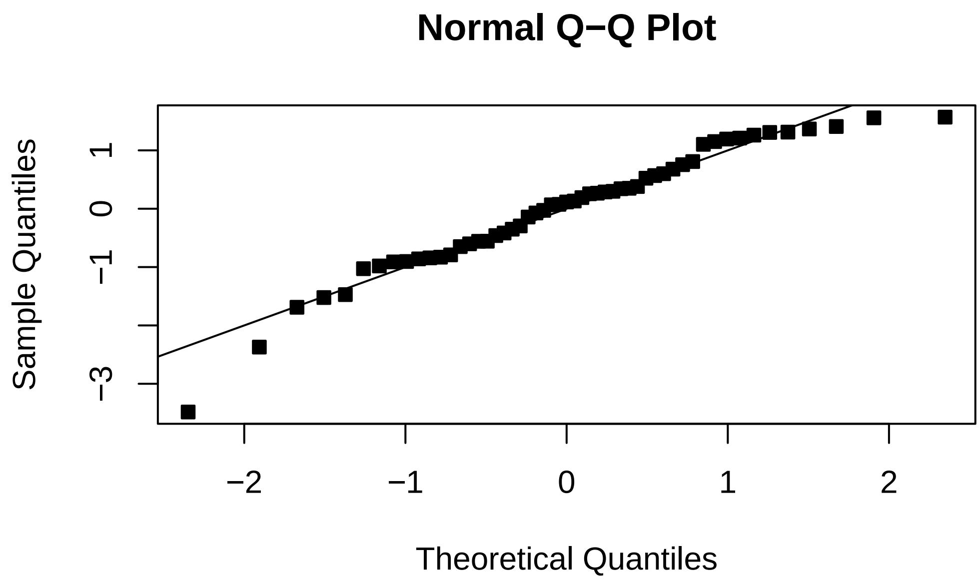
```
######## Let's try the log transformation
par(mfrow=c(1,1))


model3=lm(log(y_kW)~x_kWh, df)
model3
```

```
Call:
lm(formula = log(y_kW) ~ x_kWh, data = df)

Coefficients:
(Intercept)        x_kWh
  -0.558713     0.001172
```

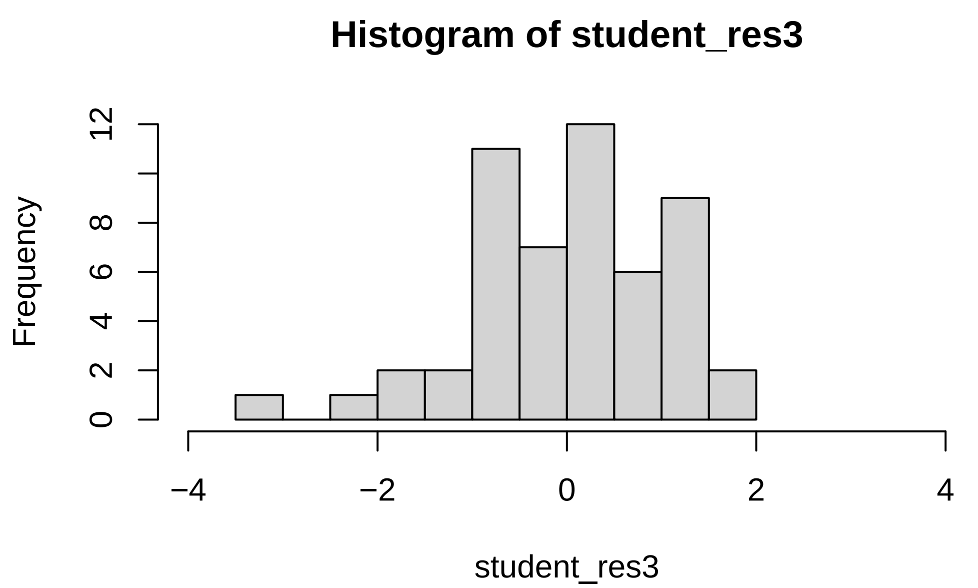
```
summ3=summary(model3); summ3
```

```
Call:
lm(formula = log(y_kW) ~ x_kWh, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.29261 -0.47256  0.08414  0.49628  1.12143 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.5587131  0.2057201  -2.716    0.009 ** 
x_kWh        0.0011716  0.0001555   7.533 7.86e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7347 on 51 degrees of freedom
Multiple R-squared:  0.5266,	Adjusted R-squared:  0.5174 
F-statistic: 56.74 on 1 and 51 DF,  p-value: 7.862e-10
```

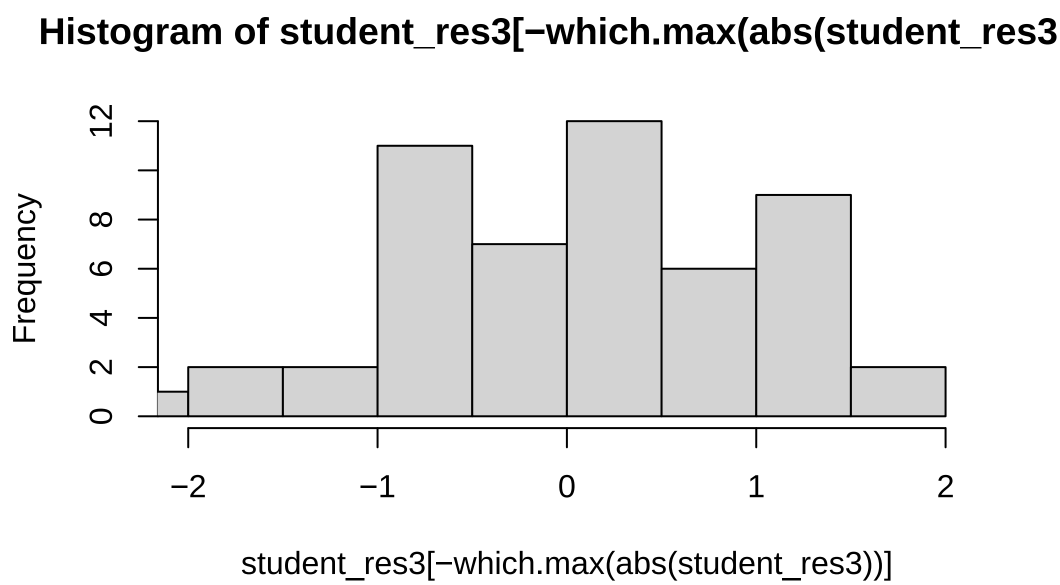
```r
student_res3=rstudent(model3)
MSE3=summ3$sigma^2
qqnorm(student_res3,pch=22,bg=1)
abline(0,1)
```



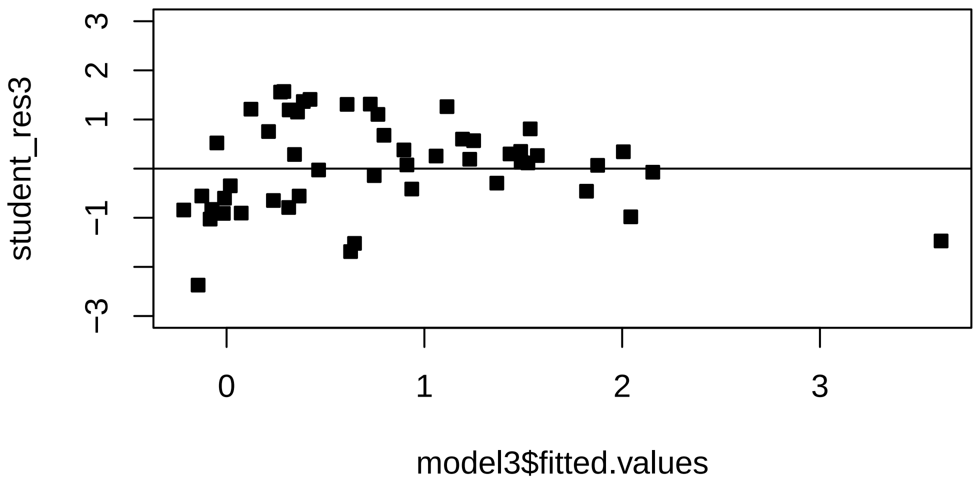

```r
hist(student_res3,breaks=10,xlim=c(-4,4))
```



```r
hist(student_res3[-which.max(abs(student_res3))],breaks=10,xlim=c(-2,2))
```




```r
plot(model3$fitted.values,student_res3,pch=22,bg=1,ylim=c(-3,3))
abline(h=0)
```



5.2 Linearizing the model

Moving on, we may suspect that the relationship between the regressors and the response is nonlinear, either through empirical evidence or theoretical justification. In some cases a nonlinear function can be linearized by using a suitable transformation. Such nonlinear models are called intrinsically linear. For example, consider the model $Y = \beta_0 e^{\beta_1 X}\epsilon$. Taking the log of both sides yields:

$$\log(Y) = \log(\beta_0) + \beta_1 X + \log(\epsilon).$$

Reparameterizing with $Z = \log(Y)$, $\alpha_0 = \log(\beta_0)$ and $\eta = \log(\epsilon)$, we have that

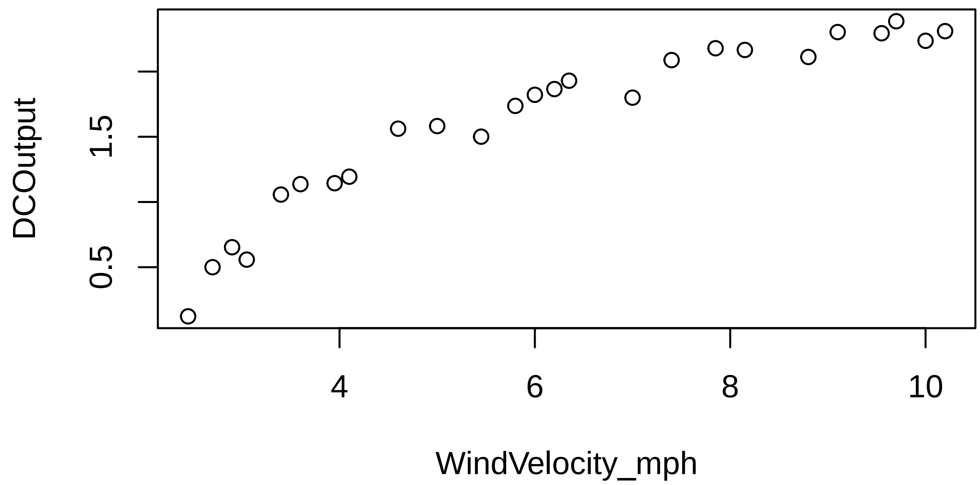
$$Z = \alpha_0 + \beta_1 X + \eta.$$

If we are willing to assume that η are symmetric about 0 with a constant variance, then we can run a linear regression with the model given above. To get estimates for the original units Y, we can transform back as previously discussed. A model is linearizable if there exists some reparameterization which places the model in the form of the MLR.

Exercise 5.1. Show the following models are linearizable - that is, find the linear reparameterization of the following models: 1. $Y = \beta_0 X_1^{\beta}$ 2. $Y = \beta_0 X^{\beta_1 X}$ 3. $Y = \beta_0 + \log X$ 4. $Y = X/(\beta_0 X - \beta_1)$

Example 5.3. A research engineer is investigating the use of a windmill to generate electricity. He has collected data on the DC output from his windmill and the corresponding wind velocity. See below. Find a well-fitting regression model for this data.

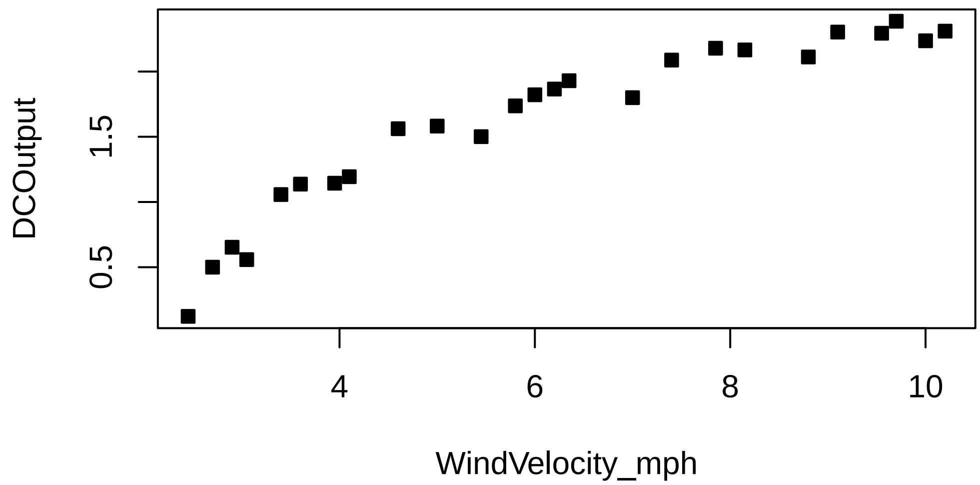
```
########### Windmill data




# Create the data frame
df_wind <- data.frame(
  WindVelocity_mph = c(5.00, 6.00, 3.40, 2.70, 10.00, 9.70, 9.55, 3.05, 8.15, 6.20,
                       2.90, 6.35, 4.60, 5.80, 7.40, 3.60, 7.85, 8.80, 7.00, 5.45,
                       9.10, 10.20, 4.10, 3.95, 2.45),
  DCOutput = c(1.582, 1.822, 1.057, 0.500, 2.236, 2.386, 2.294, 0.558, 2.166, 1.866,
               0.653, 1.930, 1.562, 1.737, 2.088, 1.137, 2.179, 2.112, 1.800, 1.501,
               2.303, 2.310, 1.194, 1.144, 0.123)
)

###############


par(mfrow=c(1,1))
plot(df_wind)
```




```
summary(df_wind)
```

```
WindVelocity_mph    DCOutput
Min.   : 2.450   Min.   :0.123
1st Qu.: 3.950   1st Qu.:1.144
Median : 6.000   Median :1.800
Mean   : 6.132   Mean   :1.610
3rd Qu.: 8.150   3rd Qu.:2.166
Max.   :10.200   Max.   :2.386
```

```
plot(df_wind,pch=22,bg=1)
```




```r
model=lm(DCOutput~WindVelocity_mph, df_wind)
model
```

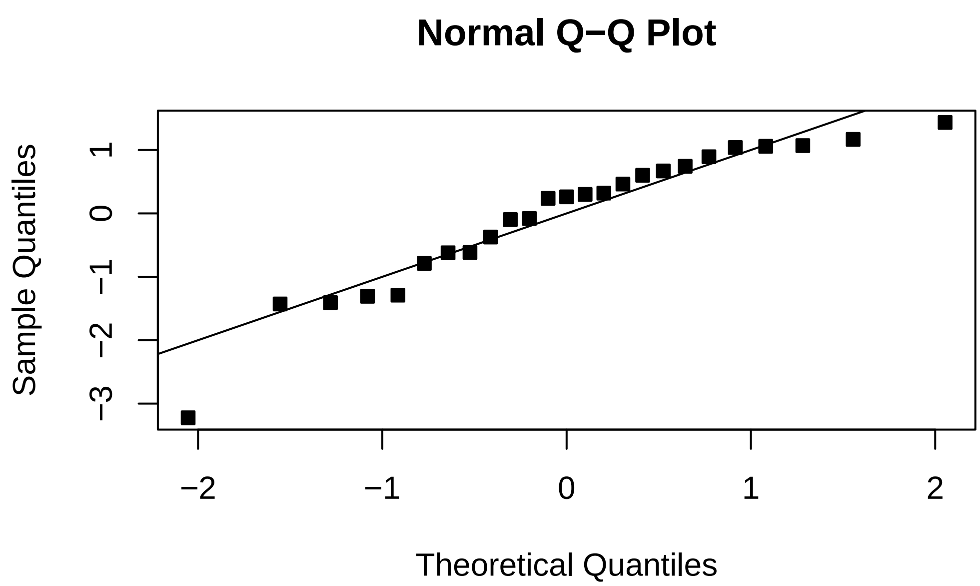
```
Call:
lm(formula = DCOutput ~ WindVelocity_mph, data = df_wind)

Coefficients:
     (Intercept)  WindVelocity_mph
          0.1309            0.2411
```

```r
summ=summary(model); summ
```

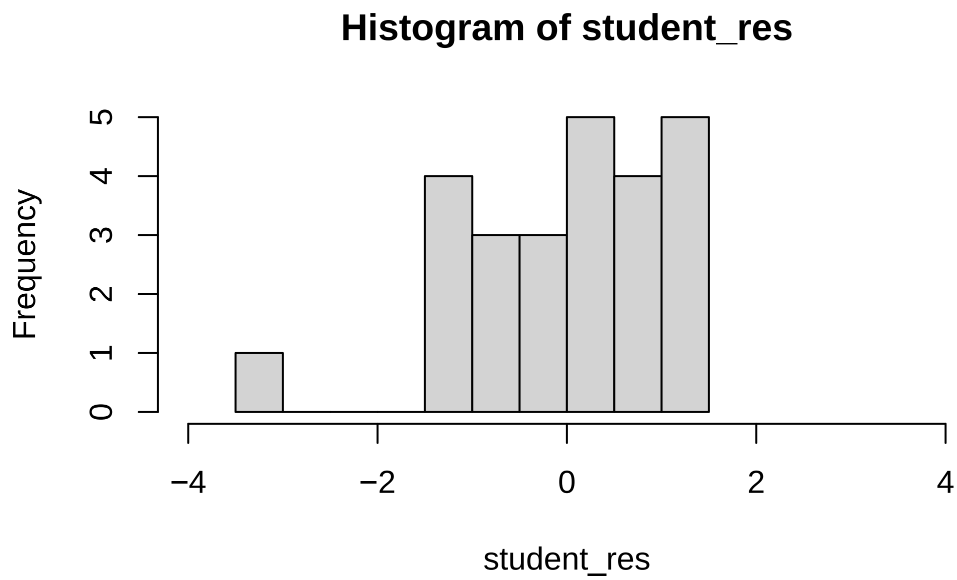
```
Call:
lm(formula = DCOutput ~ WindVelocity_mph, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max
-0.59869 -0.14099  0.06059  0.17262  0.32184
```

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.13088    0.12599   1.039     0.31
WindVelocity_mph  0.24115    0.01905  12.659 7.55e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2361 on 23 degrees of freedom
Multiple R-squared:  0.8745,    Adjusted R-squared:  0.869
F-statistic: 160.3 on 1 and 23 DF,  p-value: 7.546e-12
```

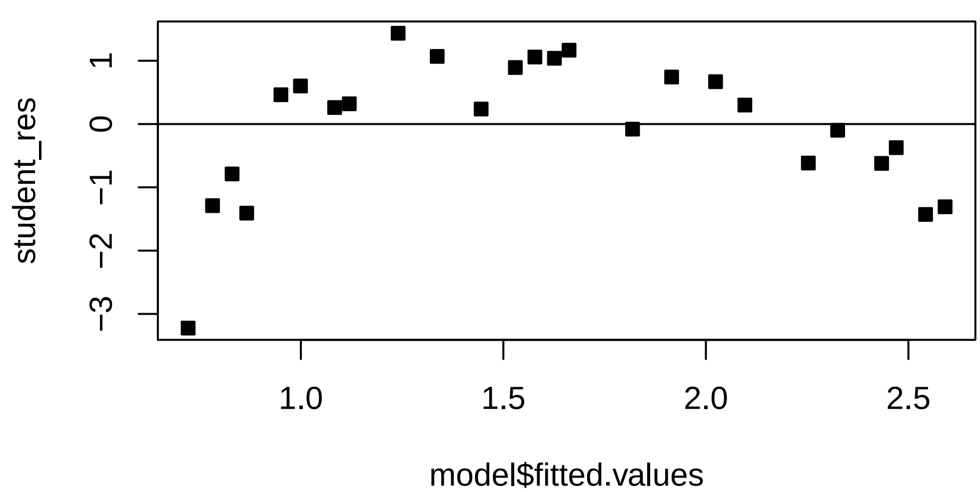
```r
student_res=rstudent(model)

MSE=summ$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```




```r
hist(student_res,breaks=10,xlim=c(-4,4))
```

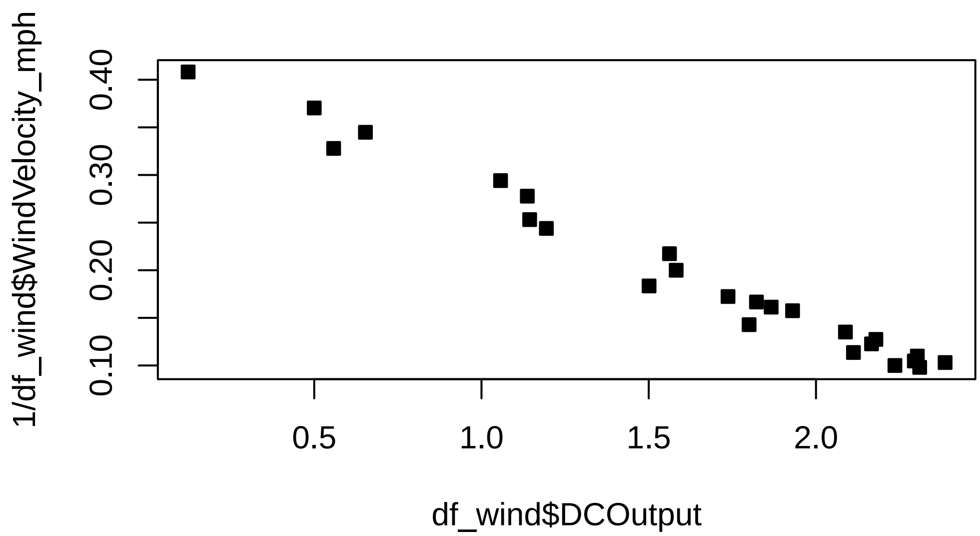



```
plot(model$fitted.values,student_res,pch=22,bg=1)
abline(h=0)
```



The fit is not good. Looking at the scatterplot, we might initially consider using a quadratic model to account for the pictured curvature. However, the scatterplot suggests that as wind speed increases, DC output approaches an upper limit of approximately 2.5. This is also consistent with the theory of windmill operation. Since the quadratic model will eventually bend downward as wind speed increases, it would not be appropriate for these data. A more reasonable model for the windmill data that incorporates an upper asymptote would be based on $1/X$.

```
plot(df_wind$DCOutput,1/df_wind$WindVelocity_mph,pch=22,bg=1)
```




```
# plot(df$DCOutput,log(df$WindVelocity_mph))
df_wind$WindVelocity_mph_inv=1/df_wind$WindVelocity_mph
model2=lm(DCOutput~WindVelocity_mph_inv, df_wind)
model2
```

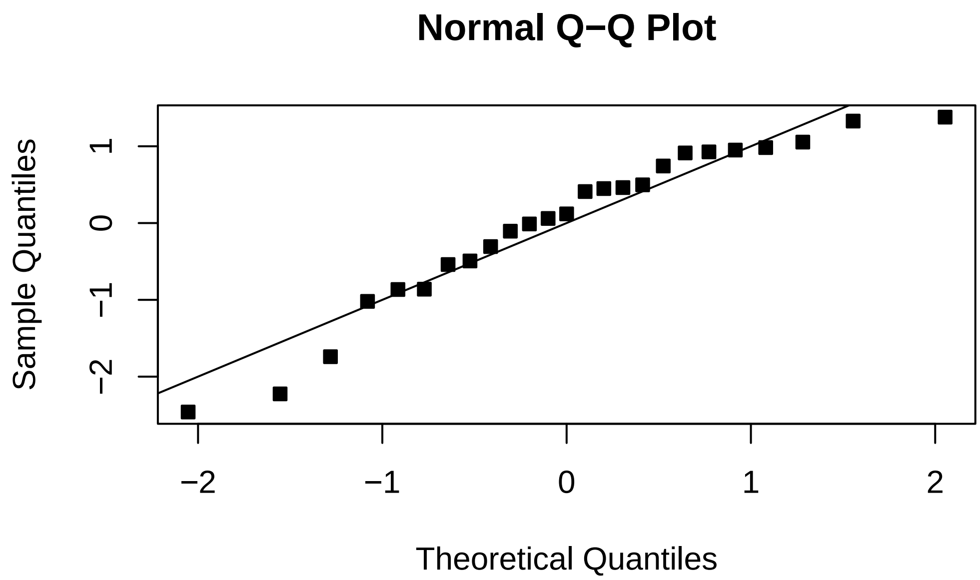
```
Call:
lm(formula = DCOutput ~ WindVelocity_mph_inv, data = df_wind)

Coefficients:
         (Intercept)  WindVelocity_mph_inv
```

```
             2.979                 -6.935
```

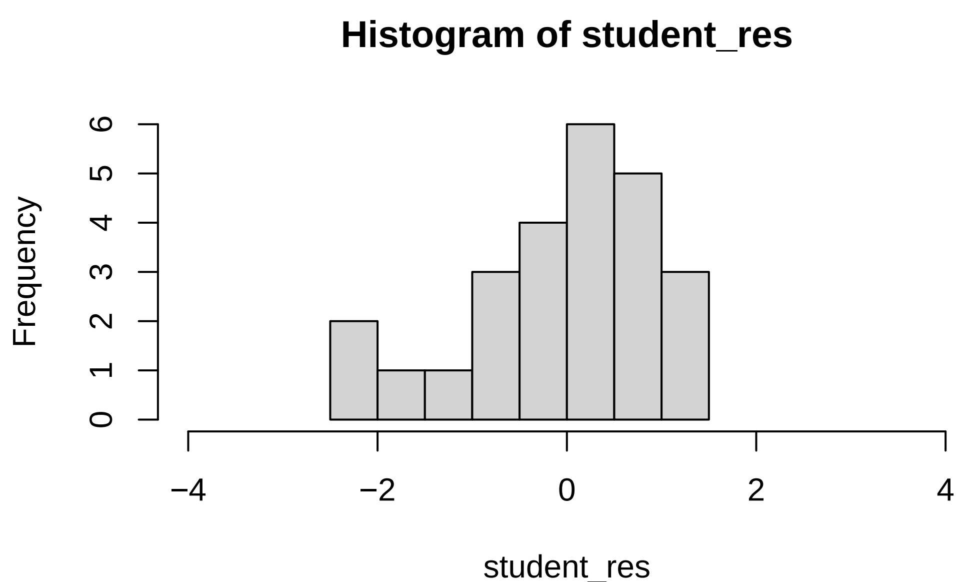
```r
summ=summary(model2); summ
```

```
Call:
lm(formula = DCOutput ~ WindVelocity_mph_inv, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.20547 -0.04940  0.01100  0.08352  0.12204 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)            2.9789     0.0449   66.34   <2e-16 ***
WindVelocity_mph_inv  -6.9345     0.2064  -33.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09417 on 23 degrees of freedom
Multiple R-squared:   0.98, Adjusted R-squared:  0.9792 
F-statistic:  1128 on 1 and 23 DF,  p-value: < 2.2e-16
```

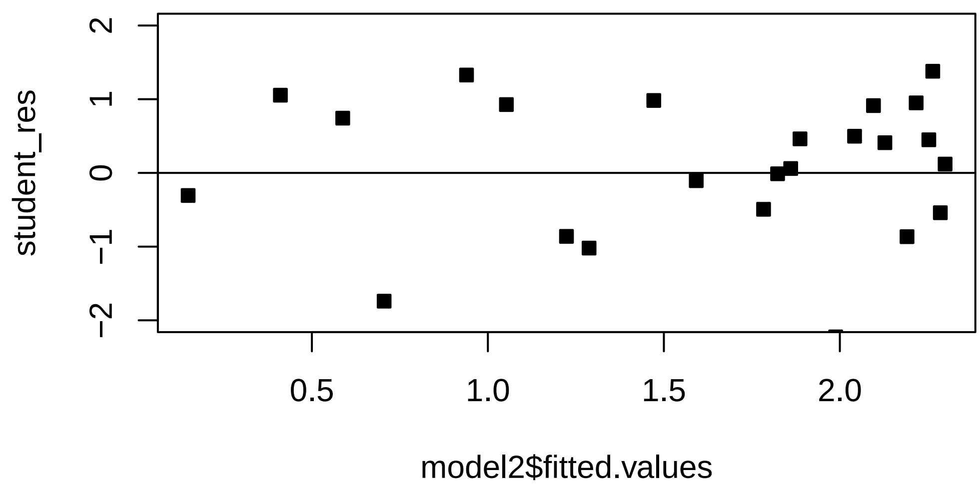
```r
student_res=rstudent(model2)
MSE=summ$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```




```
hist(student_res,xlim=c(-4,4))
```



```
plot(model2$fitted.values,student_res,pch=22,bg=1,ylim=c(-2,2))
abline(h=0)
```



5.3 Box Cox Transformations

One technique is to use the data to estimate which transformation is best, a popular instance is the Box-Cox transformation. Consider the class of transformations: $\{y^\lambda : \lambda \in \mathbb{R}\}$. The regression model and λ can be estimated simultaneously using the method of maximum likelihood. Recall that we used the method of least squares to estimate the model parameters - maximum likelihood is an alternative estimation strategy. Think of λ like an extra model parameter, on top of β and σ that we need to estimate.

Let

$$\tilde{y} = \log^{-1}(1/n \sum_{i=1}^{n} \log y_i)$$

$$y_\lambda = \begin{cases} \dfrac{y^\lambda - 1}{\lambda \tilde{y}^{\lambda-1}} & \lambda \neq 0 \\ \tilde{y} \log y & \lambda = 0 \end{cases}.$$

We then fit the following model

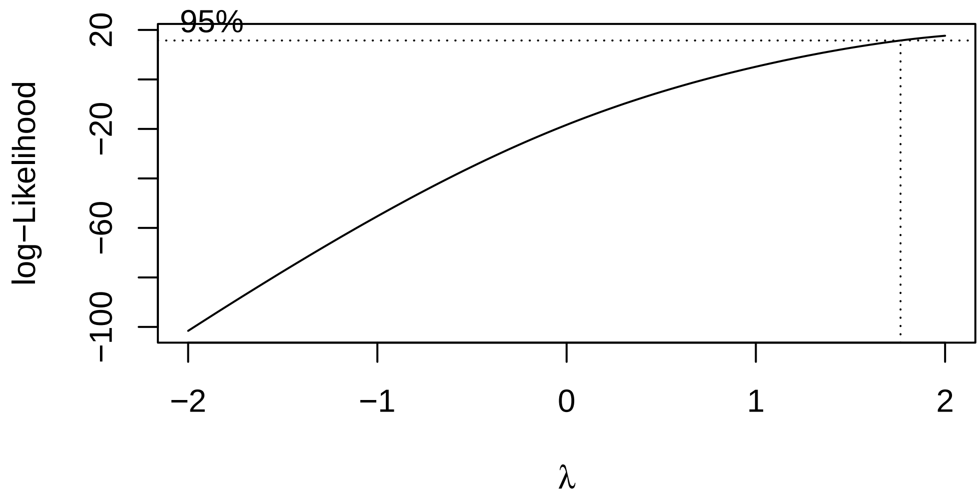
$$y_\lambda = X\beta + \epsilon.$$

Even though $y^\lambda \neq y_\lambda$, we use y^λ (or $\log y$ if $\lambda = 0$) as the final response - as it is more interpretable. It is entirely acceptable to use y_λ as the response for the final model - this model will have a scale difference and an origin shift in comparison to the model using y^λ (or $\log y$). Usually the final λ used in the model is rounded to a nice number for interpretation. A computational procedure is used for estimating λ, which we will not cover here. In general, we can compute a confidence interval for λ, and if it contains 1 then we may not need to transform.

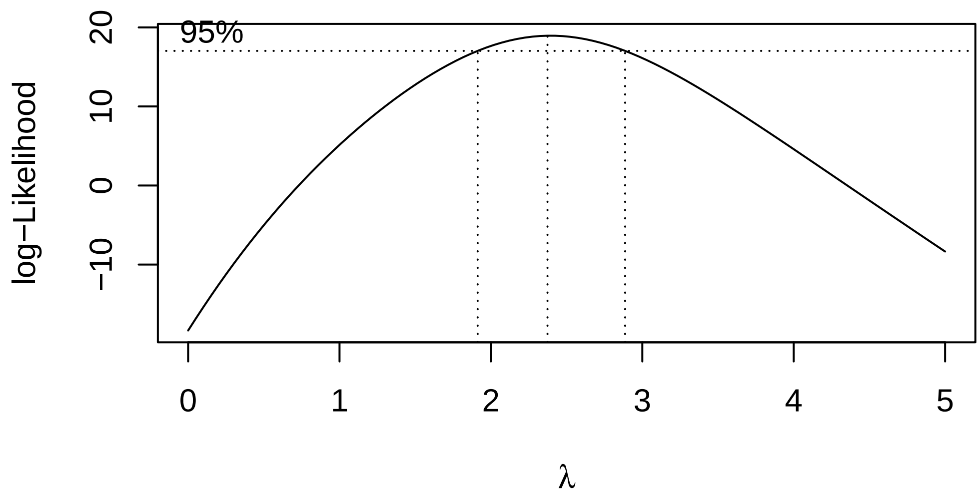
Example 5.4. Let's apply the Box-Cox transformation to the two previous examples.

The R function `boxcox` from the `MASS` package can be used to execute the Box-Cox transformation. It requires you to specify a grid of points for λ, given below by `seq(-2, 2, 1/10)`. We can set the `plotit` parameter to `TRUE` in order to see if this grid is big enough. We should see a peak or mode in the log-likelihood function that is plotted. If we don't, we can expand the grid on the side which has the largest value of the log-likelihood. Observe below that we need to include points higher than 2 on the grid, as the function is still increasing for at $\lambda = 2$:

```
# You need the MASS package.
# install.packages('MASS')

bc=MASS::boxcox(DCOutput~WindVelocity_mph,data=df_wind,
       lambda = seq(-2, 2, 1/10),
       plotit = TRUE,
       eps = 1/50,
       xlab = expression(lambda),
       ylab = "log-Likelihood")
```




```
# bc

#Observe that

bc=MASS::boxcox(DCOutput~WindVelocity_mph,data=df_wind,
                lambda = seq(0, 5, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
```




```
# bc

#Seems like we should try lambda=2
```

The confidence interval goes from just below 2 to just below 3. Let's pick a round number, and try the transformation $\lambda = 2$.

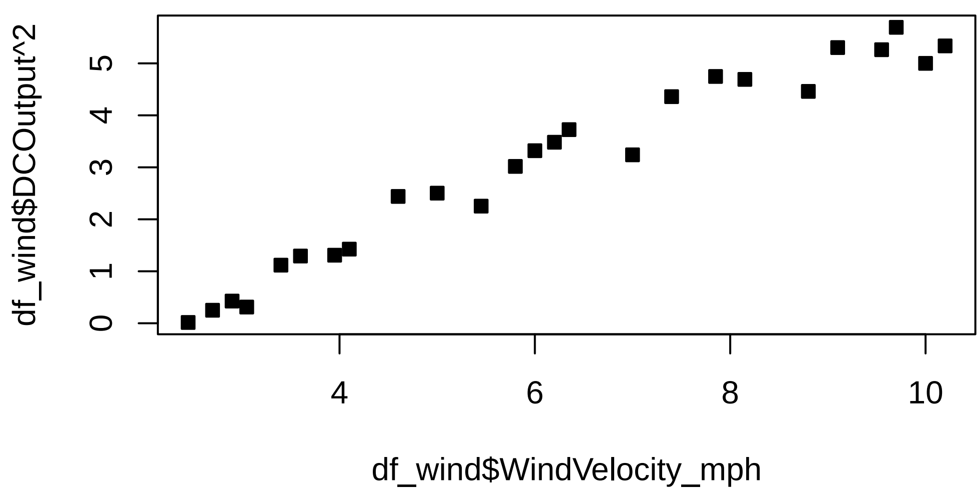
```
# plot(df$DCOutput,log(df$WindVelocity_mph))
model3=lm(DCOutput^2~WindVelocity_mph, df_wind)
model3
```

```
Call:
lm(formula = DCOutput^2 ~ WindVelocity_mph, data = df_wind)

Coefficients:
     (Intercept)  WindVelocity_mph
         -1.3585            0.7107
```

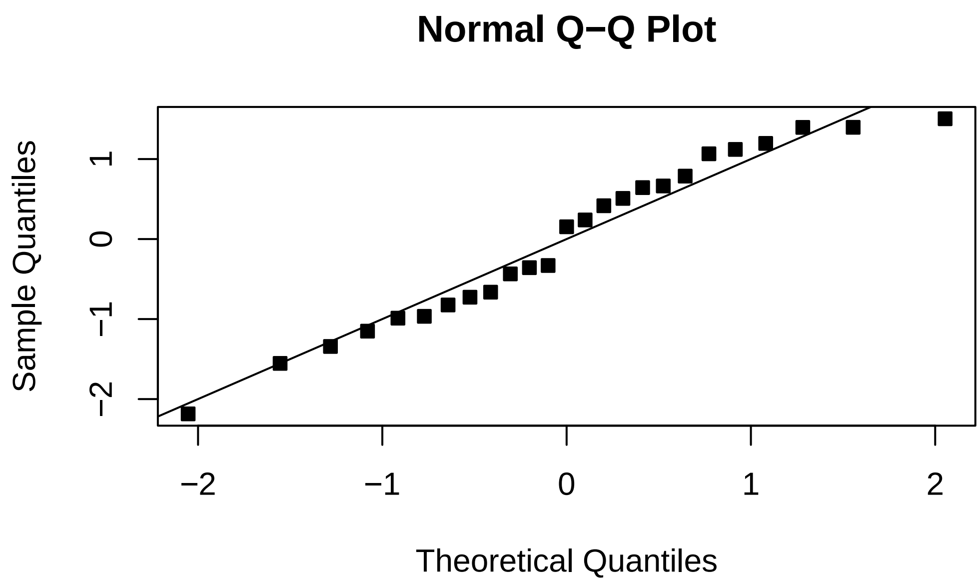
```
summ3=summary(model3); summ3
```

```
Call:
lm(formula = DCOutput^2 ~ WindVelocity_mph, data = df_wind)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.74840 -0.31027  0.05951  0.30793  0.57072 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -1.35851    0.21239  -6.396 1.58e-06 ***
WindVelocity_mph  0.71066    0.03211  22.130  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3979 on 23 degrees of freedom
Multiple R-squared:  0.9551,	Adjusted R-squared:  0.9532 
F-statistic: 489.7 on 1 and 23 DF,  p-value: < 2.2e-16
```

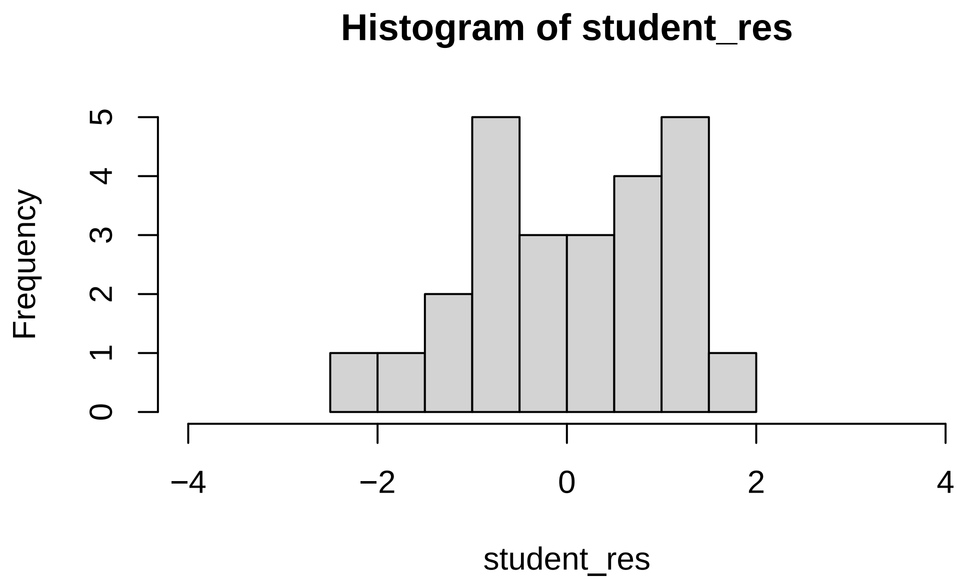
```
plot(df_wind$WindVelocity_mph,df_wind$DCOutput^2,pch=22,bg=1)
```



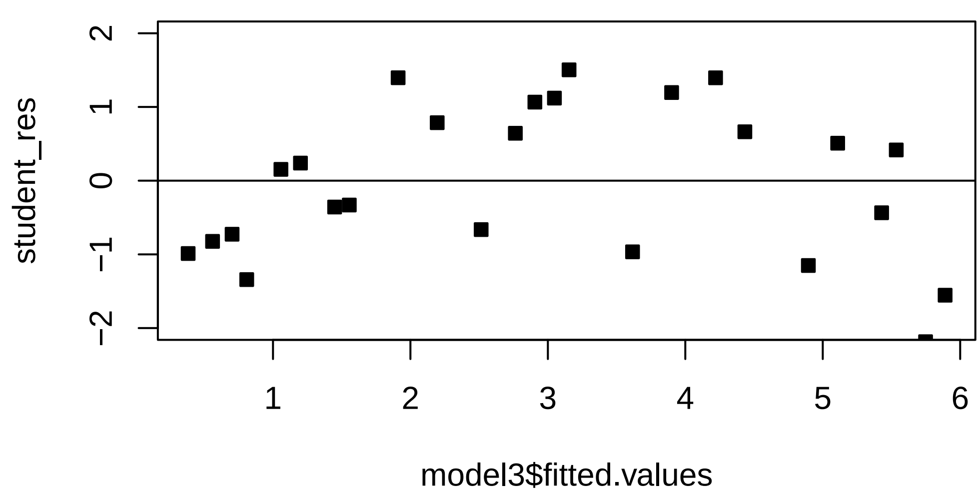
```
student_res=rstudent(model3)
MSE=summ3$sigma^2
qqnorm(student_res,pch=22,bg=1)
abline(0,1)
```




```
hist(student_res,xlim=c(-4,4))
```



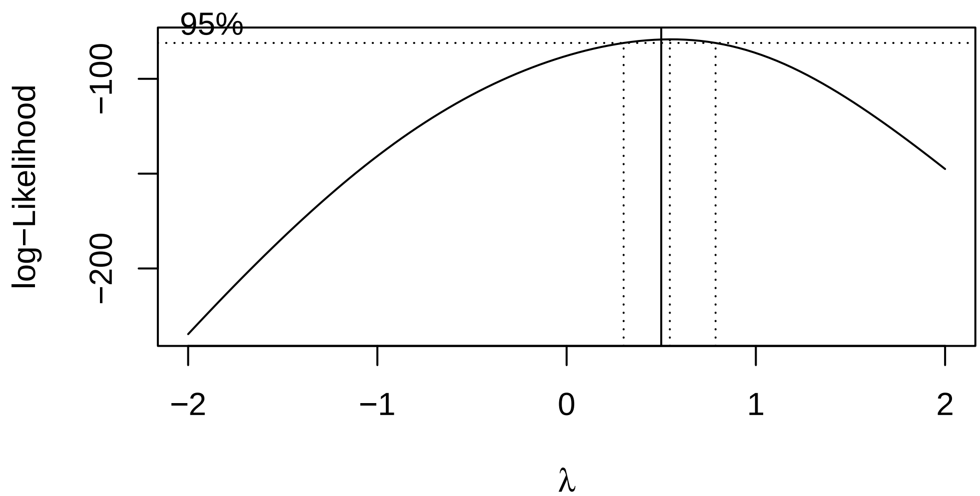

```r
plot(model3$fitted.values,student_res,pch=22,bg=1,ylim=c(-2,2))
abline(h=0)
```



The fit is not bad. The R^2 is very high. There is a pattern in the QQplot and a slight pattern in the residuals plot. For knowing nothing about wind velocity, it is not bad.

The electricity data clearly points to the square root transformation - matching the analysis we did previously.

```
## Electricity

model=lm(y_kW~x_kWh, df)

bc=MASS::boxcox(y_kW~x_kWh,data=df,
                lambda = seq(-2, 2, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
abline(v=0.5)
```




```
# bc
```

5.4 Homework stop

Complete the assigned Chapter 5 questions and complete assignment 2.

6 Indicator Variables

Often we will have regressors that are categorical. We now discuss how to include those in a regression model. In general, categorical variables can be included in a regression model via **indicator variables**.

6.1 What are indicator variables?

If a regressor has two categories A and B, that regressor can be included in the model as

$$z = \begin{cases} 0 & \text{if the observation is type A} \\ 1 & \text{if the observation is type B} \end{cases}$$

Sometimes people choose

$$z = \begin{cases} -1 & \text{if the observation is type A} \\ 1 & \text{if the observation is type B} \end{cases}.$$

The variable z is an indicator variable. Indicator variables are in numeric form, and can therefore be included in the design matrix X in the usual way we do for continuous regressors.

Example 6.1. Let's recall Example 3.1 and suppose we have some new data as follows:

It is difficult to accurately determine a person's body fat percentage without immersing them in water. However, we can easily obtain the weight of a person. A researcher would like to know if weight and body fat percentage are related? They also suspect that sex plays a role in the prediction. This researcher collected the following data:

Individual	1	2	3	4	5	6	7	8	9	10
Weight (lb)	175	181	200	159	196	192	205	173	187	188
Body Fat (%)	6	21	15	6	22	31	32	21	25	30
Sex	F	M	F	F	M	F	F	M	M	F
———	——	——	——	——	——	——	——	——	——	——
Individual	11	12	13	14	15	16	17	18	19	20

Individual	1	2	3	4	5	6	7	8	9	10
Weight (lb)	188	240	175	168	246	160	215	159	146	219
Body Fat (%)	10	20	22	9	38	10	27	12	10	28
Sex	F	F	M	M	F	F	M	F	F	M

Write out the appropriate indicator variable for Sex. Interpret the resulting regression equation for regressing Body fat against weight and sex. Interpret the coefficient that corresponds to the Sex variable.

We have that

$$X_{2i} = \begin{cases} 0 & \text{if the subject } i \text{ is male} \\ 1 & \text{if the subject } i \text{ is female} \end{cases}.$$

The regression equation is then

$$Y_i|X = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i,$$

where X_{i1} is the weight of individual i.

When $X_{i2} = 0$, then the regression equation for males is given by $Y_i|X = \beta_0 + \beta_1 X_{i1} + \epsilon_i$. Therefore, the expected body fat percentage for males is $\mathrm{E}[Y_i|X] = \beta_0 + \beta_1 X_{i1}$. Additionally, when $X_{i2} = 1$, then the regression equation for females is given by $Y_i|X = \beta_0 + \beta_1 X_{i1} + \beta_2 + \epsilon_i$. It follows that the expected body fat percentage for females is $\mathrm{E}[Y_i|X] = \beta_0 + \beta_1 X_{i1} + \beta_2$. Thus, we have that the expected body fat percentage for females is β_2 higher than for males, holding weight constant. This is the interpretation of the coefficient for the dummy variable in this case. Observe that for males and females, the regression lines are parallel. The model says that sex accounts for a constant shift in your expected body fat, but the slope (the coefficient for weight) of the regression line remains the same.

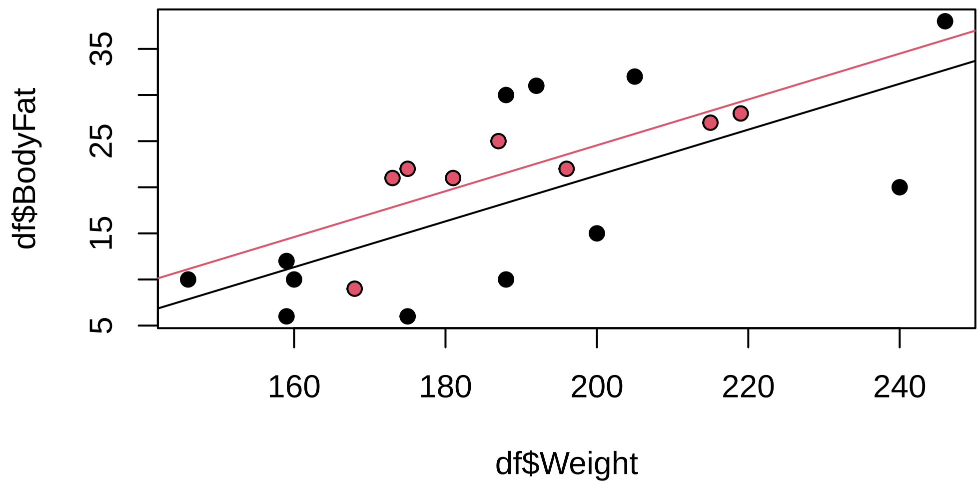
Let's observe.

```
# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )
Sex=c("F","M","F","F","M","F","F","M","M","F","F","F","M","M","F","F","M","F","F","M")

df=data.frame(Weight=Weight,BodyFat=BodyFat,Sex=Sex,stringsAsFactors = T)

df$Sex=relevel(df$Sex,"M")
```

```r
mod=lm(BodyFat~Weight+Sex,data=df)
summary(mod)
```

```

Call:
lm(formula = BodyFat ~ Weight + Sex, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.2198  -5.3804  -0.1767   3.6719  11.7136 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -25.17526   11.73681  -2.145 0.046695 *  
Weight        0.24861    0.06061   4.102 0.000743 ***
SexF         -3.27233    3.21493  -1.018 0.323014    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.042 on 17 degrees of freedom
Multiple R-squared:  0.5149,	Adjusted R-squared:  0.4578 
F-statistic: 9.021 on 2 and 17 DF,  p-value: 0.002137
```

```r
plot(df$Weight,df$BodyFat,bg=((df$Sex=="M")+1),pch=21)
abline(coef(mod)[1],coef(mod)[2],col=2)
abline(coef(mod)[1]+coef(mod)[3],coef(mod)[2],col=1)
```



We can generalize this idea out of this example. In the case of two categories, the interpretation of the coefficient for the dummy variable is given as follows: Holding other regressors constant, on average, the change in response attributed to the case where the dummy variable is 1, relative to the case where the dummy variable is 0, is given by the coefficient for the dummy variable.

Moving on, a regressor that has k categories can be represented by $k-1$ indicator variables:

x_1	x_2	...	x_{k-1}	Category
0	0	...	0	1
1	0	...	0	2
0	1	...	0	3
$\vdots$	$\vdots$	...	$\vdots$	$\vdots$
0	0	...	1	k

In this case, the category, or level, where all dummy variables are equal to 0 is the **reference category**. The reference category is the baseline we will compare all other categories to. You may want to choose this carefully. In this case, the interpretation of each of the $k-1$ coefficients is going to be as follows: Holding other regressors constant, on average, the change in response attributed to the case where the dummy variable corresponding to the coefficient is 1, relative to the reference category, is given by the coefficient for the dummy variable. Note that the reference category has no variable associated with it.

Example 6.2. When evaluating factors that affect the price of real estate, we may wish to consider location, while adjusting for lot size, year built and finished square feet. The data set `clean_data.csv` contains the prices of various types of real estate, as well as several important regressors. Regress the sale price on location, lot size, year built and finished square feet. Interpret the coefficient related to District 14. According to the model, holding other variables constant, what district has the highest priced properties? the lowest? Observe that District 7 has a non significant coefficient. In this case, what does it mean for District 7 to have a coefficient of 0?

```
####### Packages needed  ##############

library(lubridate)
```

```
Warning: package 'lubridate' was built under R version 4.2.3


Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
# Example: We would like to see how sale price of a home is related to
# various factors

####### Loading data  ##############

df_clean2=read.csv('clean_data.csv',stringsAsFactors = T)
df_clean2$District=as.factor(df_clean2$District)

# The first level is the reference category
attributes(df_clean2$District)$levels[1]
```

```
[1] "1"
```

```
####### Fitting the model  ##############
model=lm(Sale_price~District+Fin_sqft+Lotsize+Year_Built,df_clean2)

summ=summary(model); summ
```

```
Call:
lm(formula = Sale_price ~ District + Fin_sqft + Lotsize + Year_Built,
    data = df_clean2)

Residuals:
    Min      1Q  Median      3Q     Max
-399923  -25360     426   23383 1580056

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.170e+05  4.175e+04 -21.963  < 2e-16 ***
District2    1.335e+04  2.312e+03   5.772 7.92e-09 ***
District3    1.815e+05  2.462e+03  73.723  < 2e-16 ***
District4   -3.525e+04  4.777e+03  -7.378 1.66e-13 ***
District5    5.552e+04  1.988e+03  27.923  < 2e-16 ***
District6    1.202e+04  2.849e+03   4.218 2.47e-05 ***
District7   -4.300e+03  2.482e+03  -1.732  0.08327 .
District8    1.732e+04  2.708e+03   6.396 1.62e-10 ***
District9    2.810e+04  2.474e+03  11.361  < 2e-16 ***
District10   6.363e+04  2.073e+03  30.699  < 2e-16 ***
District11   7.032e+04  1.990e+03  35.333  < 2e-16 ***
District12   1.014e+04  3.240e+03   3.129  0.00175 **
District13   6.877e+04  2.057e+03  33.430  < 2e-16 ***
District14   1.026e+05  2.086e+03  49.212  < 2e-16 ***
District15  -3.375e+04  3.050e+03 -11.066  < 2e-16 ***
Fin_sqft     6.519e+01  6.525e-01  99.899  < 2e-16 ***
Lotsize      3.906e+00  1.228e-01  31.812  < 2e-16 ***
Year_Built   4.506e+02  2.153e+01  20.930  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 55680 on 24604 degrees of freedom
Multiple R-squared:  0.5889,    Adjusted R-squared:  0.5886
F-statistic:  2073 on 17 and 24604 DF,  p-value: < 2.2e-16
```

```r
coef(model)[which.max(coef(model))]
```

```
District3
 181495.9
```

```r
coef(model)[order(coef(model))[1:3]]
```

```
(Intercept)   District4  District15 
 -916960.88   -35245.83   -33748.17 
```

1. Interpret the coefficient for District 14: Holding lot size, finished square feet and year built constant, on average, the change in the price of a property in District 14, relative to District 1, is 102 600.
2. According to the model, holding other variables constant, what district has the highest priced properties on average? the lowest? The highest coefficient is District 3, and it is positive, so,holding other variables constant, on average District 3 has the highest priced properties. The lowest coefficient is District 4, and it is negative, so, holding other variables constant, on average District 4 has the lowest priced properties.
3. Observe that District 7 has a non significant coefficient. In this case, what does it mean for District 7 to have a coefficient of 0? This means that there is not enough evidence to show that District 1 and District 7 have different prices, holding other variables constant, on average.

> **Note**
>
> If all coefficients for the dummy variables are positive, then the reference category has the lowest average value of the response. Analogously, if all coefficients for the dummy variables are negative, then the reference category has the lowest average value of the response. Why? Use the interpretation of the coefficients to answer this question.

> **Note**
>
> ANOVA is Regression!- In one-way ANOVA, recall that we test for a difference in group means for a continuous response. We can represent the treatment groups with dummy variables and view this as a regression problem. That is, regressing the outcome on the dummy variables. It turns out that the regression ANOVA, that is, the overall F-test, applied to these dummy variables is equivalent to the one-way ANOVA (see Section 8.3 of the textbook.)

6.2 Interaction effects

An interaction effect occurs when the effect of one regressor on the response depends on the value of another regressor. In other words, the combined effect of two variables is not simply additive; the value of one variable modifies the impact of the other. In linear regression, this means that the coefficient of one regressor depends on the other.

We now give a simple example:

Suppose we are studying the effect of hours studied (X_1) and attendance (X_2) on exam scores (Y). An interaction effect between X_1 and X_2 would imply that the effect of studying on exam scores is different depending on the level of attendance. This can be modeled by including an interaction term ($X_1 \times X_2$) in the regression equation:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 (X_{1i} \times X_{2i}) + \epsilon_i.$$

The coefficient (_3) represents the interaction effect between X1 and X2. For instance, if $\beta_3 > 0$, then the more the student has attended the course, the more beneficial the student's hours studied will be.

Some examples of how interaction effects are applied in real life are given by:

- **Psychology**: Studying how different treatments and demographic factors interact to influence behavior.
- **Marketing**: Analyzing how different marketing strategies and customer demographics interact to affect sales.
- **Medicine**: Investigating how different drugs and patient characteristics interact to affect health outcomes.

Example 6.3. Let's recall Example 6.1 and suppose the researcher would like you to include the interaction effect between Weight and Sex. Explain how the regression line changes with a non-zero interaction effect. Interpret the estimated interaction effect. Is it significant?

The regression equation is

$$Y_i | X = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i1} X_{i2} + \epsilon_i.$$

If $\beta_3 \neq 0$ then we proceed as follows. When $X_{i2} = 0$, then the regression equation for males is still given by $Y_i | X = \beta_0 + \beta_1 X_{i1} + \epsilon_i$. Therefore, the expected body fat percentage for males is $\mathrm{E}[Y_i | X] = \beta_0 + \beta_1 X_{i1}$. Additionally, when $X_{i2} = 1$, then the regression equation for females is given by

$$Y_i | X = \beta_0 + \beta_1 X_{i1} + \beta_2 + \beta_3 X_{i1} + \epsilon_i = \beta_0 + (\beta_1 + \beta_3) X_{i1} + \beta_2 + \epsilon_i.$$

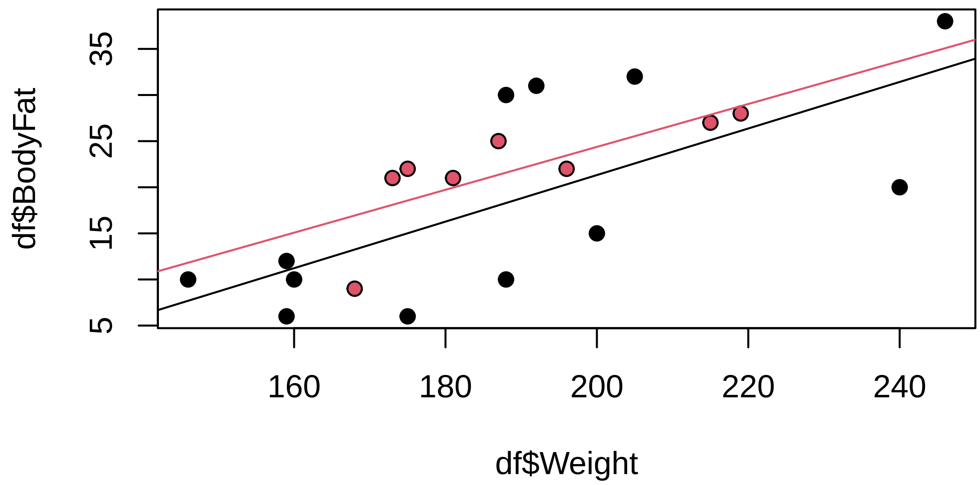
It follows that the expected body fat percentage for females is $\mathrm{E}[Y_i | X] = \beta_0 + (\beta_1 + \beta_3) X_{i1} + \beta_2$. Thus, adding an interaction effect allows the model to generate a completely different regression line, that is a different slope **and** intercept for females. The expected body fat percentage for females is then $\beta_2 + \beta_3 X_{i1}$ higher than for males, holding weight constant. Adding an interaction effect allows the slope to also vary, depending on whether the subject is male or female.

Let's observe.

```r
# Make the data frame
Weight=c(175 , 181 , 200 , 159 , 196 , 192 , 205 , 173 , 187 , 188 ,
         188 , 240 , 175 , 168 , 246 , 160 , 215 , 159 , 146 , 219 )
BodyFat =c(6 , 21 , 15 , 6 , 22 , 31 , 32 ,21 , 25 , 30 ,
           10 , 20 , 22 , 9 , 38 , 10 , 27 , 12 , 10 , 28 )
Sex=c("F","M","F","F","M","F","F","M","M","F","F","F","M","M","F","F","M","F","F","M")

df=data.frame(Weight=Weight,BodyFat=BodyFat,Sex=Sex,stringsAsFactors = T)

df$Sex=relevel(df$Sex,"M")

mod=lm(BodyFat~Weight+Sex+Weight*Sex,data=df)
summary(mod)
```

```
Call:
lm(formula = BodyFat ~ Weight + Sex + Weight * Sex, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-11.4171  -5.3084   0.1178   3.4912  11.7087

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -22.13450   27.12486  -0.816    0.426
Weight        0.23255    0.14269   1.630    0.123
SexF         -7.02921   30.18090  -0.233    0.819
Weight:SexF   0.01987    0.15869   0.125    0.902

Residual standard error: 7.255 on 16 degrees of freedom
Multiple R-squared:  0.5153,    Adjusted R-squared:  0.4245
F-statistic: 5.671 on 3 and 16 DF,  p-value: 0.007662
```

```r
plot(df$Weight,df$BodyFat,bg=((df$Sex=="M")+1),pch=21)
abline(coef(mod)[1],coef(mod)[2],col=2)
abline(coef(mod)[1]+coef(mod)[3],coef(mod)[2]+coef(mod)[4],col=1)
```



Notice how the slopes of the regression lines differ! Now, the estimated interaction effect is `0.01987`. Let's interpret it. We have that, on average, for every one lb increase in weight, the body fat percentage of a female increases by 0.01987 more than that of a male.

(In this case, the term is not significant, so we would probably drop it.)

> **Note**
>
> We can include interaction effects in the regression model in R by adding `variable_1*variable_2` to the right-hand side of the formula equation.

Example 6.4. When evaluating factors that affect the price of real estate, we may wish to consider location, while adjusting for lot size, year built and finished square feet. The data set `clean_data.csv` contains the prices of various types of real estate, as well as several important regressors. Regress the sale price on location, lot size, year built and finished square feet. Add an interaction term between year built and location. Interpret the interaction term for District 14.

```
####### Fitting the model  ##############
model=lm(Sale_price~District+Fin_sqft+Lotsize+Year_Built+Year_Built*District,df_clean2)

summ=summary(model); summ
```

```
Call:
lm(formula = Sale_price ~ District + Fin_sqft + Lotsize + Year_Built +
    Year_Built * District, data = df_clean2)

Residuals:
    Min      1Q  Median      3Q     Max
-400250  -24589     441   23005 1569420

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.567e+06  2.215e+05  -7.077 1.52e-12 ***
District2              9.848e+05  3.594e+05   2.740 0.006148 **
District3             -2.929e+05  2.723e+05  -1.076 0.282044
District4              6.838e+05  3.573e+05   1.914 0.055676 .
District5              1.551e+06  2.634e+05   5.888 3.97e-09 ***
District6              5.445e+05  2.683e+05   2.029 0.042446 *
District7             -7.242e+05  3.234e+05  -2.239 0.025139 *
District8              8.463e+05  3.000e+05   2.821 0.004790 **
District9             -5.339e+04  3.041e+05  -0.176 0.860634
District10             8.690e+05  2.602e+05   3.339 0.000842 ***
District11             1.899e+05  2.660e+05   0.714 0.475313
District12             1.282e+06  2.994e+05   4.283 1.85e-05 ***
District13             6.296e+05  2.557e+05   2.463 0.013799 *
District14             1.502e+06  2.392e+05   6.278 3.49e-10 ***
District15             5.815e+04  2.610e+05   0.223 0.823727
Fin_sqft               6.497e+01  6.659e-01  97.556  < 2e-16 ***
Lotsize                3.989e+00  1.237e-01  32.256  < 2e-16 ***
Year_Built             7.850e+02  1.139e+02   6.891 5.68e-12 ***
District2:Year_Built  -4.986e+02  1.841e+02  -2.708 0.006770 **
District3:Year_Built   2.547e+02  1.409e+02   1.807 0.070772 .
District4:Year_Built  -3.703e+02  1.858e+02  -1.993 0.046287 *
District5:Year_Built  -7.666e+02  1.352e+02  -5.668 1.46e-08 ***
District6:Year_Built  -2.727e+02  1.388e+02  -1.965 0.049443 *
District7:Year_Built   3.734e+02  1.667e+02   2.240 0.025118 *
District8:Year_Built  -4.278e+02  1.555e+02  -2.751 0.005938 **
District9:Year_Built   3.721e+01  1.555e+02   0.239 0.810891
District10:Year_Built -4.146e+02  1.340e+02  -3.093 0.001985 **
District11:Year_Built -6.285e+01  1.366e+02  -0.460 0.645462
District12:Year_Built -6.608e+02  1.555e+02  -4.251 2.14e-05 ***
District13:Year_Built -2.887e+02  1.314e+02  -2.198 0.027981 *
District14:Year_Built -7.231e+02  1.232e+02  -5.869 4.43e-09 ***
District15:Year_Built -4.403e+01  1.347e+02  -0.327 0.743706
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 55410 on 24590 degrees of freedom
Multiple R-squared:  0.5931,	Adjusted R-squared:  0.5925
F-statistic:  1156 on 31 and 24590 DF,  p-value: < 2.2e-16
```

Observe that the interaction term between year built and District 14 is -723.1$. In addition, note that year built has a positive coefficient of 785$. We can interpret the interaction effect as follows: Holding finished square feet and lot size constant, a one year increase in year built for a home in District 14 results in an increase in price that is 723.1 lower than that of District 1. We can also reword this to make it a little more clear - Holding finished square feet and lot size constant, a one year increase in year built for a home in District 1 results in an increase in price that is 723.1 higher than that of District 1.

To see this observe that for a one unit increase in year built in District 14, we have that the price goes up by 785.0-723.1=61.9 on average, holding other variables constant. On the other hand, in District 1, the price goes up by 785.0 on average, holding other variables constant. Therefore, in general, newer homes are much more valuable in District 1.

Exercise 6.1. Interpret the main effects and the interaction effect with year built for Districts 2-4. (The main effects are the coefficients for Districts 2-4.)

> ⚠ Warning
>
> The interpretation for interaction effects is difficult and nuanced. Make sure you study this topic carefully.

6.3 Increasing codes and quantitative regressors via dummy variables

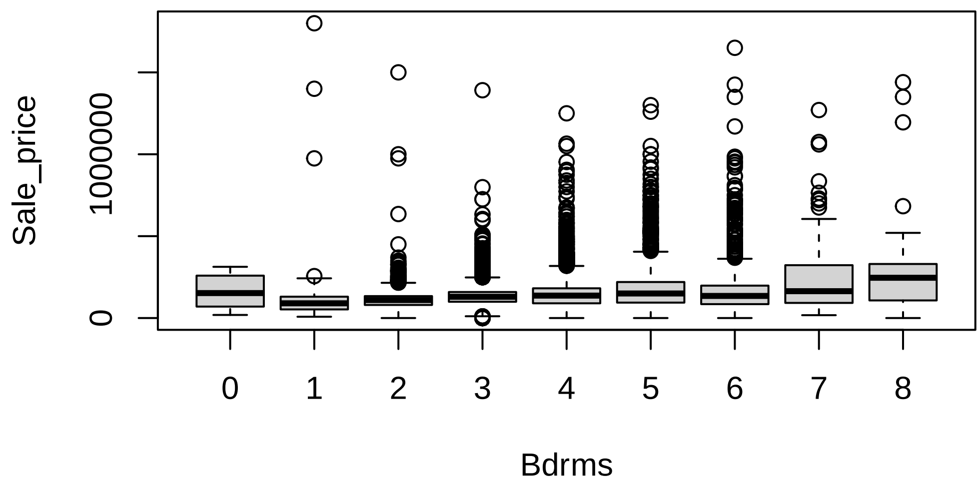
Another approach to the treatment of a qualitative variable in regression is to measure the levels of the variable by an allocated code. Suppose we model the effect of the number of bedrooms on real estate price by its numerical value, instead of categorical value. Let's see what happens to the regression equation. In general, ordinal variables may be better represented by dummy variables/indicators - however, dummy variables increase the complexity of the model, which we may not have enough data to support, and could lead to overfitting.

Example 6.5. When evaluating factors that affect the price of real estate, we may wish to consider the unadjusted effect of the number of bedrooms. Regress the sale price on number of bedrooms treating number of bedrooms as a continuous variable. Then, regress the sale price on number of bedrooms treating number of bedrooms as a categorical variable. Compare and contrast the two models, and the resulting fits.

```
####### Fitting the model  ##############
unique(df_clean2$Bdrms)
```

```
 [1] >8 2  0  4  7  3  1  6  8  5
Levels: >8 0 1 2 3 4 5 6 7 8
```

```
# Drop these rows
df_clean3=df_clean2[df_clean2$Bdrms!='>8',]
df_clean3$Bdrms=droplevels(df_clean3$Bdrms)
df_clean3$Bdrms=relevel(df_clean3$Bdrms,"0")
# Add new continuous variable
df_clean3$Bdrms2=as.numeric(df_clean3$Bdrms)-1

boxplot(Sale_price~Bdrms,df_clean3)
```



```r
model_ca=lm(Sale_price~Bdrms,df_clean3)
model_co=lm(Sale_price~Bdrms2,df_clean3)

summ=summary(model_ca); summ
```

```
Call:
lm(formula = Sale_price ~ Bdrms, data = df_clean3)

Residuals:
    Min      1Q  Median      3Q     Max 
-271312  -43342   -6342   26658 1672084 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   161825      29764   5.437 5.47e-08 ***
Bdrms1        -33909      30818  -1.100 0.271216    
Bdrms2        -53113      29800  -1.782 0.074712 .  
Bdrms3        -28483      29774  -0.957 0.338758    
Bdrms4        -15157      29785  -0.509 0.610830    
Bdrms5         24156      29852   0.809 0.418414    
Bdrms6          6826      29861   0.229 0.819189    
Bdrms7         81735      30717   2.661 0.007798 ** 
Bdrms8        109487      31332   3.494 0.000476 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84190 on 24586 degrees of freedom
Multiple R-squared:  0.05675,	Adjusted R-squared:  0.05644 
F-statistic: 184.9 on 8 and 24586 DF,  p-value: < 2.2e-16
```

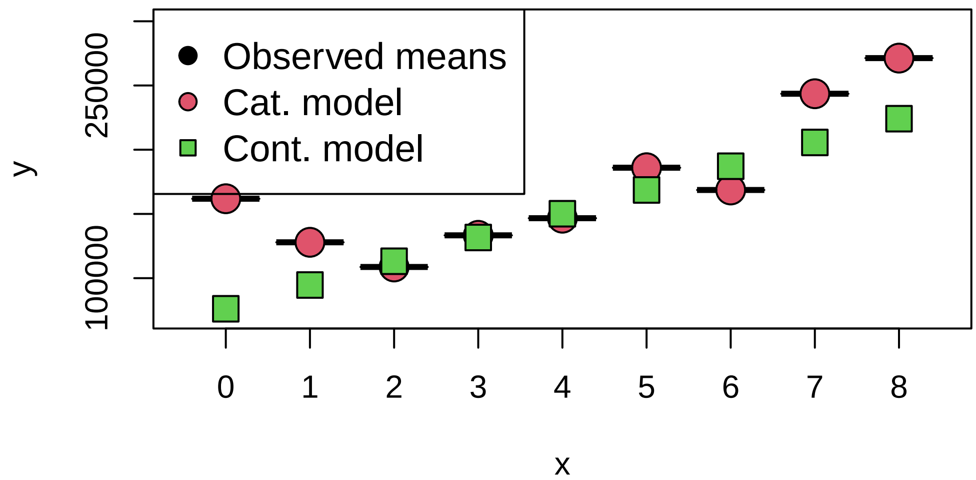
```r
summ=summary(model_co); summ
```

```
Call:
lm(formula = Sale_price ~ Bdrms2, data = df_clean3)

Residuals:
    Min      1Q  Median      3Q     Max 
-224144  -43153   -6651   26849 1705343 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  76159.0     1849.3   41.18   <2e-16 ***
Bdrms2       18498.1      522.4   35.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84540 on 24593 degrees of freedom
Multiple R-squared:  0.04851,   Adjusted R-squared:  0.04847
F-statistic:  1254 on 1 and 24593 DF,  p-value: < 2.2e-16
```

Observe that the continuous model says that for every additional bedroom, on average the price increases by 18 498$. On the other hand, the categorical model says that the change in price depends on the number of bedrooms. For example, going from 0 bedrooms to 1 bedroom, we actually see a reduction in price of -33 909$. Let's graph the expected price for each number of bedrooms from both models:

```r
####### Fitting the model  ##############
new_dat=data.frame('Bdrms'=sort(unique(df_clean3$Bdrms)))
new_dat2=data.frame('Bdrms2'=0:8)
# predict(model_ca,new_dat)
# predict(model_co,new_dat2)

observed_means=aggregate(Sale_price~Bdrms,data=df_clean3, FUN = "mean")
plot(observed_means[,1],observed_means[,2],pch=25,bg=1,cex=3,ylim=c( 70000,300000))
points(x=observed_means[,1],predict(model_ca,new_dat),pch=21,bg=2,cex=2)
points(x=observed_means[,1],predict(model_co,new_dat2),pch=22,bg=3,cex=2)
legend("topleft",legend=c("Observed means","Cat. model","Cont. model"),pch=c(21,21,22),pt.
```



Observe that the continuous model does not match the data at all, while the categorical model is able to model the **non-linear** relationship between the number of bedrooms and the sale price! Why is this the case? Treating a regressor as continuous implies that there is a linear relationship between that regressor and the response. On the other hand, modelling the variable with indicators does not place any assumption on the relationship between the regressor and the response. The drawback, is that we need 7 more parameters in the model.

> **Warning**
>
> When deciding to treat continuous or ordinal variables as continuous, it is critical that you evaluate whether it is acceptable to assume a linear relationship between the regression and the response. If you cannot verify this assumption, or it seems invalid, it is best to treat the regressor as categorical.

Quantitative regressors can also be represented by indicator variables. Sometimes this is necessary because it is difficult to collect accurate information on the quantitative regressor, or the exact values are obscured for privacy reasons. Treating a quantitative factor as a qualitative one increases the complexity of the model. This approach also reduces the degrees of freedom for error. However, the indicator variable approach does not require the analyst to make any prior assumptions about the functional form of the relationship between the response and the regressor variable, as previously discussed.

6.4 A larger scale example

It is a good time to stop introducing new material and do a larger scale example.

Example 6.6. Explore the pricing data, and evaluate what factors influence the price of a property. Be sure to assess the fit of the model and check assumptions.

```
####### Packages needed  ##############

library(lubridate)

# Example: We would like to see how sale price of a home is related to
# various factors

####### Loading data  ##############

df_clean2=read.csv('clean_data.csv',stringsAsFactors = T)

####### Analyzing the data via EDA  ##############

names(df_clean2)
```

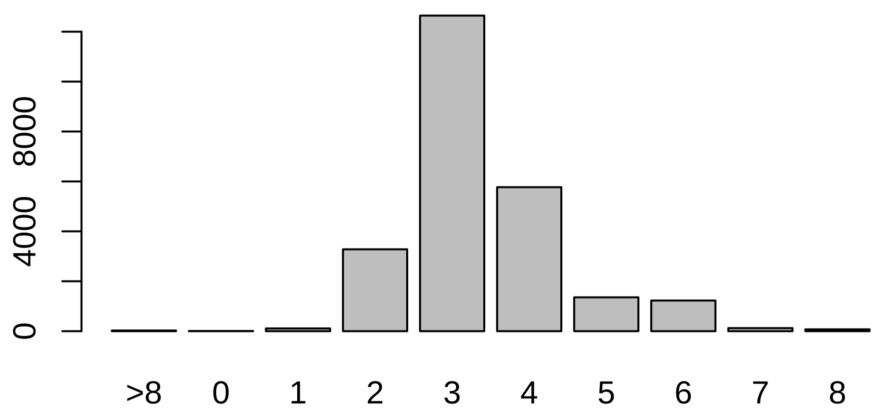
```
 [1] "District"   "Extwall"    "Stories"    "Year_Built" "Fin_sqft"
 [6] "Units"      "Bdrms"      "Fbath"      "Lotsize"    "Sale_date"
[11] "Sale_price"
```

```
head(df_clean2)
```

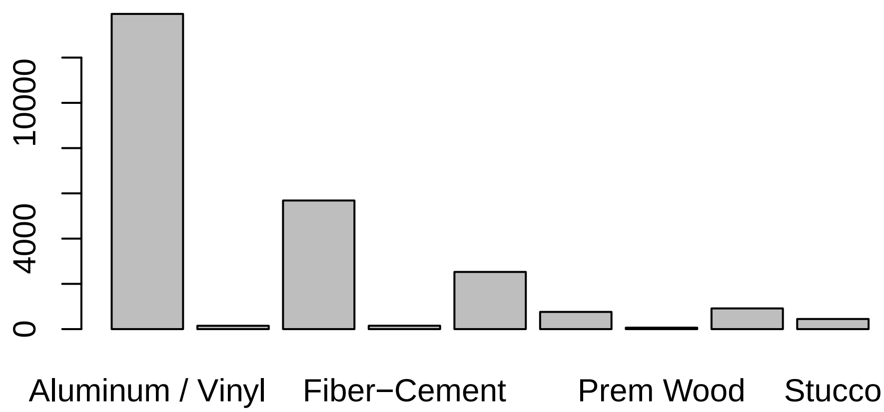
```
  District Extwall Stories Year_Built Fin_sqft Units Bdrms Fbath Lotsize
1        7   Frame       2       1913     3476    >3    >8     1    5040
2        3   Frame       2       1897     1992    >3     2     2    2880
3        4   Frame       2       1907     2339    >3     0     1    3185
4        4   Frame       2       1890     2329    >3     4     1    5781
5        4   Stone      >2       1891     7450     2     7    >4   15600
6       12   Frame     1.5       1906     2462     2     3     2    5075
  Sale_date Sale_price
1     11719      42000
2     11808     145000
3     11839      30000
4     11961      66500
```

```
5      11992      150500
6      11992       75000
```

```r
barplot(table(df_clean2$Bdrms))
```




```r
barplot(table(df_clean2$Extwall))
barplot(table(df_clean2$Extwall))
```




```r
dim(df_clean2)
```

```
[1] 24622    11
```

```r
####### Fitting the model  ##############
model=lm(Sale_price~.,df_clean2)

summ=summary(model); summ
```

```
Call:
lm(formula = Sale_price ~ ., data = df_clean2)

Residuals:
    Min      1Q  Median      3Q     Max 
-638318  -30032    1735   29252 1621794 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -3.674e+05  4.580e+04  -8.023 1.08e-15 ***
```

```
District               1.854e+03  9.619e+01  19.271  < 2e-16 ***
ExtwallBlock           4.751e+03  5.104e+03   0.931   0.3520
ExtwallBrick           7.562e+03  1.011e+03   7.480 7.67e-14 ***
ExtwallFiber-Cement    1.096e+04  5.220e+03   2.101   0.0357 *
ExtwallFrame          -6.693e+03  1.357e+03  -4.934 8.13e-07 ***
ExtwallMasonry / Frame 1.391e+02  2.385e+03   0.058   0.9535
ExtwallPrem Wood      -1.128e+03  7.818e+03  -0.144   0.8853
ExtwallStone           1.679e+02  2.136e+03   0.079   0.9374
ExtwallStucco          1.389e+04  2.990e+03   4.646 3.40e-06 ***
Stories1               7.364e+04  1.436e+04   5.127 2.96e-07 ***
Stories1.5             8.066e+04  1.434e+04   5.624 1.89e-08 ***
Stories2               8.842e+04  1.429e+04   6.188 6.17e-10 ***
Year_Built             1.244e+01  1.973e+01   0.631   0.5283
Fin_sqft               1.000e+02  1.313e+00  76.166  < 2e-16 ***
Units1                 1.335e+05  1.014e+04  13.169  < 2e-16 ***
Units2                 4.277e+04  1.014e+04   4.219 2.47e-05 ***
Units3                -1.593e+04  1.092e+04  -1.458   0.1447
Bdrms0                 1.731e+05  2.578e+04   6.714 1.93e-11 ***
Bdrms1                 2.070e+05  1.385e+04  14.948  < 2e-16 ***
Bdrms2                 1.647e+05  1.259e+04  13.082  < 2e-16 ***
Bdrms3                 1.575e+05  1.251e+04  12.588  < 2e-16 ***
Bdrms4                 1.397e+05  1.246e+04  11.208  < 2e-16 ***
Bdrms5                 1.357e+05  1.246e+04  10.890  < 2e-16 ***
Bdrms6                 1.166e+05  1.248e+04   9.341  < 2e-16 ***
Bdrms7                 8.164e+04  1.331e+04   6.133 8.75e-10 ***
Bdrms8                 1.006e+05  1.398e+04   7.195 6.43e-13 ***
Fbath0                -1.564e+05  1.841e+04  -8.494  < 2e-16 ***
Fbath1                -1.346e+05  1.327e+04 -10.142  < 2e-16 ***
Fbath2                -1.191e+05  1.318e+04  -9.031  < 2e-16 ***
Fbath3                -6.609e+04  1.310e+04  -5.045 4.57e-07 ***
Fbath4                 5.041e+03  1.405e+04   0.359   0.7197
Lotsize                1.947e+00  1.349e-01  14.429  < 2e-16 ***
Sale_date              5.089e+00  3.635e-01  13.999  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61440 on 24588 degrees of freedom
Multiple R-squared:  0.4997,    Adjusted R-squared:  0.499
F-statistic: 744.1 on 33 and 24588 DF,  p-value: < 2.2e-16
```

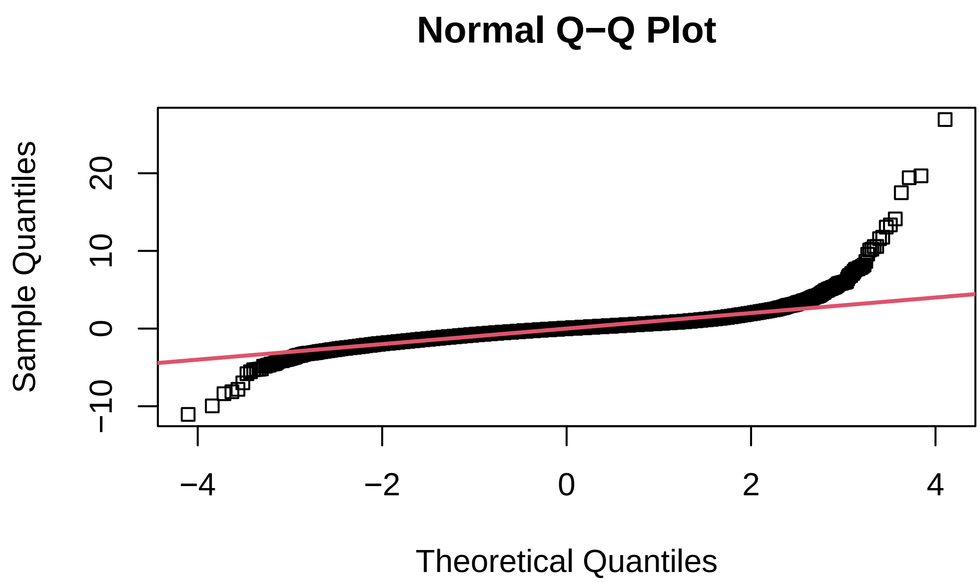
```
####### Now, let's interpret the output ##############
summ$r.squared
```

```
[1] 0.4996725
```

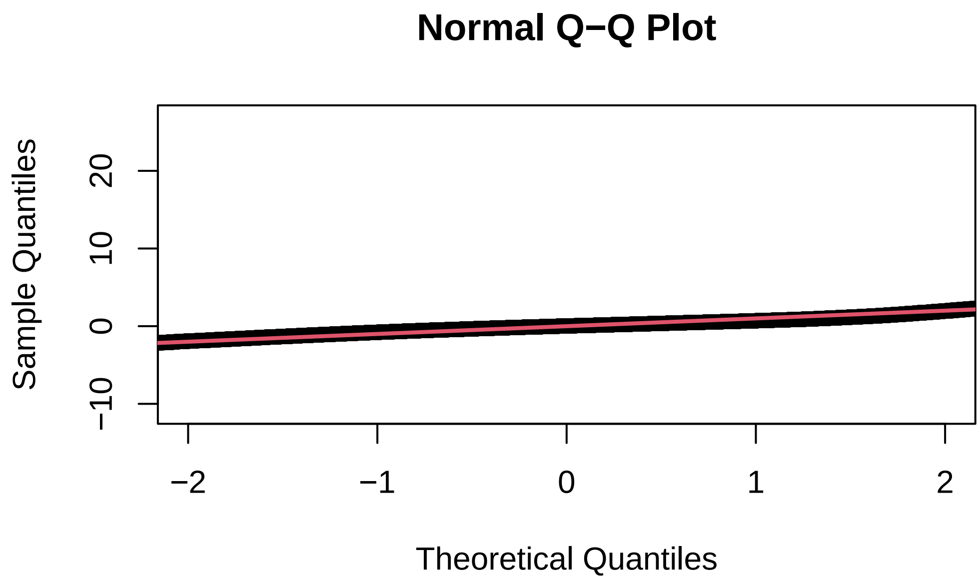
```
summ$adj.r.squared
```

```
[1] 0.499001
```

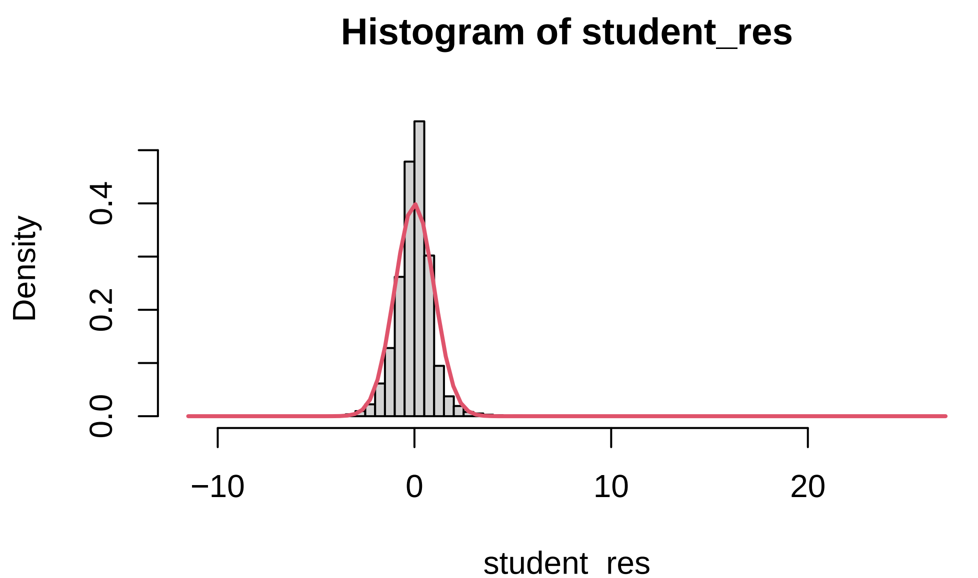
```
####### Residual analysis ##############
student_res=rstudent(model)
MSE=summ$sigma^2


qqnorm(student_res,pch=22)
abline(0,1,col=2,lwd=2)
```



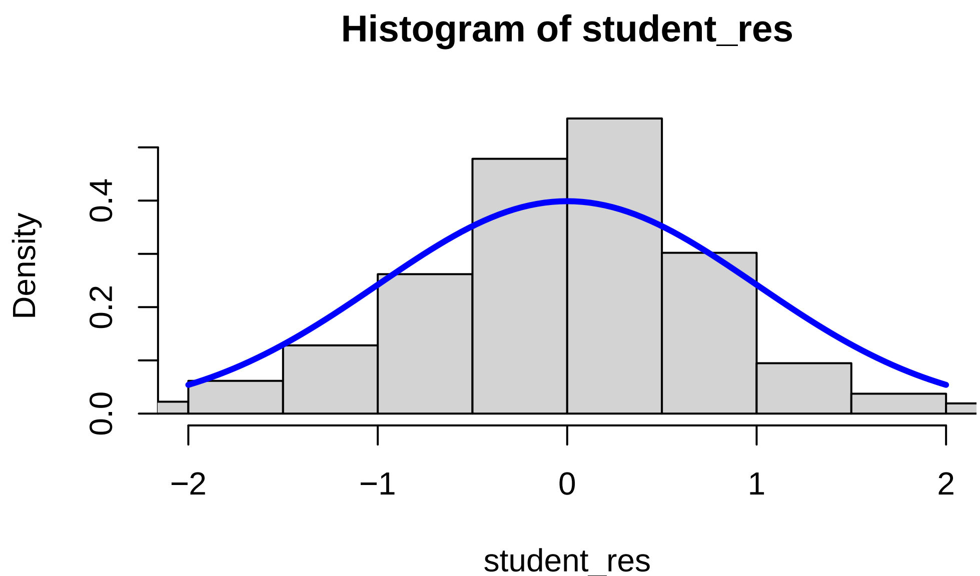

```
qqnorm(student_res,pch=22,xlim=c(-2,2))
abline(0,1,col=2,lwd=2)
```



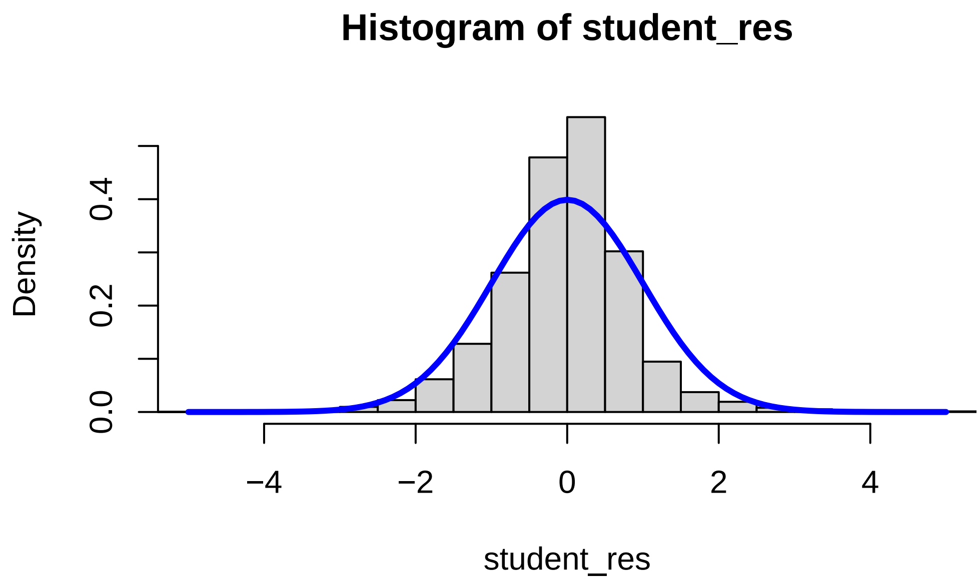

```
hist(student_res,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T,col=2,lwd=2)
```



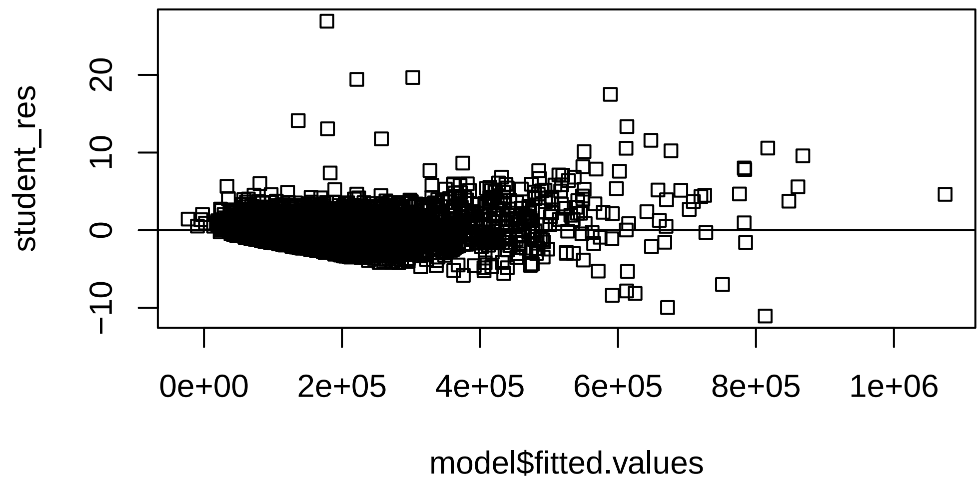
```r
hist(student_res,freq=F,breaks=100,xlim=c(-2,2))
curve(dnorm(x,0,1),add=T,col='blue',lwd=3)
```



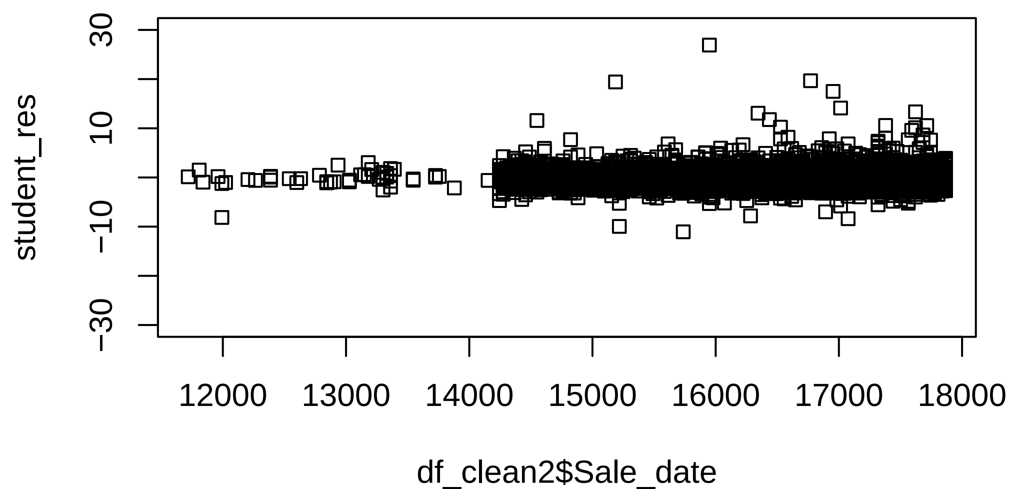

```r
hist(student_res,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T,col='blue',lwd=3)
```



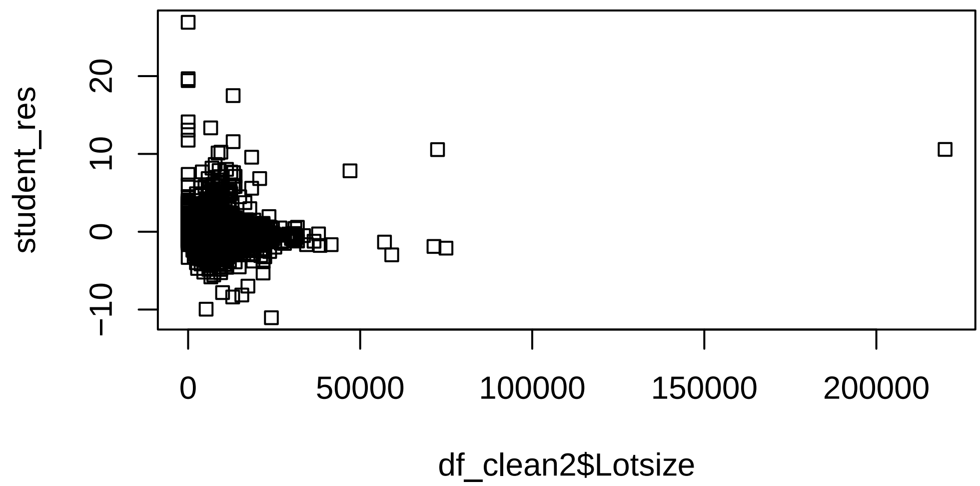

```
# SOS
plot(model$fitted.values,student_res,pch=22)
abline(h=0)
```



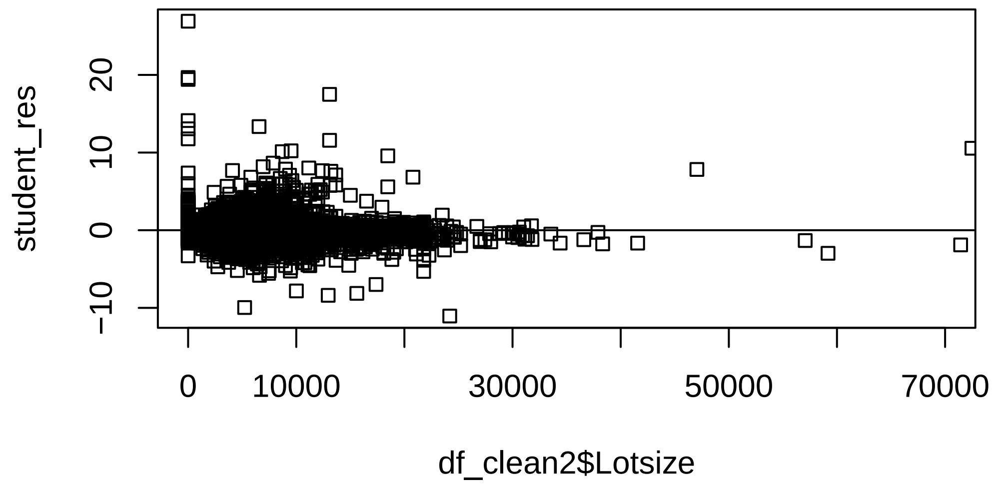

```r
plot(df_clean2$Sale_date ,student_res,pch=22,ylim=c(-30,30))
```



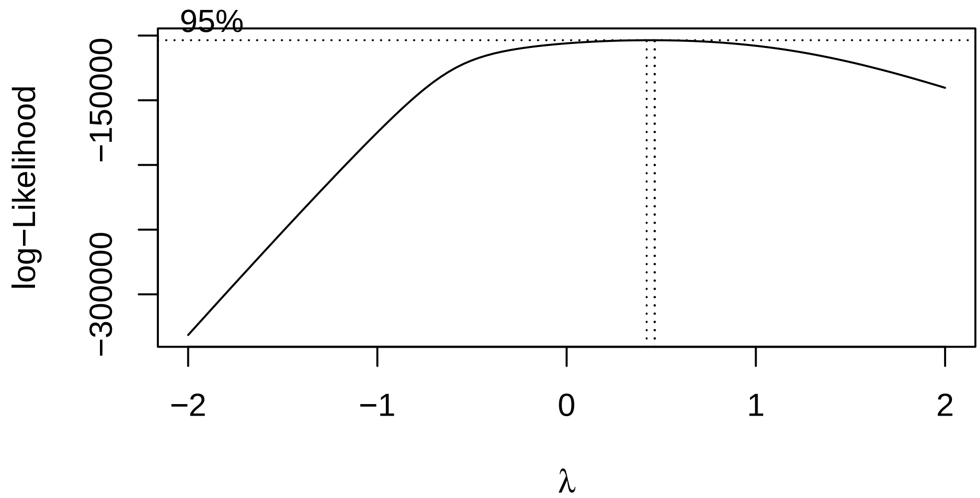
```r
plot(df_clean2$Lotsize ,student_res,pch=22)
```



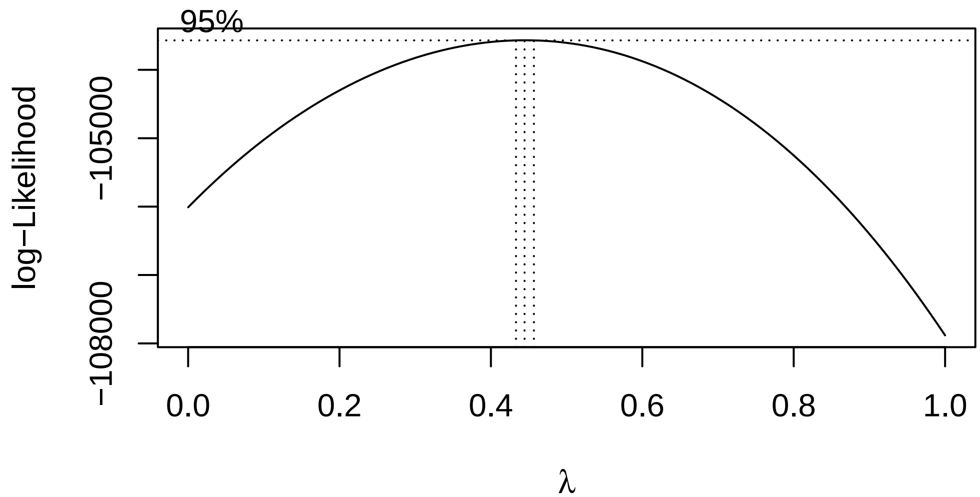

```r
plot(df_clean2$Lotsize ,student_res,pch=22,xlim=c(0,70000))
abline(h=0)
```




```
# Let's try a transformation

# model=lm(Sale_price~.,df_clean2)
df_clean2=df_clean2[df_clean2$Sale_price>0,]

bc=MASS::boxcox(Sale_price~.,data=df_clean2,
                lambda = seq(-2, 2, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
```




```
df_clean3=df_clean2[df_clean2$Sale_price>0,]
# bc=MASS::boxcox(Sale_price~.,data=df_clean3,
                # lambda = seq(-2, 2, 1/10),
                # plotit = TRUE,
                # eps = 1/50,
                # xlab = expression(lambda),
                # ylab = "log-Likelihood")


bc=MASS::boxcox(Sale_price~.,data=df_clean3,
                lambda = seq(0, 1, 1/10),
                plotit = TRUE,
                eps = 1/50,
                xlab = expression(lambda),
                ylab = "log-Likelihood")
```




```r
model2=lm(sqrt(Sale_price)~.,df_clean3)
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

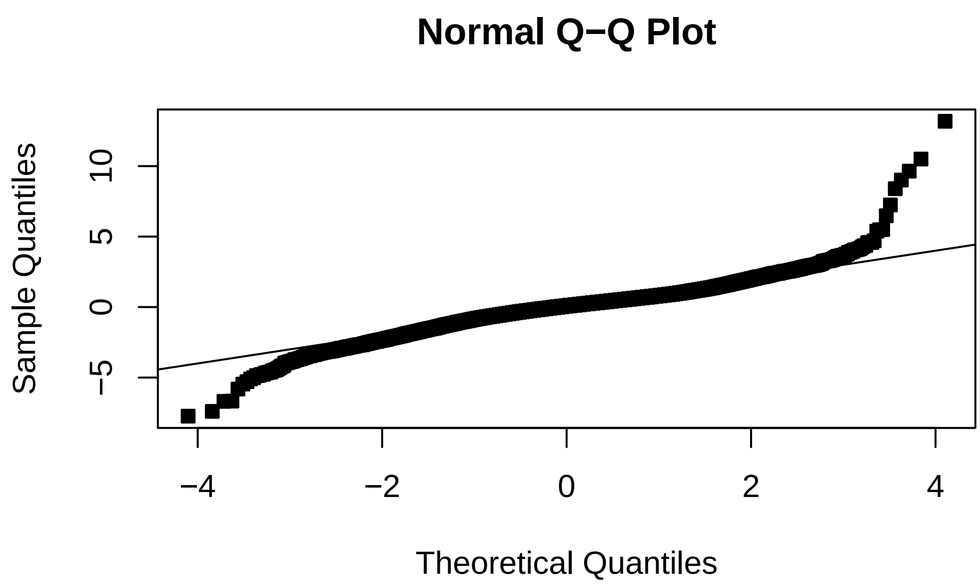
```
Call:
lm(formula = sqrt(Sale_price) ~ ., data = df_clean3)

Residuals:
    Min      1Q  Median      3Q     Max 
-542.68  -37.46    6.16   42.34  973.57 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -7.741e+02  5.556e+01 -13.932  < 2e-16 ***
District                3.607e+00  1.167e-01  30.915  < 2e-16 ***
ExtwallBlock            4.563e-01  6.189e+00   0.074 0.941222    
ExtwallBrick            1.310e+01  1.226e+00  10.684  < 2e-16 ***
ExtwallFiber-Cement     1.593e+01  6.329e+00   2.518 0.011815 *  
ExtwallFrame           -9.554e+00  1.646e+00  -5.806 6.48e-09 ***
```

```
ExtwallMasonry / Frame  7.916e+00  2.896e+00   2.734 0.006268 **
ExtwallPrem Wood        1.176e+01  9.479e+00   1.240 0.214945
ExtwallStone            8.667e+00  2.591e+00   3.345 0.000823 ***
ExtwallStucco           1.884e+01  3.626e+00   5.195 2.07e-07 ***
Stories1                4.819e+01  1.741e+01   2.767 0.005660 **
Stories1.5              6.198e+01  1.739e+01   3.564 0.000366 ***
Stories2                7.132e+01  1.732e+01   4.117 3.86e-05 ***
Year_Built              2.876e-01  2.394e-02  12.011  < 2e-16 ***
Fin_sqft                1.009e-01  1.593e-03  63.301  < 2e-16 ***
Units1                  1.272e+02  1.229e+01  10.348  < 2e-16 ***
Units2                  2.193e+01  1.229e+01   1.784 0.074427 .
Units3                 -2.377e+01  1.324e+01  -1.795 0.072732 .
Bdrms0                  1.537e+02  3.125e+01   4.917 8.85e-07 ***
Bdrms1                  1.697e+02  1.679e+01  10.109  < 2e-16 ***
Bdrms2                  1.460e+02  1.526e+01   9.566  < 2e-16 ***
Bdrms3                  1.481e+02  1.517e+01   9.764  < 2e-16 ***
Bdrms4                  1.281e+02  1.511e+01   8.480  < 2e-16 ***
Bdrms5                  1.245e+02  1.511e+01   8.244  < 2e-16 ***
Bdrms6                  1.025e+02  1.513e+01   6.776 1.26e-11 ***
Bdrms7                  7.174e+01  1.614e+01   4.445 8.84e-06 ***
Bdrms8                  1.004e+02  1.698e+01   5.917 3.33e-09 ***
Fbath0                 -6.028e+01  2.232e+01  -2.700 0.006930 **
Fbath1                 -3.910e+01  1.609e+01  -2.430 0.015100 *
Fbath2                 -1.371e+01  1.599e+01  -0.858 0.391107
Fbath3                  3.069e+01  1.589e+01   1.932 0.053401 .
Fbath4                  6.407e+01  1.703e+01   3.762 0.000169 ***
Lotsize                 1.851e-03  1.636e-04  11.317  < 2e-16 ***
Sale_date               6.306e-03  4.411e-04  14.295  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 74.49 on 24559 degrees of freedom
Multiple R-squared:  0.436, Adjusted R-squared:  0.4353
F-statistic: 575.4 on 33 and 24559 DF,  p-value: < 2.2e-16
```

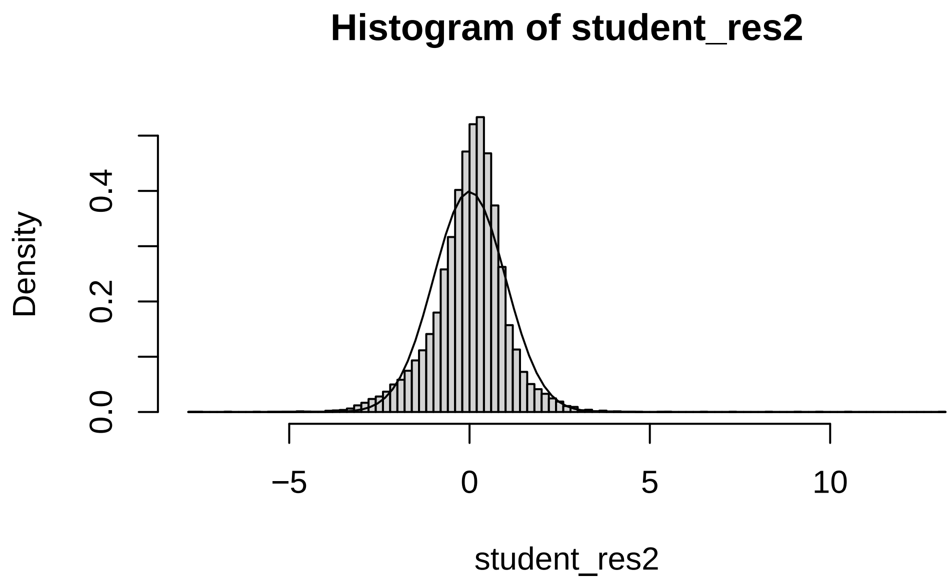
```
summ2$adj.r.squared
```

```
[1] 0.4352901
```

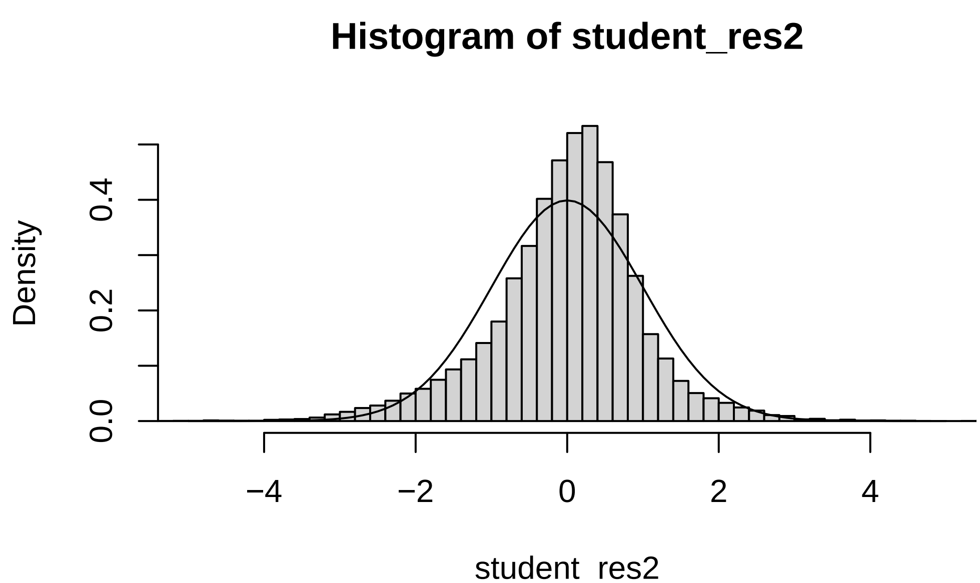
```
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



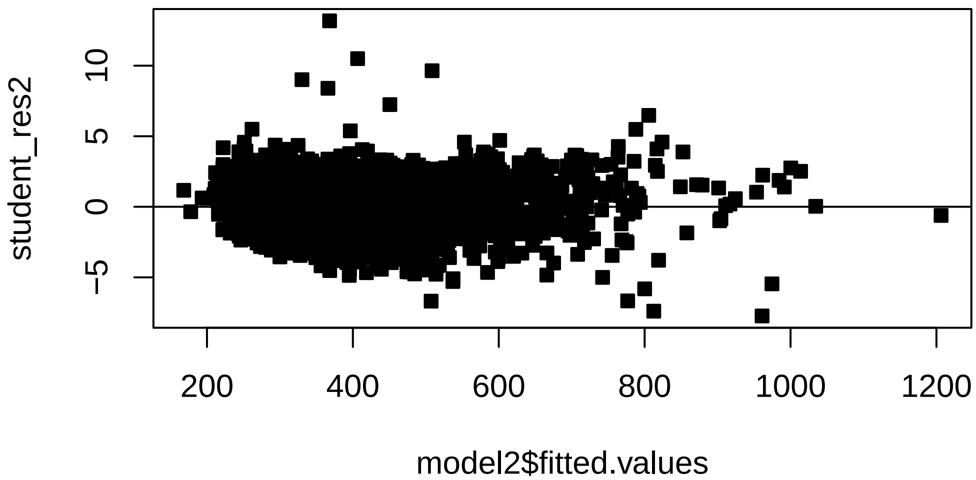

```
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



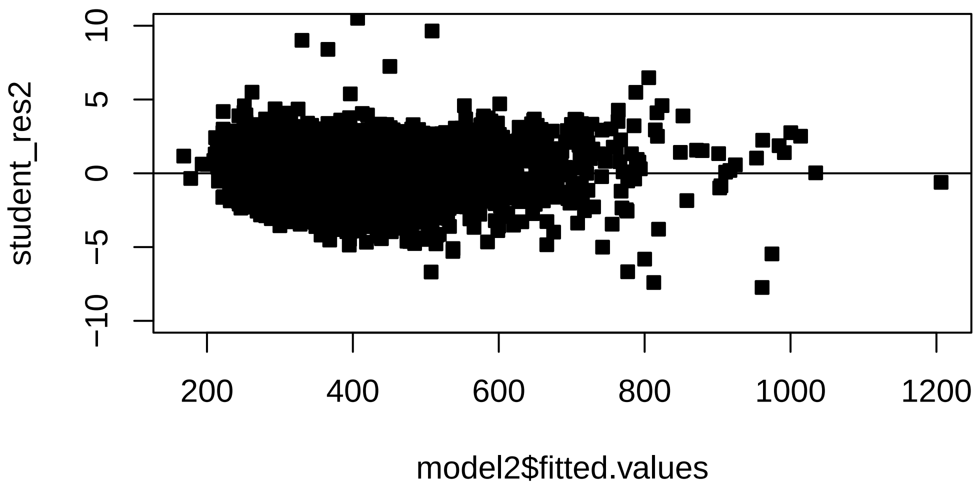

```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



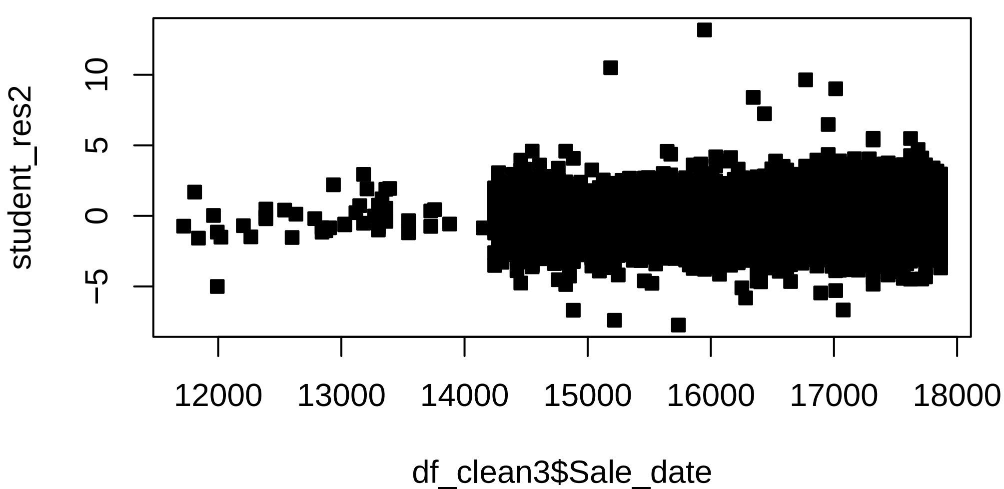
```r
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



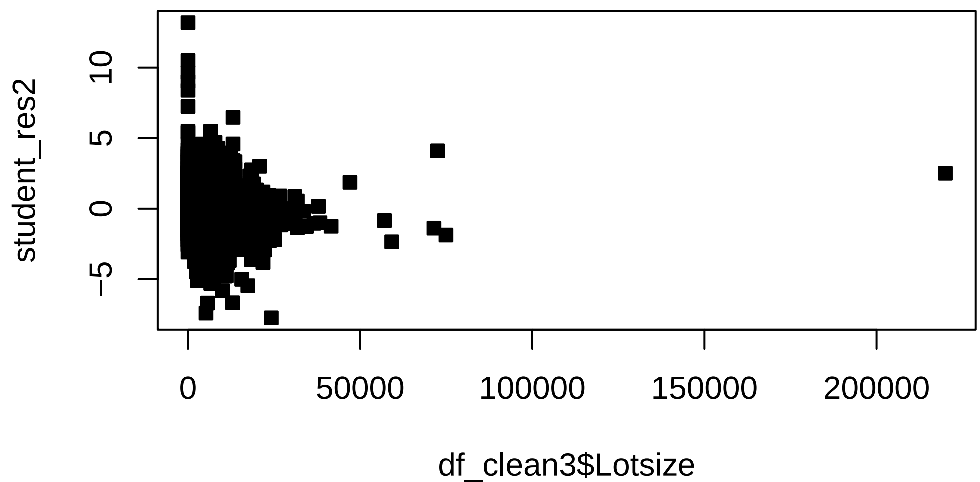

```r
plot(model2$fitted.values,student_res2,pch=22,bg=1,ylim=c(-10,10))
abline(h=0)
```



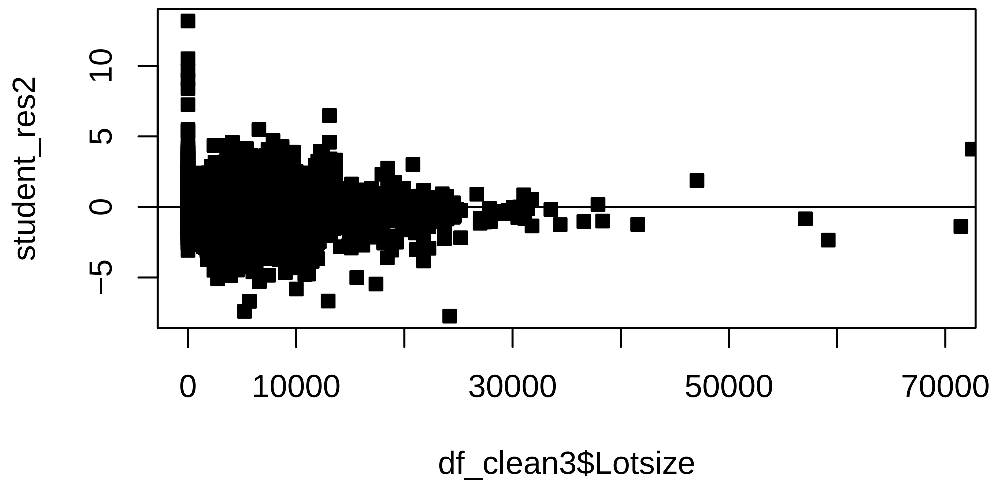

```
plot(df_clean3$Sale_date ,student_res2,pch=22,bg=1)
```



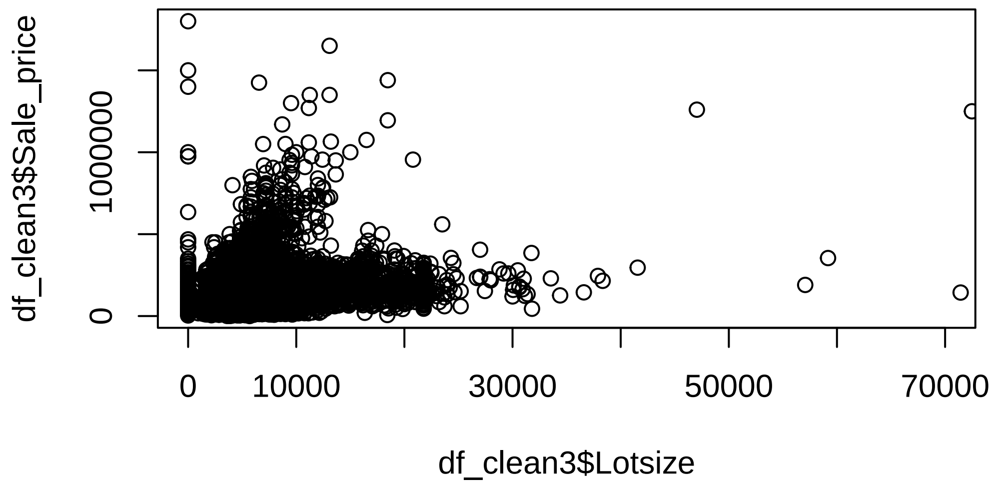
```r
plot(df_clean3$Lotsize ,student_res2,pch=22,bg=1)
```




```r
plot(df_clean3$Lotsize ,student_res2,pch=22,bg=1,xlim=c(0,70000))
abline(h=0)
```




```
# It feels like the slope of lot size
# depends on something,
#like two categories
plot(df_clean3$Lotsize,df_clean3$Sale_price,xlim=c(0,70000))
```




```
# plot(1/df_clean3$Lotsize,df_clean3$Sale_price)

sum(df_clean3$Lotsize==0)
```

```
[1] 146
```

```
df_clean4=df_clean3[df_clean3$Lotsize!=0,]

model2=lm(sqrt(Sale_price)~.,df_clean4)
# model2=lm(Sale_price^(0.4)~.,df_clean3)

summ2=summary(model2); summ2
```

```
Call:
lm(formula = sqrt(Sale_price) ~ ., data = df_clean4)
```

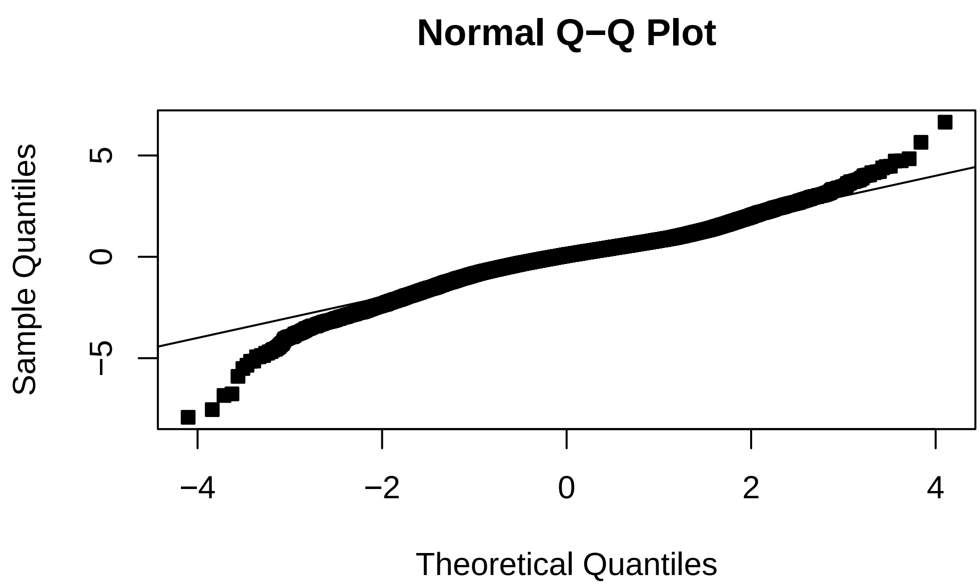
```
Residuals:
    Min      1Q  Median      3Q     Max 
-544.58  -36.99    6.27   42.23  481.64 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -7.754e+02  5.483e+01 -14.142  < 2e-16 ***
District                3.744e+00  1.148e-01  32.609  < 2e-16 ***
ExtwallBlock           -3.526e+00  6.132e+00  -0.575 0.565249    
ExtwallBrick            1.266e+01  1.204e+00  10.511  < 2e-16 ***
ExtwallFiber-Cement     1.604e+01  6.210e+00   2.584 0.009776 ** 
ExtwallFrame           -9.920e+00  1.627e+00  -6.095 1.11e-09 ***
ExtwallMasonry / Frame  8.766e+00  2.845e+00   3.082 0.002061 ** 
ExtwallPrem Wood        1.158e+01  9.299e+00   1.245 0.213087    
ExtwallStone            8.975e+00  2.542e+00   3.530 0.000416 ***
ExtwallStucco           1.876e+01  3.573e+00   5.251 1.52e-07 ***
Stories1                4.698e+01  1.708e+01   2.750 0.005963 ** 
Stories1.5              6.078e+01  1.706e+01   3.563 0.000368 ***
Stories2                6.916e+01  1.699e+01   4.070 4.72e-05 ***
Year_Built              2.939e-01  2.365e-02  12.427  < 2e-16 ***
Fin_sqft                1.000e-01  1.570e-03  63.697  < 2e-16 ***
Units1                  1.221e+02  1.222e+01   9.992  < 2e-16 ***
Units2                  1.633e+01  1.223e+01   1.336 0.181724    
Units3                 -2.964e+01  1.316e+01  -2.253 0.024293 *  
Bdrms0                  1.533e+02  3.080e+01   4.976 6.54e-07 ***
Bdrms1                  1.302e+02  1.678e+01   7.760 8.80e-15 ***
Bdrms2                  1.424e+02  1.498e+01   9.507  < 2e-16 ***
Bdrms3                  1.461e+02  1.489e+01   9.817  < 2e-16 ***
Bdrms4                  1.268e+02  1.483e+01   8.553  < 2e-16 ***
Bdrms5                  1.240e+02  1.482e+01   8.363  < 2e-16 ***
Bdrms6                  1.027e+02  1.485e+01   6.914 4.83e-12 ***
Bdrms7                  7.282e+01  1.583e+01   4.599 4.27e-06 ***
Bdrms8                  1.021e+02  1.666e+01   6.129 8.97e-10 ***
Fbath0                 -6.484e+01  2.266e+01  -2.861 0.004228 ** 
Fbath1                 -4.116e+01  1.579e+01  -2.606 0.009163 ** 
Fbath2                 -1.549e+01  1.569e+01  -0.987 0.323435    
Fbath3                  2.939e+01  1.559e+01   1.885 0.059381 .  
Fbath4                  6.073e+01  1.672e+01   3.632 0.000282 ***
Lotsize                 2.086e-03  1.613e-04  12.936  < 2e-16 ***
Sale_date               6.175e-03  4.340e-04  14.230  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 73.07 on 24413 degrees of freedom
Multiple R-squared:  0.4468,	Adjusted R-squared:  0.446 
F-statistic: 597.4 on 33 and 24413 DF,  p-value: < 2.2e-16
```

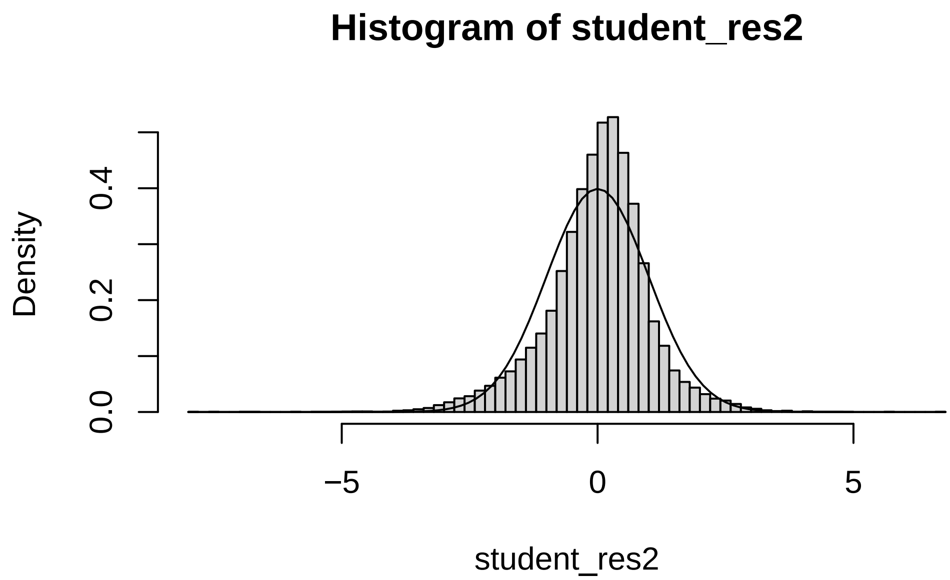
```r
summ2$adj.r.squared
```

```
[1] 0.4460105
```

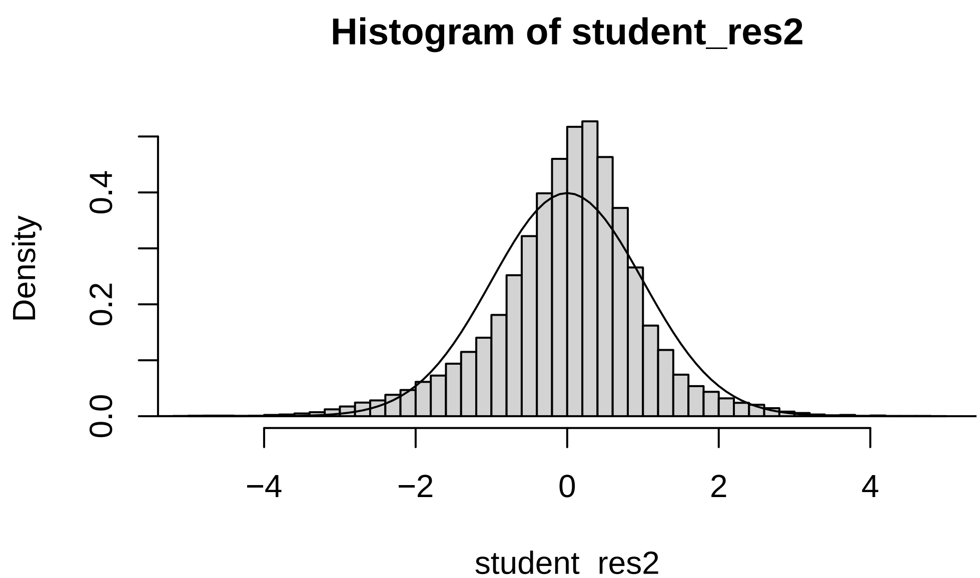
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



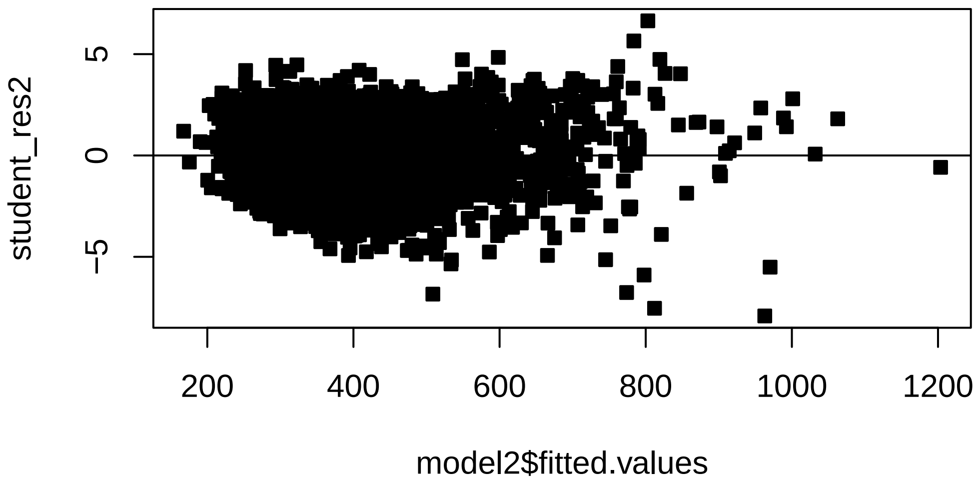

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



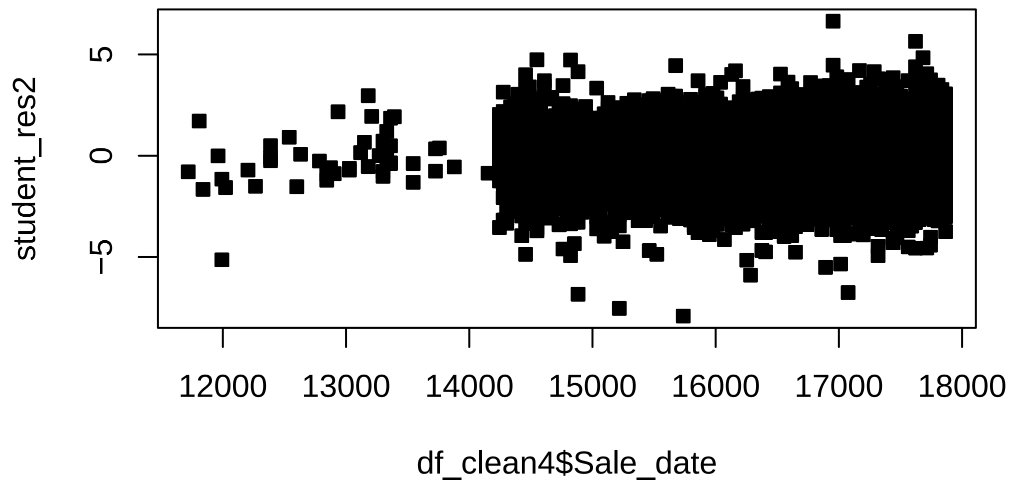

```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



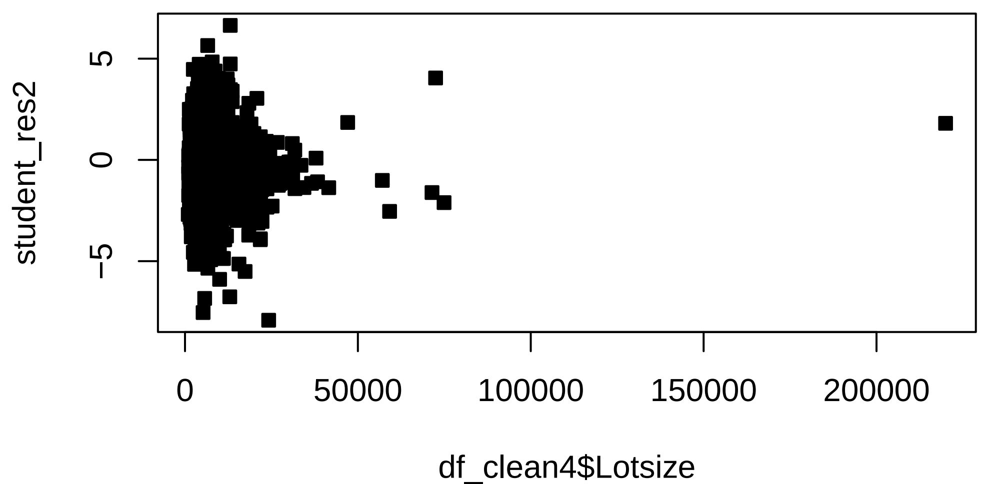
```r
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



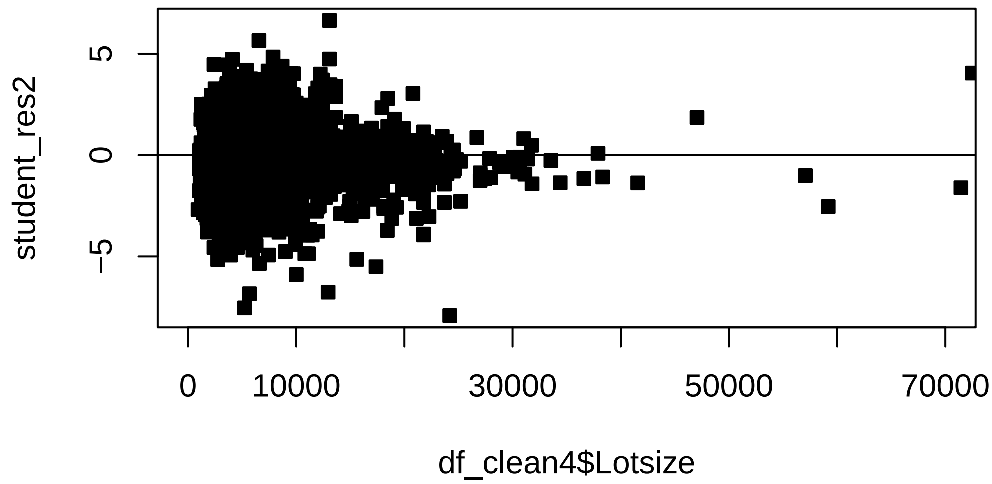

```r
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```




```
plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1)
```



```r
plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1,xlim=c(0,70000))
abline(h=0)
```




```r
df_clean4=df_clean4[df_clean4$Lotsize<70000,]

model2=lm(sqrt(Sale_price)~.,df_clean4)
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

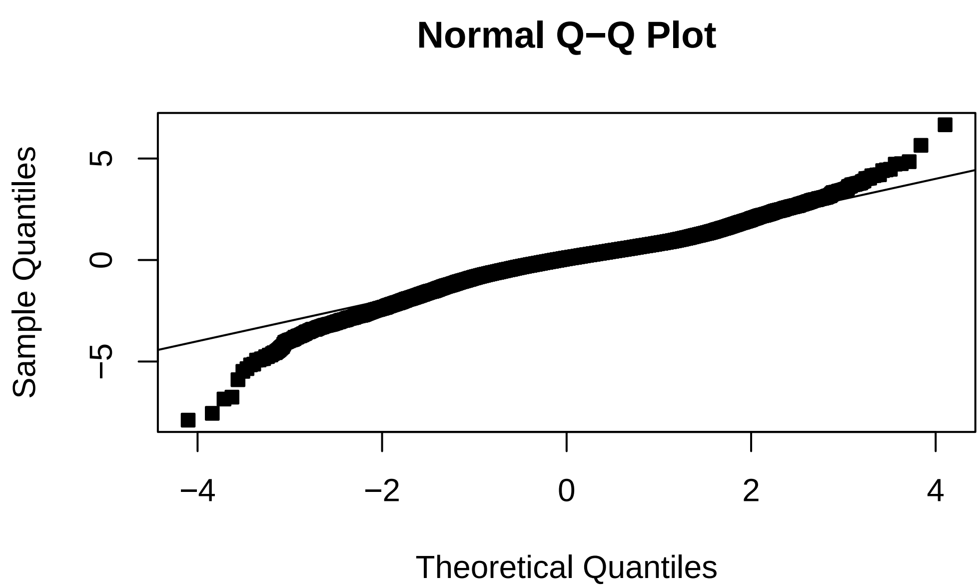
```
Call:
lm(formula = sqrt(Sale_price) ~ ., data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max 
-542.27  -36.96    6.36   42.19  482.94 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)           -7.909e+02  5.585e+01 -14.161  < 2e-16 ***
District               3.748e+00  1.148e-01  32.656  < 2e-16 ***
ExtwallBlock          -5.561e+00  6.150e+00  -0.904 0.365881
ExtwallBrick           1.270e+01  1.204e+00  10.551  < 2e-16 ***
ExtwallFiber-Cement    1.606e+01  6.207e+00   2.588 0.009663 **
ExtwallFrame          -9.858e+00  1.627e+00  -6.059 1.39e-09 ***
ExtwallMasonry / Frame  8.679e+00  2.844e+00   3.051 0.002281 **
ExtwallPrem Wood       1.454e+01  9.363e+00   1.553 0.120525
ExtwallStone           9.085e+00  2.541e+00   3.575 0.000351 ***
ExtwallStucco          1.885e+01  3.572e+00   5.278 1.32e-07 ***
Stories1               4.700e+01  1.708e+01   2.752 0.005926 **
Stories1.5             6.081e+01  1.705e+01   3.566 0.000363 ***
Stories2               6.931e+01  1.699e+01   4.080 4.52e-05 ***
Year_Built             3.028e-01  2.426e-02  12.480  < 2e-16 ***
Fin_sqft               1.000e-01  1.580e-03  63.322  < 2e-16 ***
Units1                 1.221e+02  1.222e+01   9.998  < 2e-16 ***
Units2                 1.621e+01  1.222e+01   1.327 0.184655
Units3                -2.941e+01  1.315e+01  -2.236 0.025354 *
Bdrms0                 1.529e+02  3.078e+01   4.967 6.84e-07 ***
Bdrms1                 1.300e+02  1.677e+01   7.752 9.38e-15 ***
Bdrms2                 1.421e+02  1.498e+01   9.492  < 2e-16 ***
Bdrms3                 1.459e+02  1.488e+01   9.805  < 2e-16 ***
Bdrms4                 1.266e+02  1.482e+01   8.539  < 2e-16 ***
Bdrms5                 1.239e+02  1.482e+01   8.359  < 2e-16 ***
Bdrms6                 1.024e+02  1.484e+01   6.902 5.25e-12 ***
Bdrms7                 7.278e+01  1.583e+01   4.598 4.28e-06 ***
Bdrms8                 1.020e+02  1.665e+01   6.127 9.09e-10 ***
Fbath0                -6.549e+01  2.265e+01  -2.891 0.003843 **
Fbath1                -4.185e+01  1.579e+01  -2.651 0.008027 **
Fbath2                -1.614e+01  1.568e+01  -1.030 0.303245
Fbath3                 2.830e+01  1.558e+01   1.817 0.069294 .
Fbath4                 6.041e+01  1.671e+01   3.614 0.000302 ***
Lotsize                1.921e-03  1.888e-04  10.173  < 2e-16 ***
Sale_date              6.170e-03  4.338e-04  14.225  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 73.04 on 24409 degrees of freedom
Multiple R-squared:  0.4443,    Adjusted R-squared:  0.4436
F-statistic: 591.5 on 33 and 24409 DF,  p-value: < 2.2e-16
```

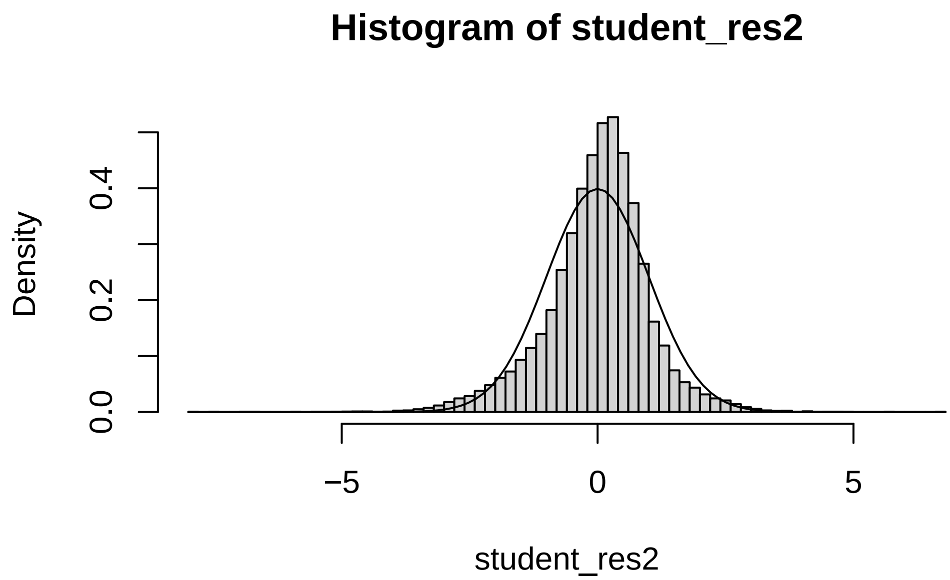
```r
summ2$adj.r.squared
```

```
[1] 0.4435961
```

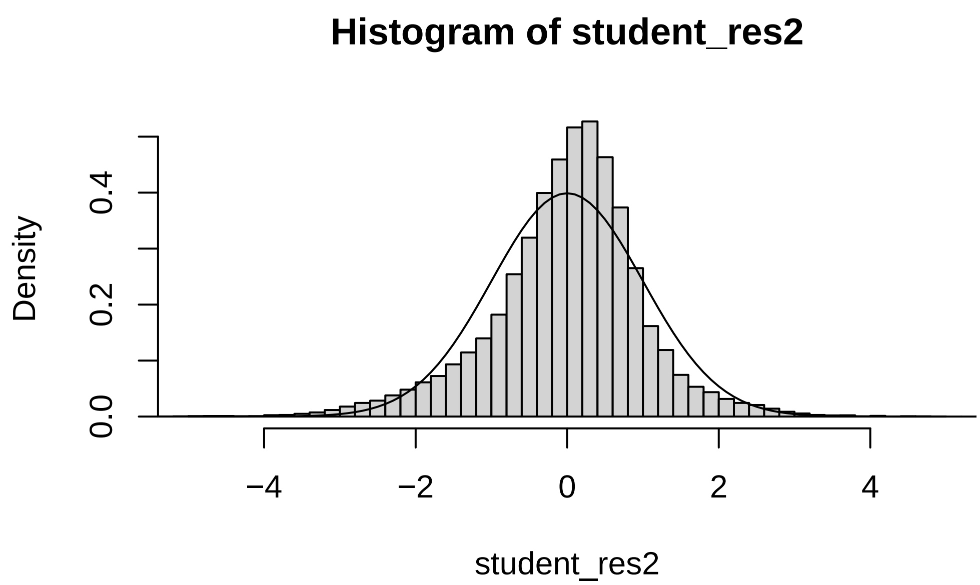
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



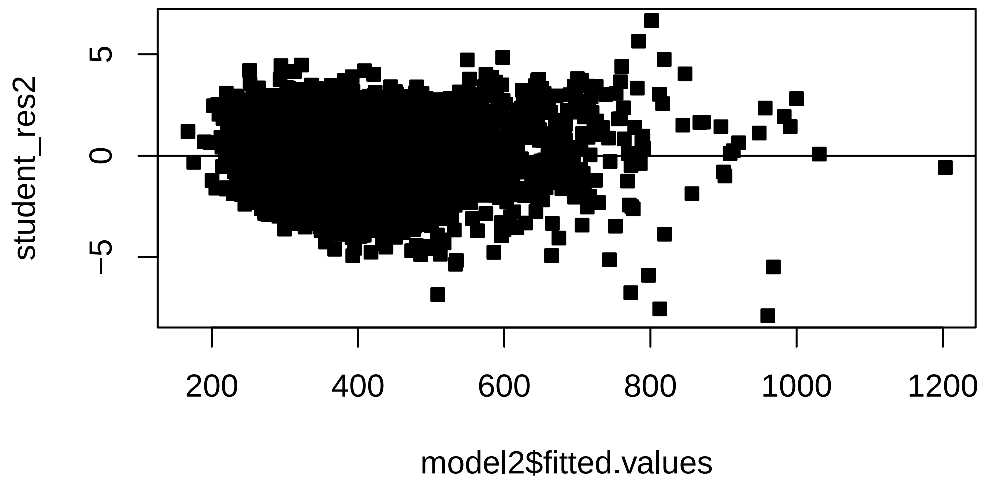

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



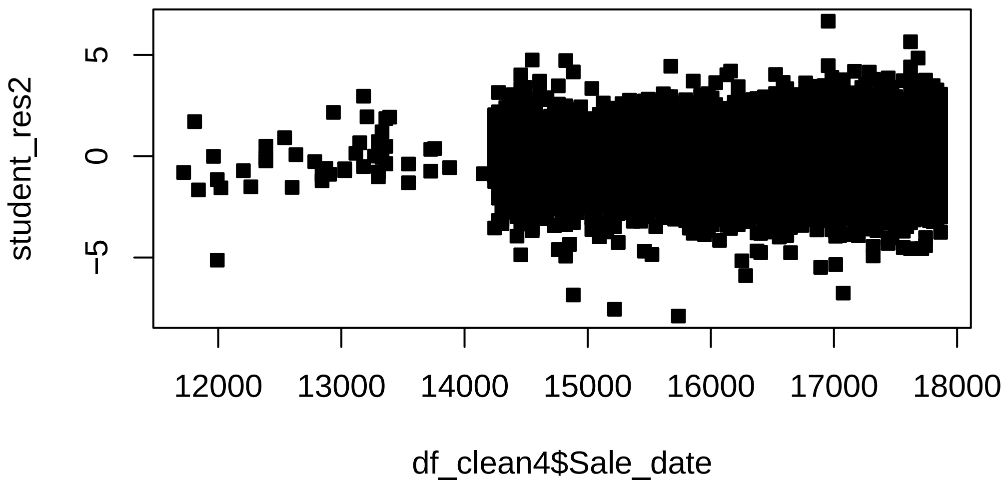

```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



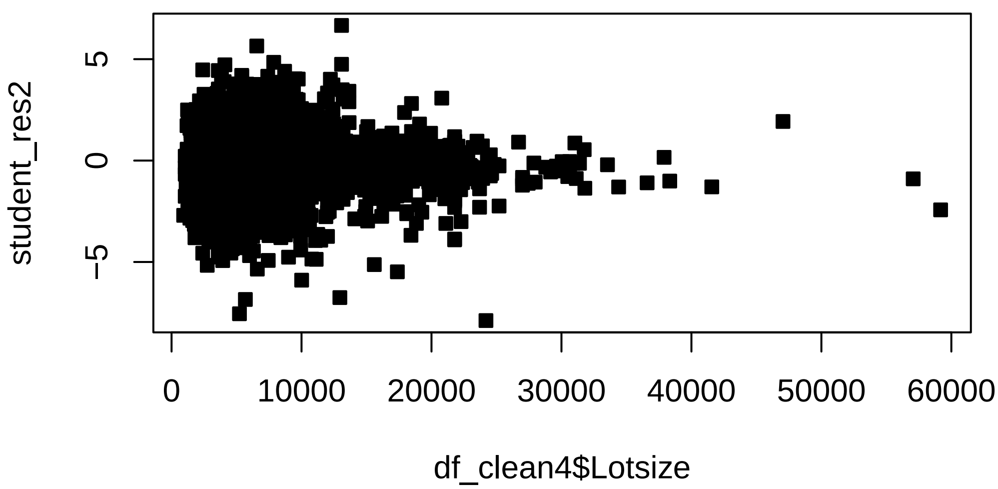
```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



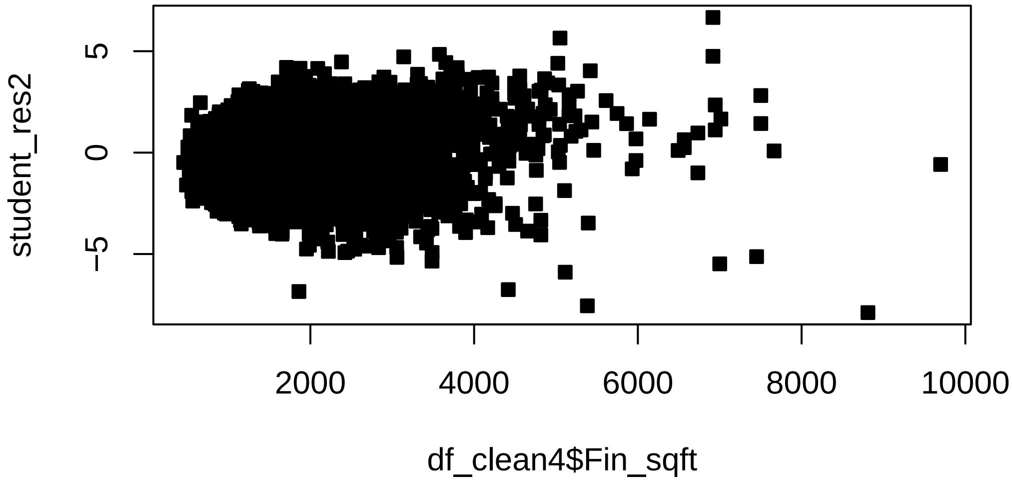

```
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



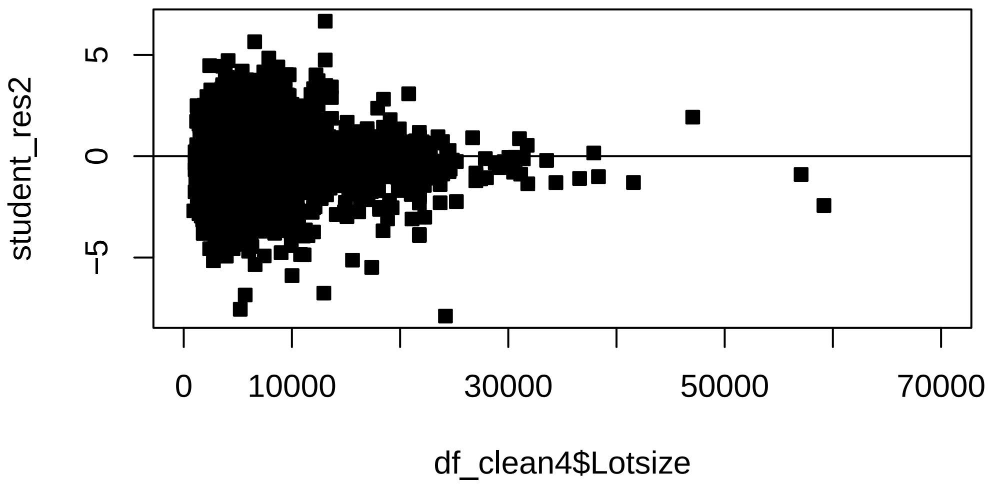

```
plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1)
```



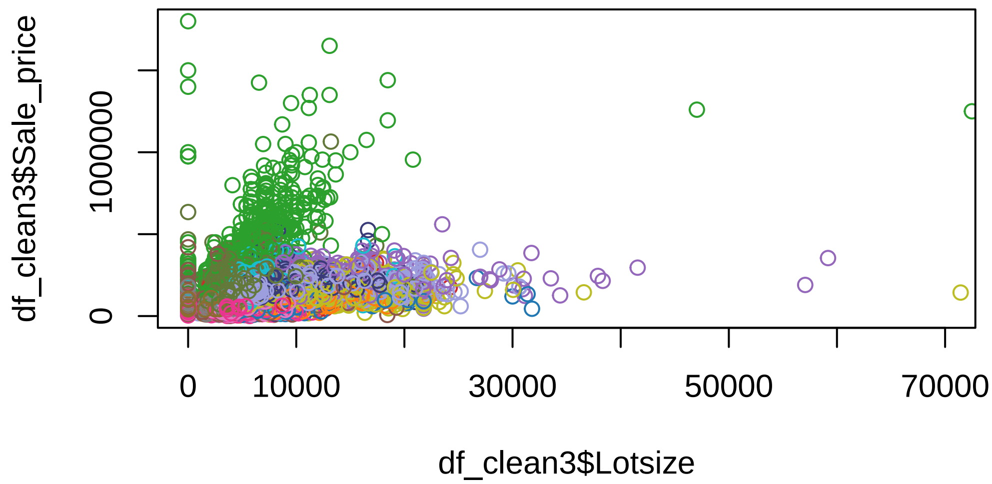
```r
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```



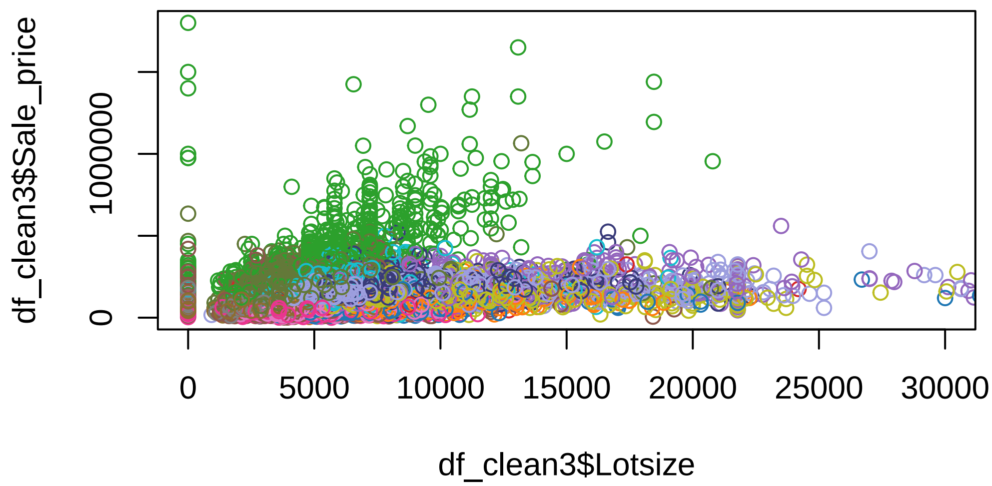

```r
plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1,xlim=c(0,70000))
abline(h=0)
```



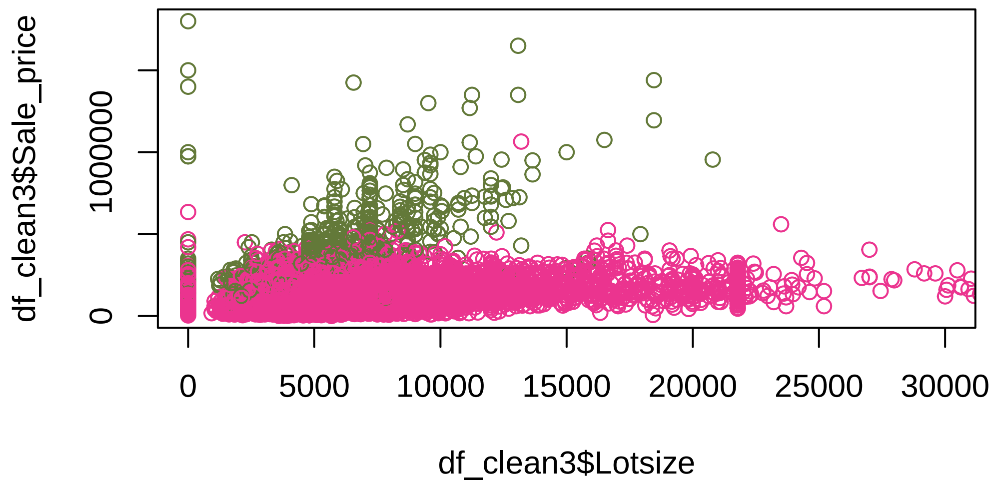

```r
custom_palette <- c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
  "#bcbd22", "#17becf", "#393b79",
  "#8c6d31", "#9c9ede", "#637939", "#eb348f"
)

plot(df_clean3$Lotsize,df_clean3$Sale_price,
     xlim=c(0,70000),
     col=custom_palette[df_clean3$District])
```




```
plot(df_clean3$Lotsize,df_clean3$Sale_price,xlim=c(0,30000),
     col=custom_palette[df_clean3$District])
```

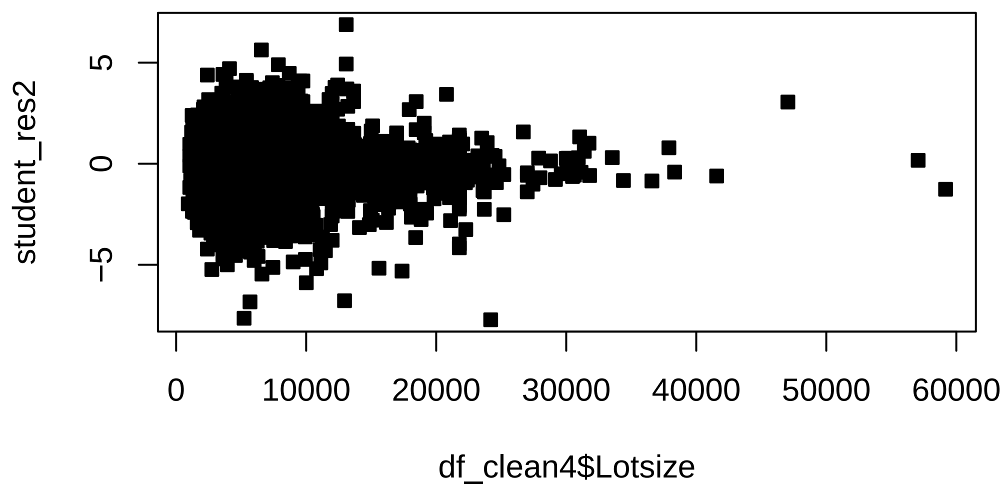


```
custom_palette <- c(
  "#eb348f","#eb348f", "#637939", "#eb348f",
  "#eb348f", "#eb348f", "#eb348f", "#eb348f",
  "#eb348f", "#eb348f", "#eb348f",
  "#eb348f", "#eb348f", "#eb348f", "#eb348f"
)

plot(df_clean3$Lotsize,df_clean3$Sale_price,xlim=c(0,30000),col=custom_palette[df_clean3$D
```




```
#green is 3 and 14 here

# order is
# Red
# Green
# Blue
# Cyan
# Magenta
# Yellow
# Black
# Gray
```

```
df_clean4$d_3=df_clean4$District==3
#df_clean4$d_3=mapply('||',df_clean4$District==3,df_clean4$District==14)
model2=lm(sqrt(Sale_price)~ District +  Extwall +
                      Stories   +    Year_Built   + Fin_sqft   +
                      Units    +     Bdrms    +
                      Fbath   +    log(Lotsize)    + Sale_date + d_3*log(Lotsize)-d_3,
          df_clean4)


model2=lm(sqrt(Sale_price)~ District +  Extwall +
            Stories   +    Year_Built   + Fin_sqft   +
            Units    +     Bdrms    +
            Fbath   +    log(Lotsize)    + Sale_date + log(Lotsize)*District,
          df_clean4)

student_res2=rstudent(model2)
plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1)
```



```r
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

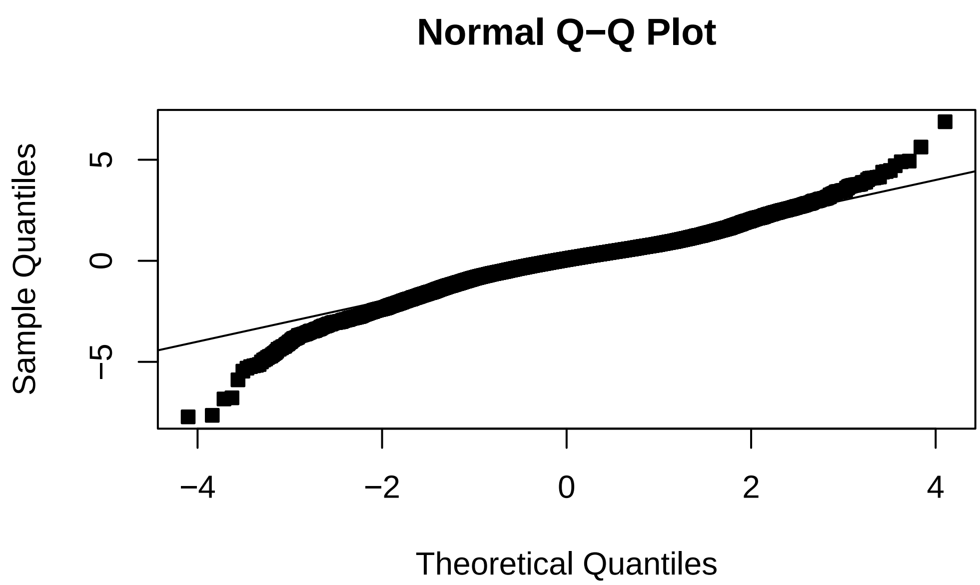
```
Call:
lm(formula = sqrt(Sale_price) ~ District + Extwall + Stories +
    Year_Built + Fin_sqft + Units + Bdrms + Fbath + log(Lotsize) +
    Sale_date + log(Lotsize) * District, data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max
-542.77  -37.26    5.82   41.94  496.12

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -5.146e+02  5.836e+01  -8.818  < 2e-16 ***
District               -2.657e+01  2.617e+00 -10.150  < 2e-16 ***
ExtwallBlock           -5.663e+00  6.119e+00  -0.925 0.354740
ExtwallBrick            1.153e+01  1.198e+00   9.625  < 2e-16 ***
ExtwallFiber-Cement     1.916e+01  6.187e+00   3.096 0.001965 **
ExtwallFrame           -1.042e+01  1.619e+00  -6.437 1.24e-10 ***
ExtwallMasonry / Frame  7.347e+00  2.831e+00   2.595 0.009466 **
ExtwallPrem Wood        1.642e+01  9.309e+00   1.763 0.077851 .
ExtwallStone            8.615e+00  2.528e+00   3.407 0.000657 ***
ExtwallStucco           1.742e+01  3.554e+00   4.902 9.54e-07 ***
Stories1                4.078e+01  1.701e+01   2.398 0.016479 *
Stories1.5              5.520e+01  1.698e+01   3.251 0.001150 **
Stories2                6.383e+01  1.691e+01   3.774 0.000161 ***
Year_Built              2.076e-01  2.572e-02   8.072 7.22e-16 ***
Fin_sqft                9.922e-02  1.574e-03  63.021  < 2e-16 ***
Units1                  1.243e+02  1.215e+01  10.225  < 2e-16 ***
Units2                  1.859e+01  1.216e+01   1.529 0.126302
Units3                 -2.874e+01  1.309e+01  -2.196 0.028078 *
Bdrms0                  1.500e+02  3.063e+01   4.898 9.76e-07 ***
Bdrms1                  1.301e+02  1.669e+01   7.796 6.65e-15 ***
Bdrms2                  1.411e+02  1.490e+01   9.474  < 2e-16 ***
Bdrms3                  1.441e+02  1.480e+01   9.731  < 2e-16 ***
Bdrms4                  1.252e+02  1.475e+01   8.489  < 2e-16 ***
Bdrms5                  1.218e+02  1.474e+01   8.263  < 2e-16 ***
Bdrms6                  1.009e+02  1.477e+01   6.834 8.47e-12 ***
```

```
Bdrms7                  7.057e+01  1.575e+01   4.481 7.45e-06 ***
Bdrms8                  9.879e+01  1.657e+01   5.963 2.51e-09 ***
Fbath0                 -7.195e+01  2.255e+01  -3.192 0.001417 **
Fbath1                 -4.903e+01  1.572e+01  -3.120 0.001811 **
Fbath2                 -2.305e+01  1.561e+01  -1.477 0.139781
Fbath3                  2.206e+01  1.551e+01   1.422 0.154986
Fbath4                  5.559e+01  1.663e+01   3.342 0.000832 ***
log(Lotsize)           -7.558e+00  3.074e+00  -2.458 0.013971 *
Sale_date               6.125e-03  4.316e-04  14.193  < 2e-16 ***
District:log(Lotsize)   3.514e+00  3.031e-01  11.595  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 72.67 on 24408 degrees of freedom
Multiple R-squared:   0.45, Adjusted R-squared:  0.4492
F-statistic: 587.3 on 34 and 24408 DF,  p-value: < 2.2e-16
```

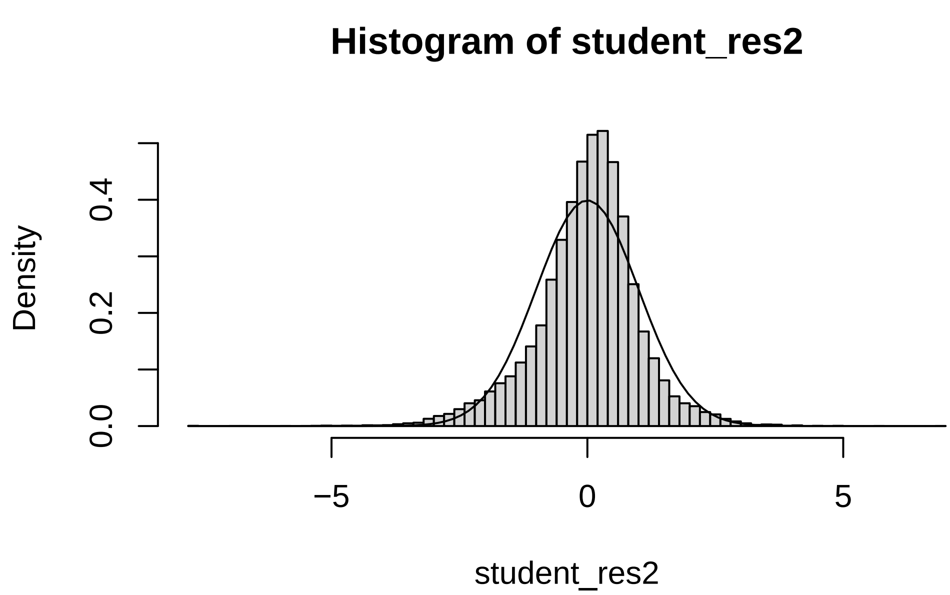
```r
summ2$adj.r.squared
```

```
[1] 0.4492185
```

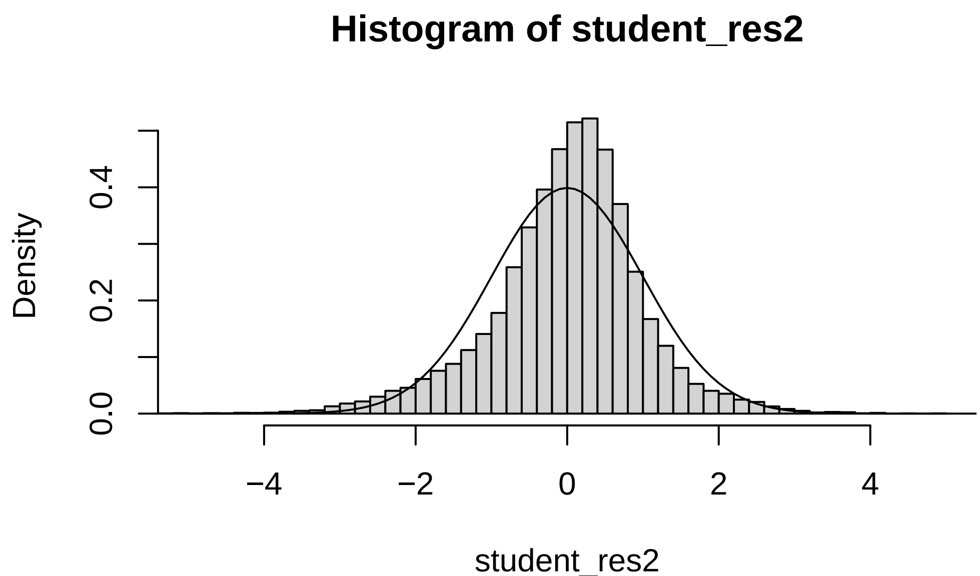
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



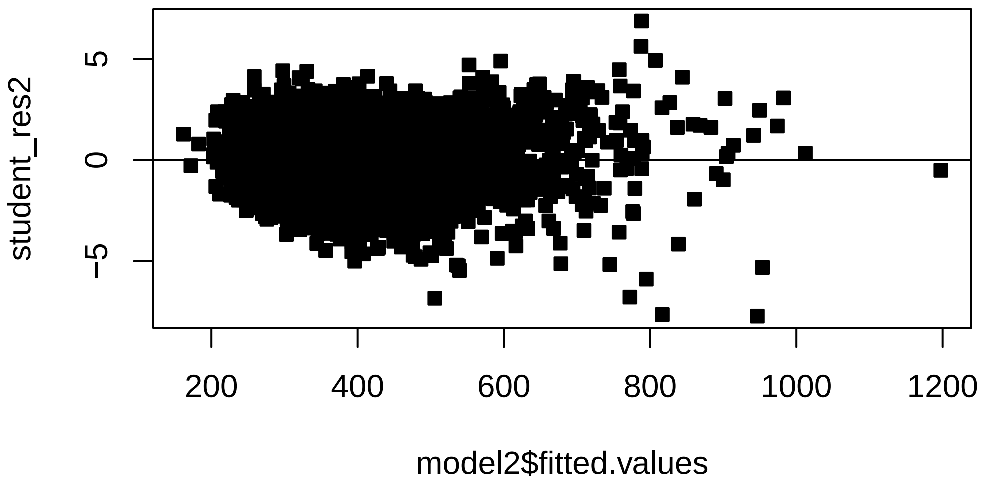

```
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



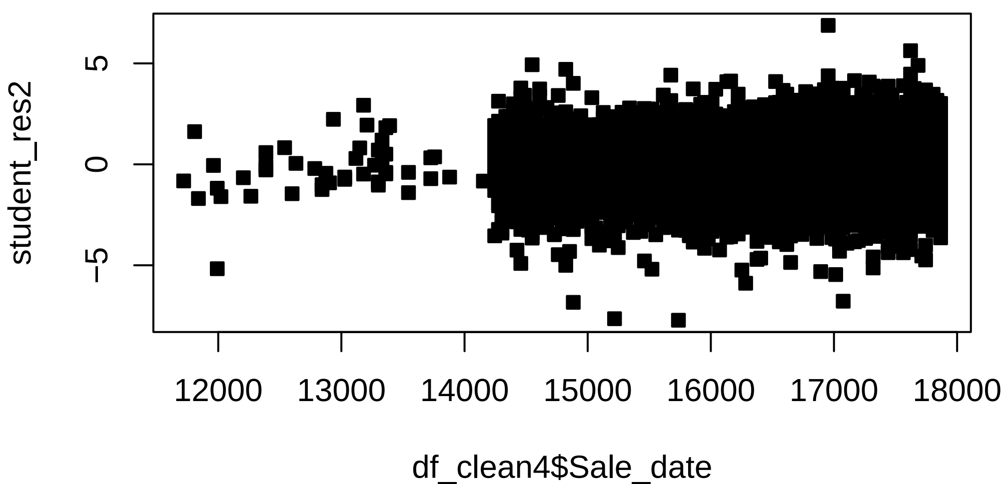
```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



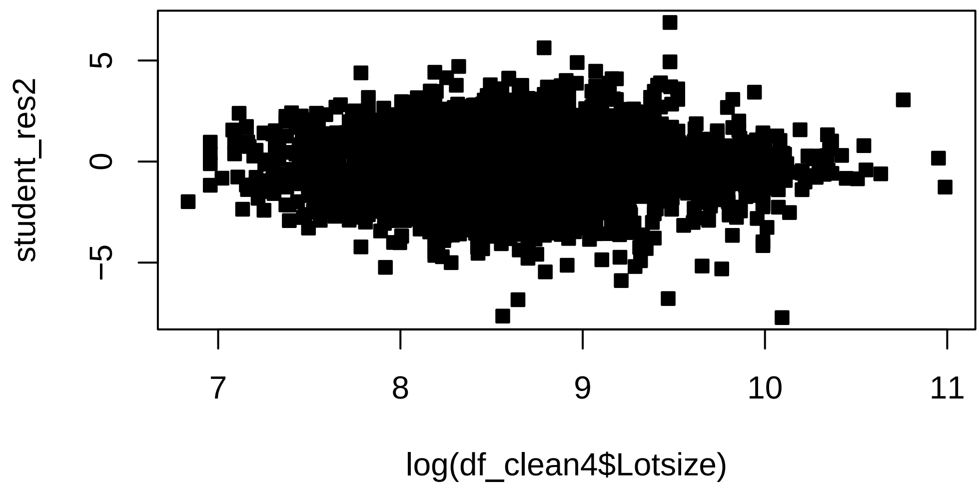

```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



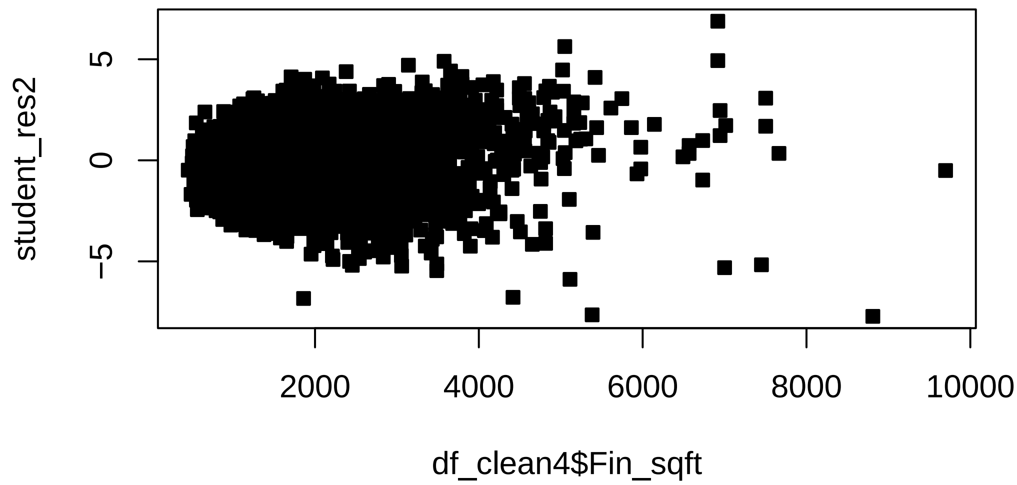

```
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



```
plot(log(df_clean4$Lotsize) ,student_res2,pch=22,bg=1)
```




```
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```




```
# plot(df_clean4$Lotsize ,student_res2,pch=22,bg=1,xlim=c(0,70000))
# abline(h=0)



# df_clean4$d_3=df_clean4$District==3
# df_clean4$d_3or14=mapply('||',df_clean4$District==3,df_clean4$District==14)
model2=lm(sqrt(Sale_price)~District +  Extwall +     Stories  +    Year_Built  + Fin_sqft
            Fbath    + Sale_date ,df_clean4)
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

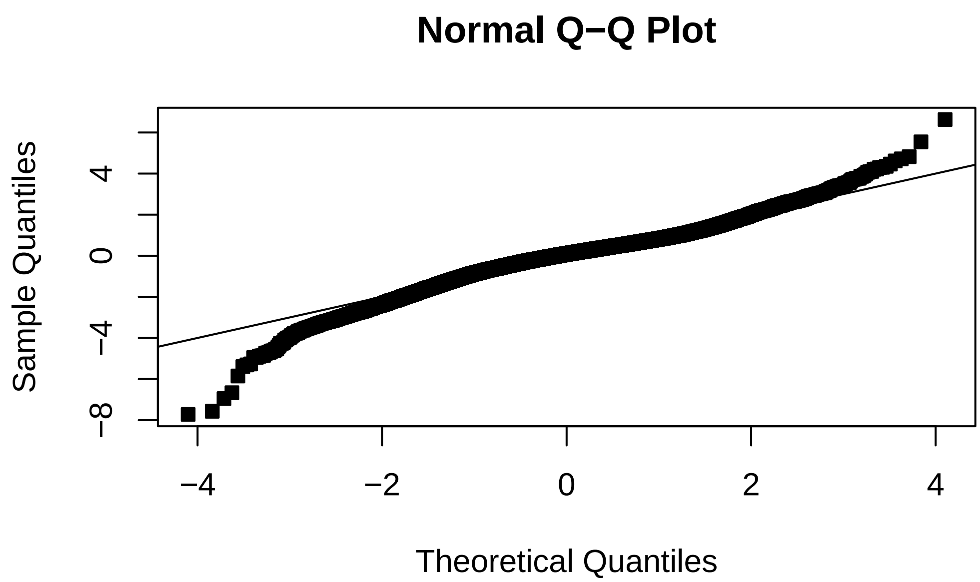
```
Call:
lm(formula = sqrt(Sale_price) ~ District + Extwall + Stories +
    Year_Built + Fin_sqft + Units + Bdrms + Fbath + Sale_date,
    data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-552.23  -37.27    6.58   42.12  481.52

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -1.005e+03  5.184e+01 -19.385  < 2e-16 ***
District                3.749e+00  1.150e-01  32.598  < 2e-16 ***
ExtwallBlock           -5.282e+00  6.163e+00  -0.857 0.391428    
ExtwallBrick            1.209e+01  1.205e+00  10.035  < 2e-16 ***
ExtwallFiber-Cement     1.379e+01  6.216e+00   2.219 0.026525 *  
ExtwallFrame           -9.325e+00  1.630e+00  -5.722 1.06e-08 ***
ExtwallMasonry / Frame  9.131e+00  2.850e+00   3.204 0.001358 ** 
ExtwallPrem Wood        1.819e+01  9.375e+00   1.940 0.052362 .  
ExtwallStone            9.162e+00  2.546e+00   3.598 0.000321 ***
ExtwallStucco           1.866e+01  3.579e+00   5.213 1.87e-07 ***
Stories1                5.433e+01  1.710e+01   3.178 0.001487 ** 
Stories1.5              6.722e+01  1.708e+01   3.937 8.29e-05 ***
Stories2                7.451e+01  1.702e+01   4.379 1.20e-05 ***
Year_Built              4.088e-01  2.196e-02  18.617  < 2e-16 ***
Fin_sqft                1.041e-01  1.531e-03  67.975  < 2e-16 ***
Units1                  1.241e+02  1.224e+01  10.138  < 2e-16 ***
Units2                  1.628e+01  1.225e+01   1.329 0.183906    
Units3                 -2.981e+01  1.318e+01  -2.262 0.023709 *  
Bdrms0                  1.556e+02  3.085e+01   5.043 4.61e-07 ***
Bdrms1                  1.349e+02  1.680e+01   8.026 1.05e-15 ***
Bdrms2                  1.460e+02  1.500e+01   9.731  < 2e-16 ***
Bdrms3                  1.495e+02  1.491e+01  10.026  < 2e-16 ***
Bdrms4                  1.295e+02  1.485e+01   8.717  < 2e-16 ***
Bdrms5                  1.262e+02  1.485e+01   8.503  < 2e-16 ***
Bdrms6                  1.041e+02  1.487e+01   6.998 2.66e-12 ***
Bdrms7                  7.311e+01  1.586e+01   4.610 4.05e-06 ***
Bdrms8                  1.010e+02  1.668e+01   6.052 1.45e-09 ***
Fbath0                 -6.371e+01  2.270e+01  -2.807 0.005010 ** 
Fbath1                 -4.003e+01  1.582e+01  -2.531 0.011389 *  
Fbath2                 -1.436e+01  1.571e+01  -0.914 0.360934    
Fbath3                  3.044e+01  1.561e+01   1.950 0.051242 .  
Fbath4                  6.273e+01  1.675e+01   3.746 0.000180 ***
Sale_date               6.220e-03  4.347e-04  14.310  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 73.19 on 24410 degrees of freedom
Multiple R-squared:  0.442, Adjusted R-squared:  0.4413 
F-statistic: 604.2 on 32 and 24410 DF,  p-value: < 2.2e-16
```

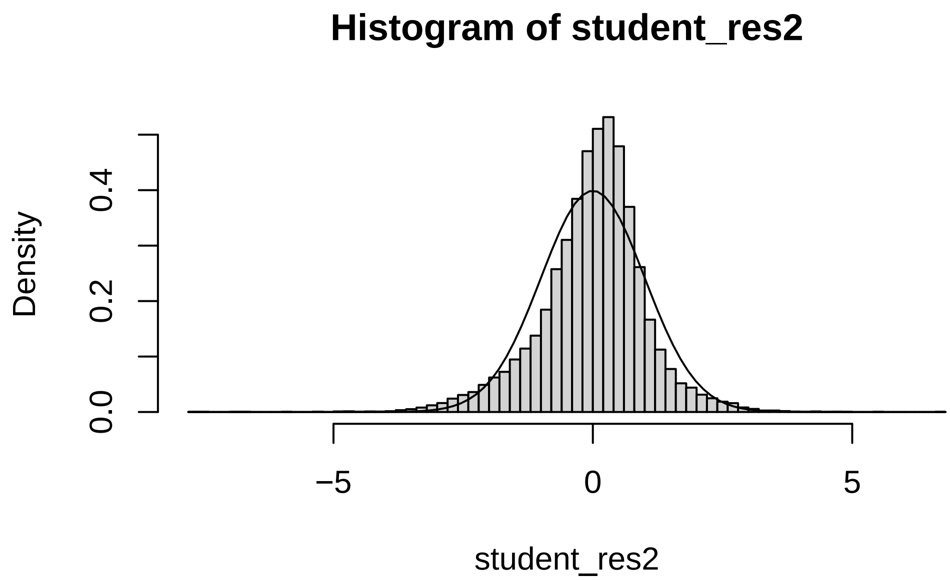
```r
summ2$adj.r.squared
```

```
[1] 0.4412601
```

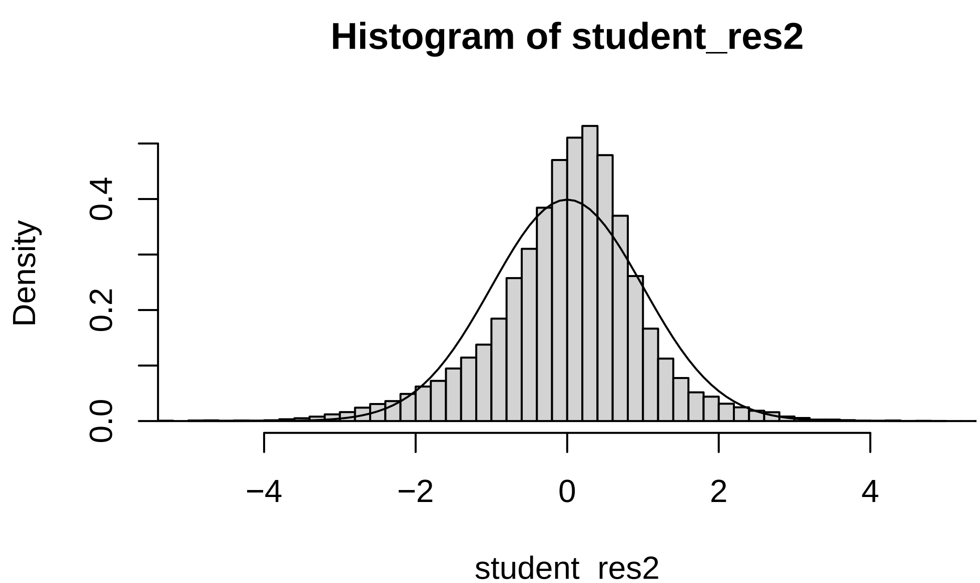
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



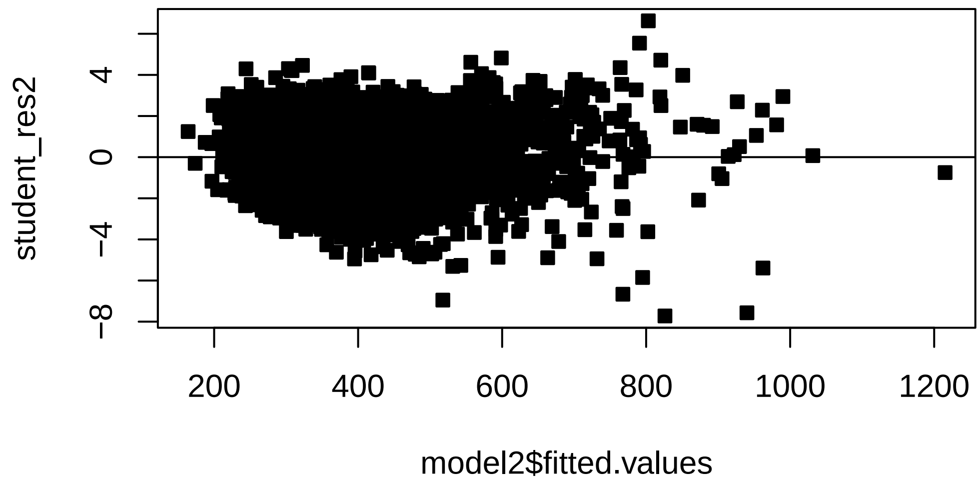

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



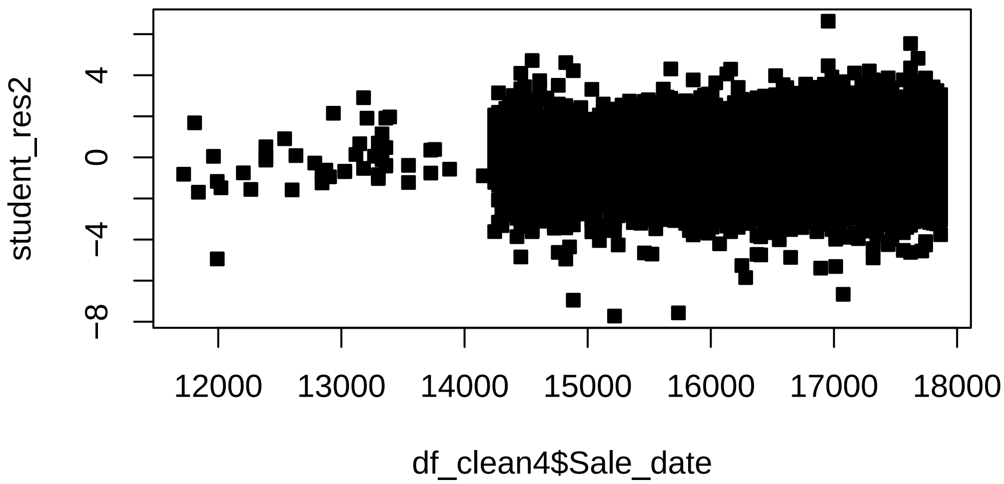

```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



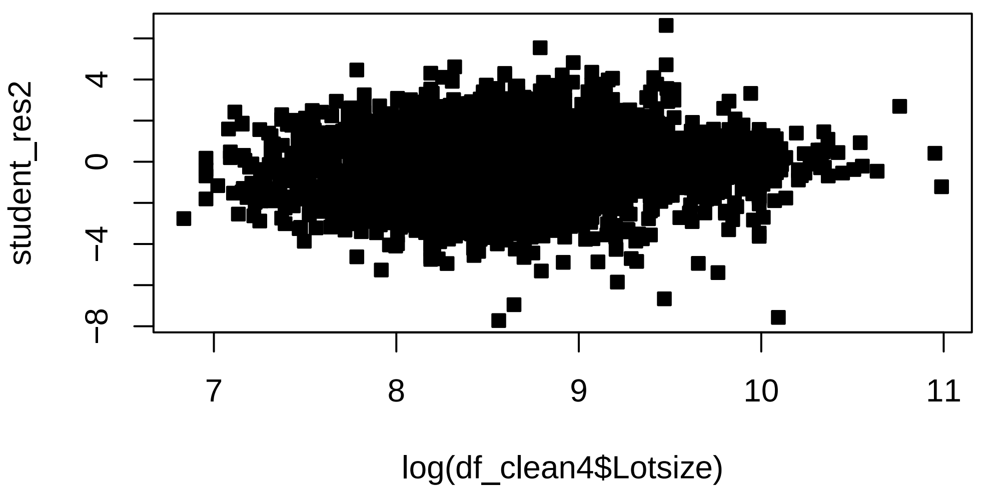
```r
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



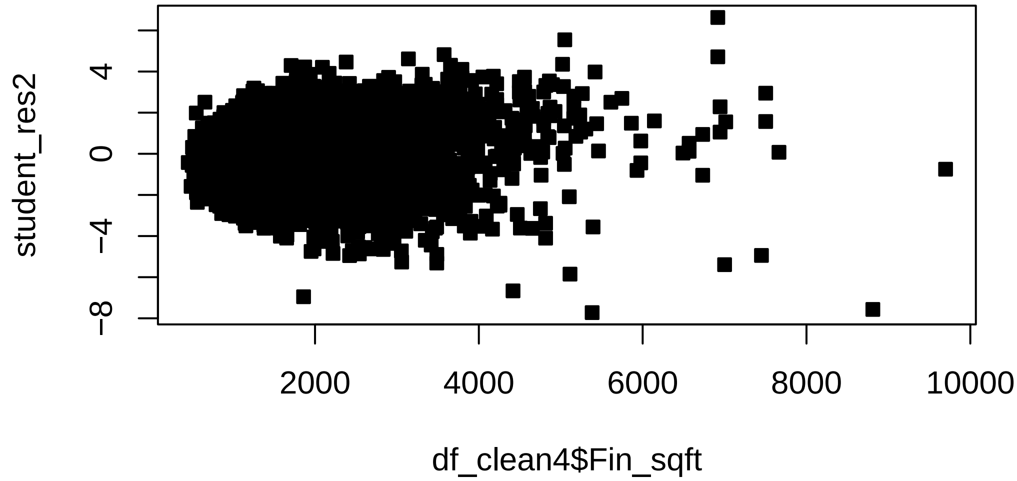

```r
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



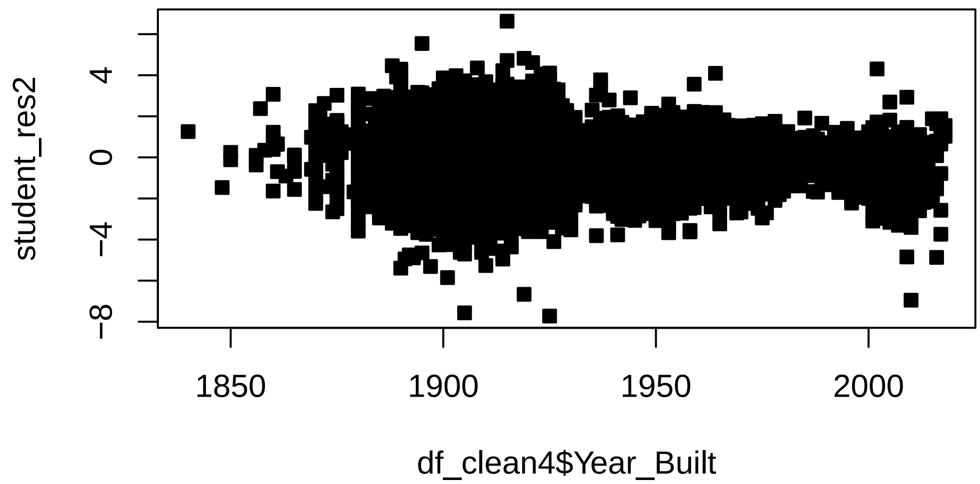

```
plot(log(df_clean4$Lotsize) ,student_res2,pch=22,bg=1)
```



```r
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```




```r
plot(df_clean4$Year_Built ,student_res2,pch=22,bg=1)
```




```
# abline(h=0)




# df_clean4$d_3=df_clean4$District==3
# df_clean4$d_3or14=mapply('||',df_clean4$District==3,df_clean4$District==14)
model2=lm(sqrt(Sale_price)~District +  Extwall +     Stories   +    sqrt(Year_Built)   + Fin
            Fbath     + Sale_date ,df_clean4)
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

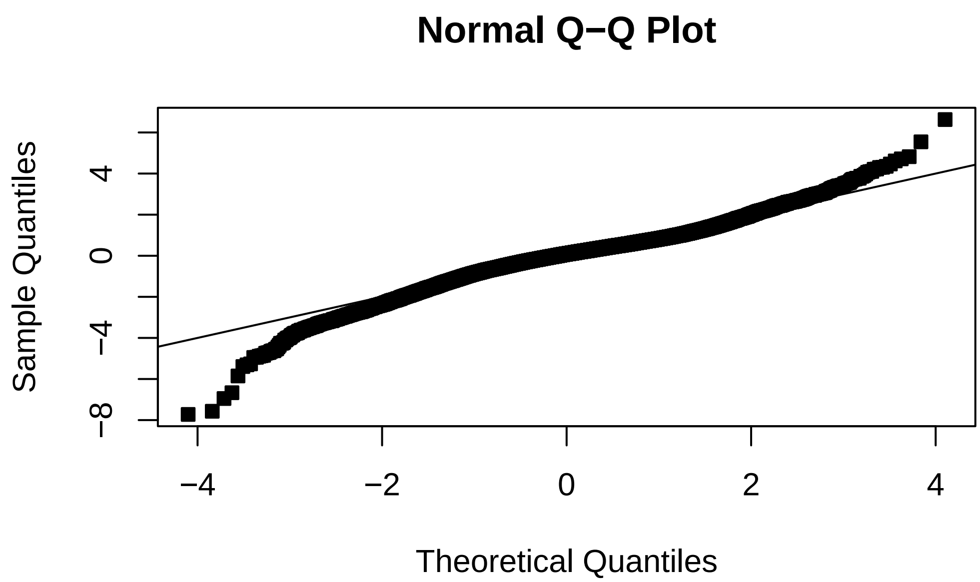
```
Call:
lm(formula = sqrt(Sale_price) ~ District + Extwall + Stories +
    sqrt(Year_Built) + Fin_sqft + Units + Bdrms + Fbath + Sale_date,
    data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-552.29  -37.29    6.60   42.10  481.44

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)           -1.801e+03  9.029e+01 -19.944  < 2e-16 ***
District               3.749e+00  1.150e-01  32.601  < 2e-16 ***
ExtwallBlock          -5.304e+00  6.163e+00  -0.861 0.389455    
ExtwallBrick           1.207e+01  1.205e+00  10.011  < 2e-16 ***
ExtwallFiber-Cement    1.388e+01  6.215e+00   2.233 0.025572 *  
ExtwallFrame          -9.313e+00  1.630e+00  -5.715 1.11e-08 ***
ExtwallMasonry / Frame  9.091e+00  2.850e+00   3.190 0.001426 ** 
ExtwallPrem Wood       1.822e+01  9.375e+00   1.943 0.052001 .  
ExtwallStone           9.123e+00  2.546e+00   3.583 0.000341 ***
ExtwallStucco          1.865e+01  3.579e+00   5.210 1.90e-07 ***
Stories1               5.427e+01  1.710e+01   3.174 0.001506 ** 
Stories1.5             6.718e+01  1.708e+01   3.934 8.38e-05 ***
Stories2               7.446e+01  1.702e+01   4.376 1.21e-05 ***
sqrt(Year_Built)       3.607e+01  1.934e+00  18.656  < 2e-16 ***
Fin_sqft               1.041e-01  1.531e-03  67.988  < 2e-16 ***
Units1                 1.241e+02  1.224e+01  10.137  < 2e-16 ***
Units2                 1.625e+01  1.225e+01   1.326 0.184694    
Units3                -2.983e+01  1.318e+01  -2.263 0.023639 *  
Bdrms0                 1.556e+02  3.085e+01   5.045 4.57e-07 ***
Bdrms1                 1.349e+02  1.680e+01   8.030 1.02e-15 ***
Bdrms2                 1.460e+02  1.500e+01   9.732  < 2e-16 ***
Bdrms3                 1.495e+02  1.491e+01  10.027  < 2e-16 ***
Bdrms4                 1.295e+02  1.485e+01   8.719  < 2e-16 ***
Bdrms5                 1.263e+02  1.485e+01   8.504  < 2e-16 ***
Bdrms6                 1.041e+02  1.487e+01   6.998 2.66e-12 ***
Bdrms7                 7.312e+01  1.586e+01   4.610 4.04e-06 ***
Bdrms8                 1.010e+02  1.668e+01   6.052 1.45e-09 ***
Fbath0                -6.361e+01  2.270e+01  -2.802 0.005078 ** 
Fbath1                -4.000e+01  1.582e+01  -2.529 0.011446 *  
Fbath2                -1.432e+01  1.571e+01  -0.911 0.362183    
Fbath3                 3.048e+01  1.561e+01   1.952 0.050931 .  
Fbath4                 6.277e+01  1.675e+01   3.748 0.000179 ***
Sale_date              6.219e-03  4.346e-04  14.309  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 73.19 on 24410 degrees of freedom
Multiple R-squared:  0.442, Adjusted R-squared:  0.4413 
F-statistic: 604.3 on 32 and 24410 DF,  p-value: < 2.2e-16
```

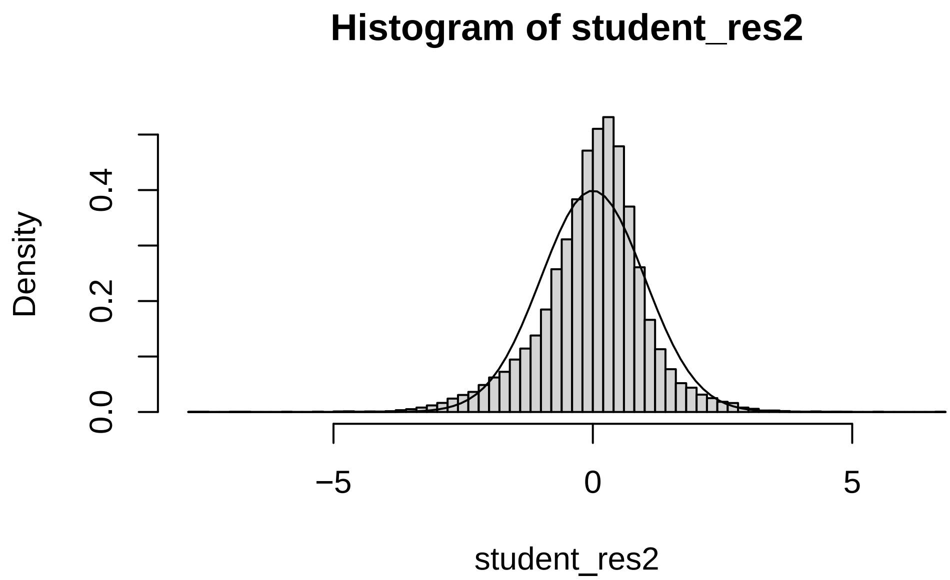
```r
summ2$adj.r.squared
```

```
[1] 0.4412929
```

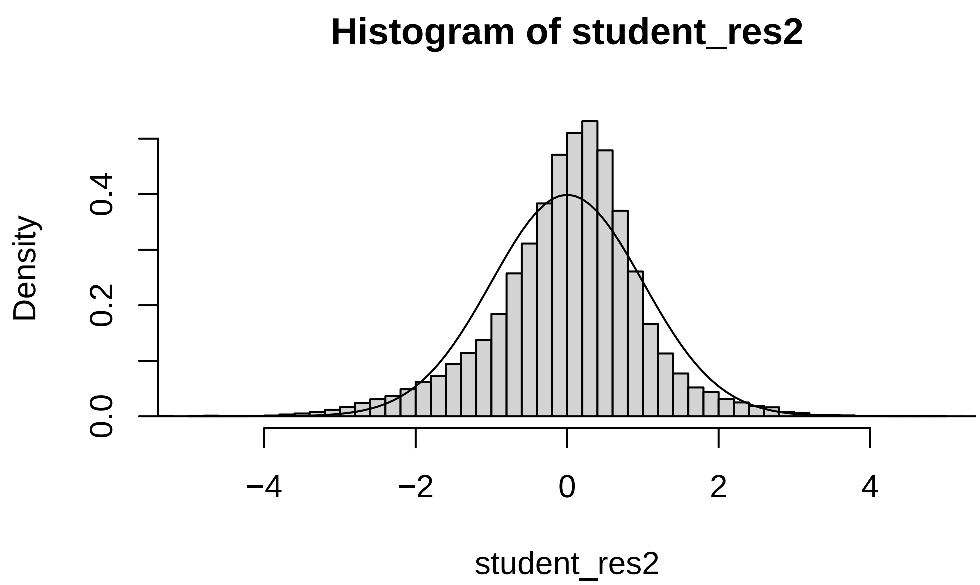
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



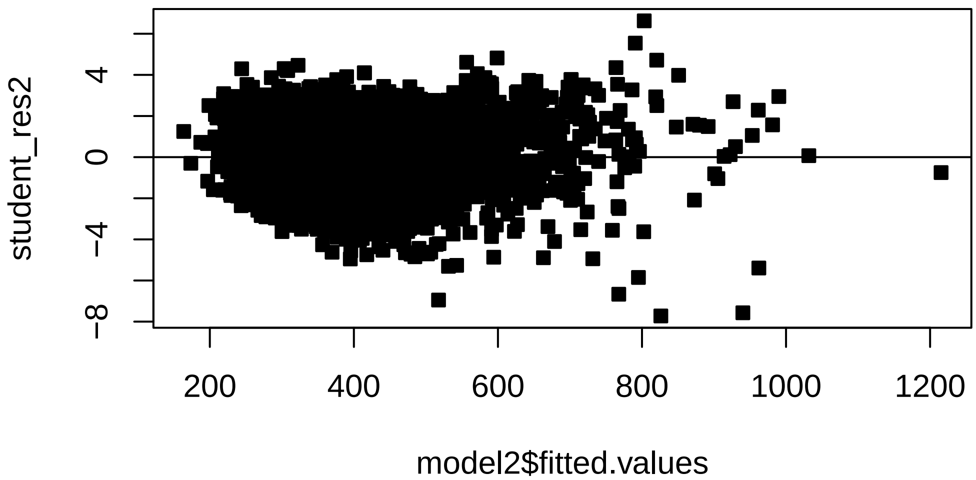

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



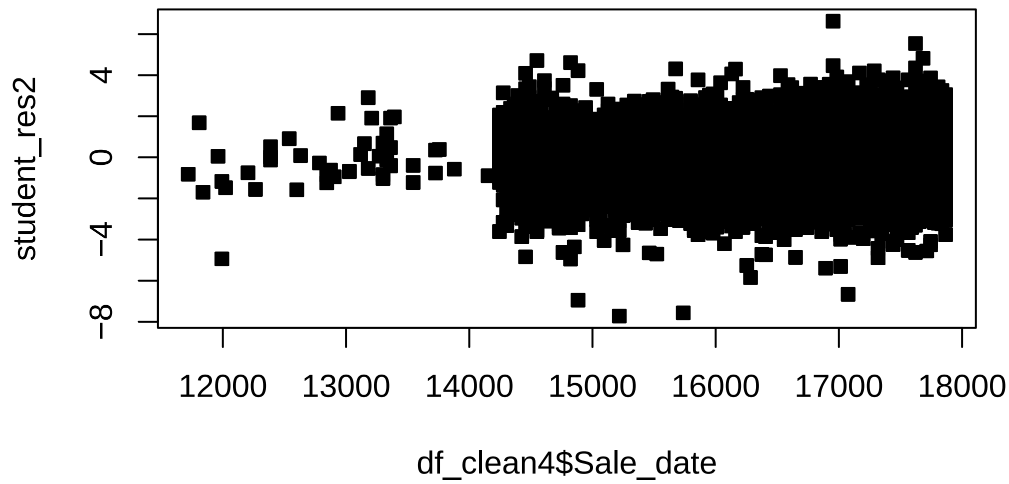

```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



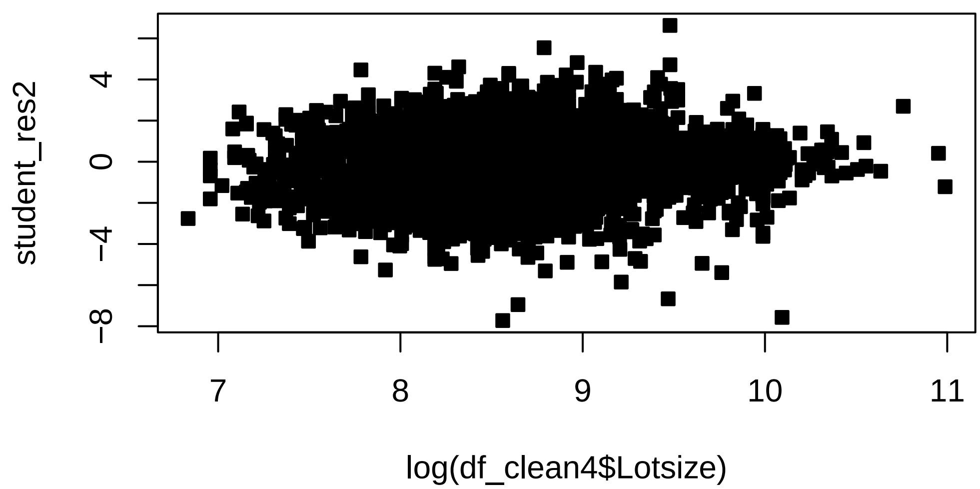
```r
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



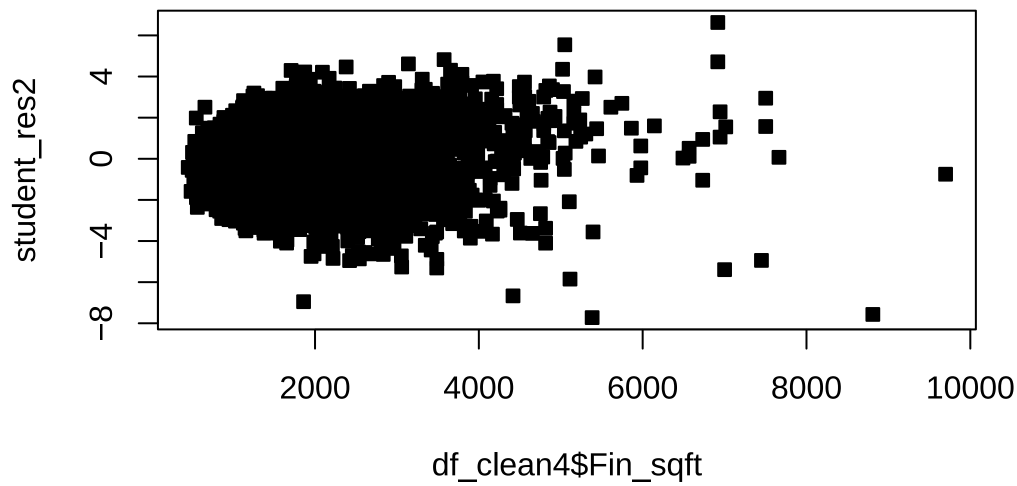

```r
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



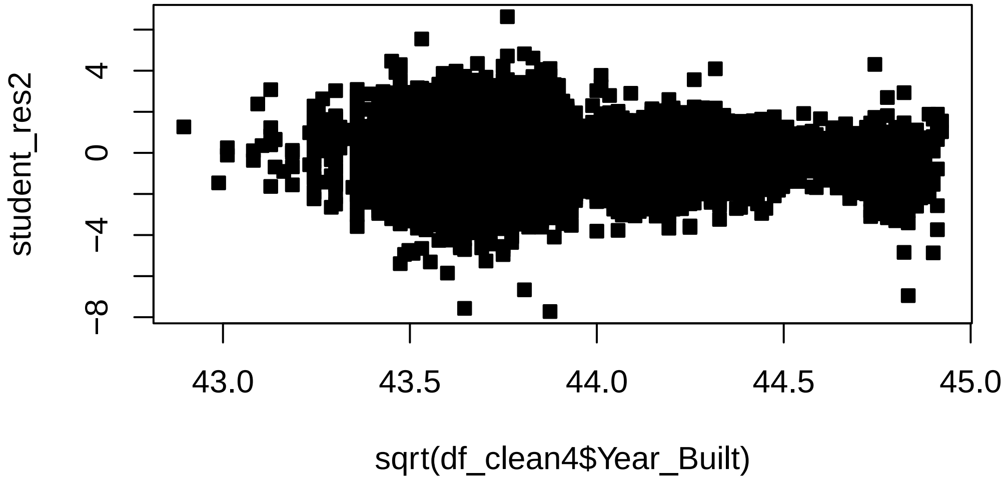

```
plot(log(df_clean4$Lotsize) ,student_res2,pch=22,bg=1)
```



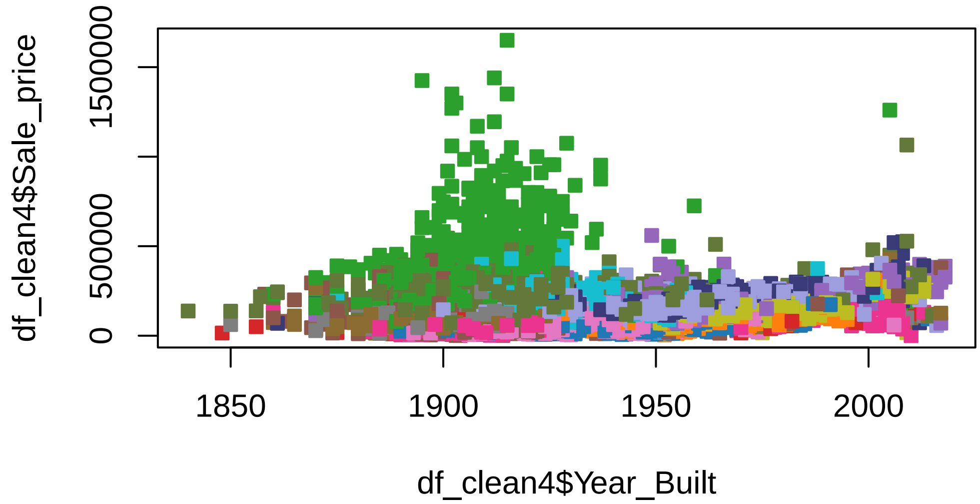
```r
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```



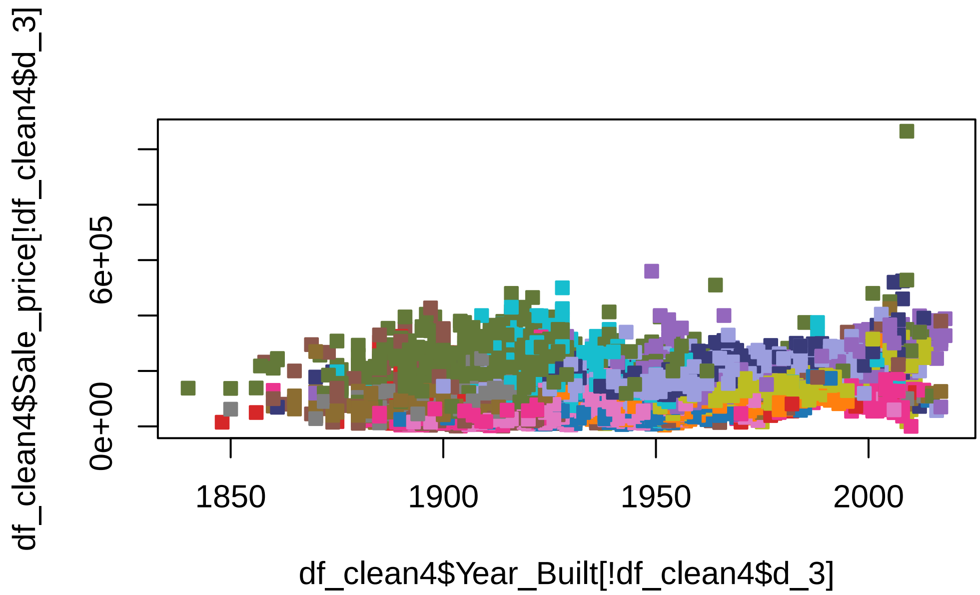

```r
plot(sqrt(df_clean4$Year_Built) ,student_res2,pch=22,bg=1)
```



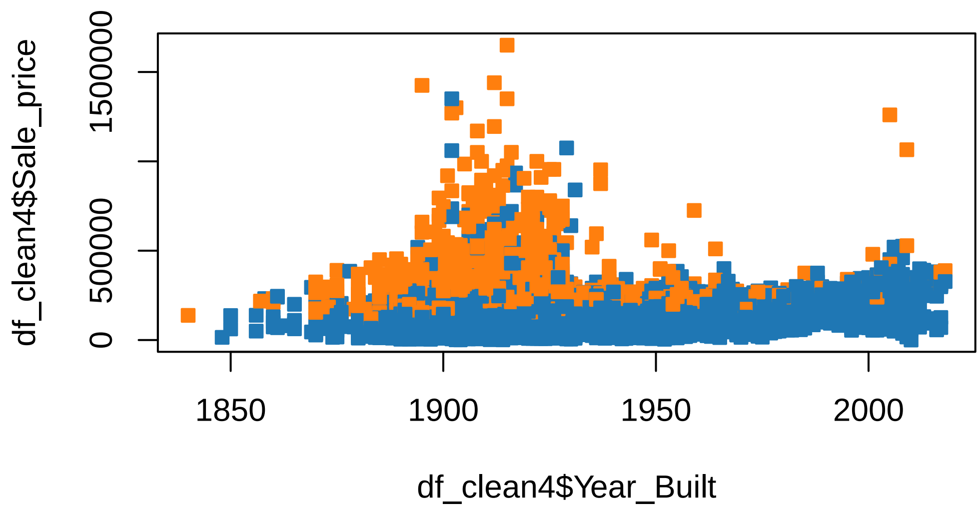

```
# abline(h=0)

custom_palette <- c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
  "#bcbd22", "#17becf", "#393b79",
  "#8c6d31", "#9c9ede", "#637939", "#eb348f"
)


plot(df_clean4$Year_Built,df_clean4$Sale_price,pch=22,bg=custom_palette[df_clean4$District
```




```
plot(df_clean4$Year_Built[!df_clean4$d_3],
     df_clean4$Sale_price[!df_clean4$d_3],
     pch=22,
     bg=custom_palette[df_clean4$District[!df_clean4$d_3]],col=custom_palette[df_clean4$Di
```




```
df_clean4$r_t=student_res2>quantile(student_res2,.9)
plot(df_clean4$Year_Built,df_clean4$Sale_price,
     pch=22,
     bg=custom_palette[df_clean4$r_t+1],
     col=custom_palette[df_clean4$r_t+1])
```




```
# plot(df_clean4$Year_Built,df_clean4$Sale_price,col=custom_palette[df_clean4$Extwall])
# for(colu in names(df_clean4)){
#
#   if (is.factor(df_clean2[1,colu])){
#     print(colu)
#     print(table(df_clean2[,colu]))
#     df_clean2[,colu]=droplevels(df_clean2[,colu])
#   }
# }




# df_clean4$d_3or14=mapply('||',df_clean4$District==3,df_clean4$District==14)
model2=lm(sqrt(Sale_price)~District +  Extwall +      Stories   + Year_Built+    District*Yea
            Fbath   +     log(Lotsize)    + Sale_date +District* log(Lotsize),df_clean4)




# model2=lm(Sale_price^(0.4)~.,df_clean3)
```

```r
summ2=summary(model2); summ2
```

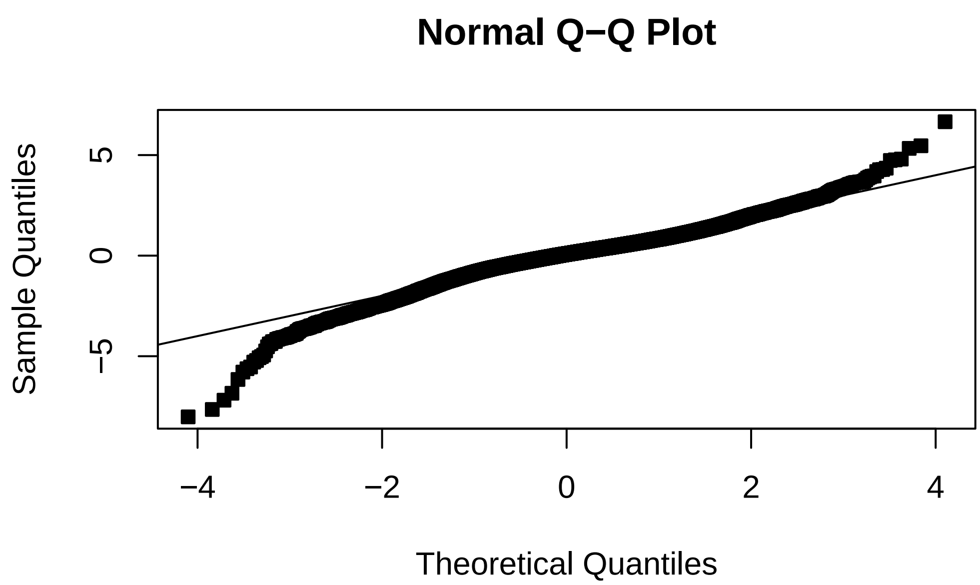
```
Call:
lm(formula = sqrt(Sale_price) ~ District + Extwall + Stories +
    Year_Built + District * Year_Built + Fin_sqft + Units + Bdrms +
    Fbath + log(Lotsize) + Sale_date + District * log(Lotsize),
    data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max
-547.36  -35.61    5.89   41.47  474.14

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.396e+03  9.630e+01  14.494  < 2e-16 ***
District               -2.441e+02  9.158e+00 -26.656  < 2e-16 ***
ExtwallBlock           -6.108e+00  6.044e+00  -1.011 0.312259
ExtwallBrick            1.146e+01  1.183e+00   9.684  < 2e-16 ***
ExtwallFiber-Cement     2.442e+01  6.115e+00   3.993 6.55e-05 ***
ExtwallFrame           -1.069e+01  1.599e+00  -6.682 2.41e-11 ***
ExtwallMasonry / Frame  8.342e+00  2.797e+00   2.983 0.002858 **
ExtwallPrem Wood        1.492e+01  9.195e+00   1.622 0.104761
ExtwallStone            9.094e+00  2.497e+00   3.641 0.000272 ***
ExtwallStucco           1.537e+01  3.511e+00   4.378 1.20e-05 ***
Stories1                4.059e+01  1.680e+01   2.416 0.015681 *
Stories1.5              5.480e+01  1.677e+01   3.268 0.001084 **
Stories2                6.143e+01  1.670e+01   3.678 0.000236 ***
Year_Built             -9.488e-01  5.316e-02 -17.847  < 2e-16 ***
Fin_sqft                9.479e-02  1.565e-03  60.560  < 2e-16 ***
Units1                  1.238e+02  1.200e+01  10.312  < 2e-16 ***
Units2                  2.178e+01  1.201e+01   1.813 0.069845 .
Units3                 -2.191e+01  1.293e+01  -1.695 0.090108 .
Bdrms0                  1.493e+02  3.025e+01   4.936 8.03e-07 ***
Bdrms1                  1.229e+02  1.648e+01   7.455 9.28e-14 ***
Bdrms2                  1.344e+02  1.472e+01   9.133  < 2e-16 ***
Bdrms3                  1.393e+02  1.462e+01   9.524  < 2e-16 ***
Bdrms4                  1.213e+02  1.456e+01   8.327  < 2e-16 ***
Bdrms5                  1.187e+02  1.456e+01   8.151 3.78e-16 ***
Bdrms6                  9.863e+01  1.459e+01   6.762 1.39e-11 ***
Bdrms7                  6.771e+01  1.555e+01   4.353 1.35e-05 ***
```

```
Bdrms8                 9.538e+01  1.636e+01   5.829 5.65e-09 ***
Fbath0                -6.106e+01  2.227e+01  -2.742 0.006118 **
Fbath1                -3.703e+01  1.553e+01  -2.385 0.017096 *
Fbath2                -1.015e+01  1.543e+01  -0.658 0.510580
Fbath3                 3.187e+01  1.532e+01   2.080 0.037538 *
Fbath4                 6.184e+01  1.643e+01   3.764 0.000168 ***
log(Lotsize)           3.120e+01  3.416e+00   9.133  < 2e-16 ***
Sale_date              5.995e-03  4.263e-04  14.064  < 2e-16 ***
District:Year_Built    1.315e-01  5.309e-03  24.762  < 2e-16 ***
District:log(Lotsize) -8.262e-01  3.469e-01  -2.382 0.017230 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 71.77 on 24407 degrees of freedom
Multiple R-squared:  0.4635,    Adjusted R-squared:  0.4627
F-statistic: 602.4 on 35 and 24407 DF,  p-value: < 2.2e-16
```

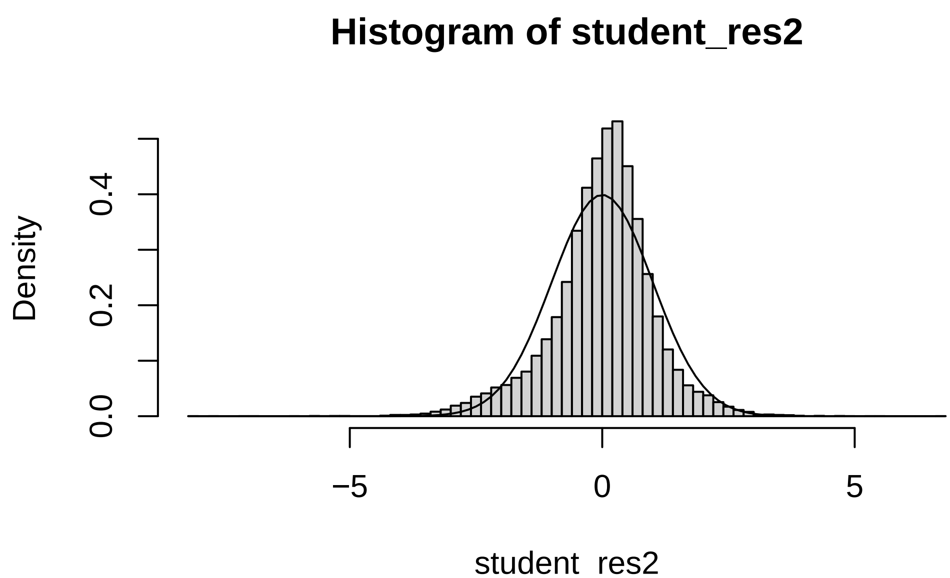
```r
summ2$adj.r.squared
```

```
[1] 0.4626948
```

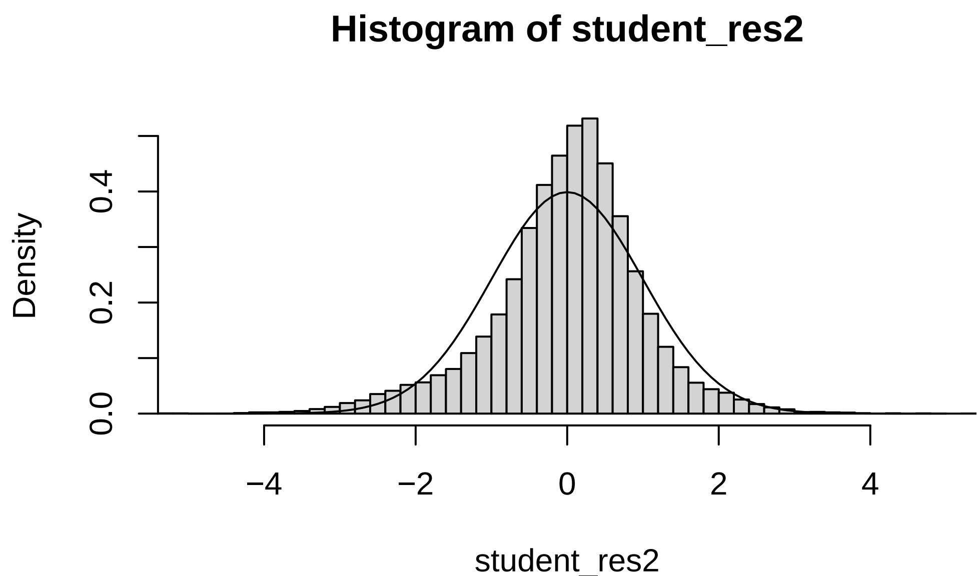
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



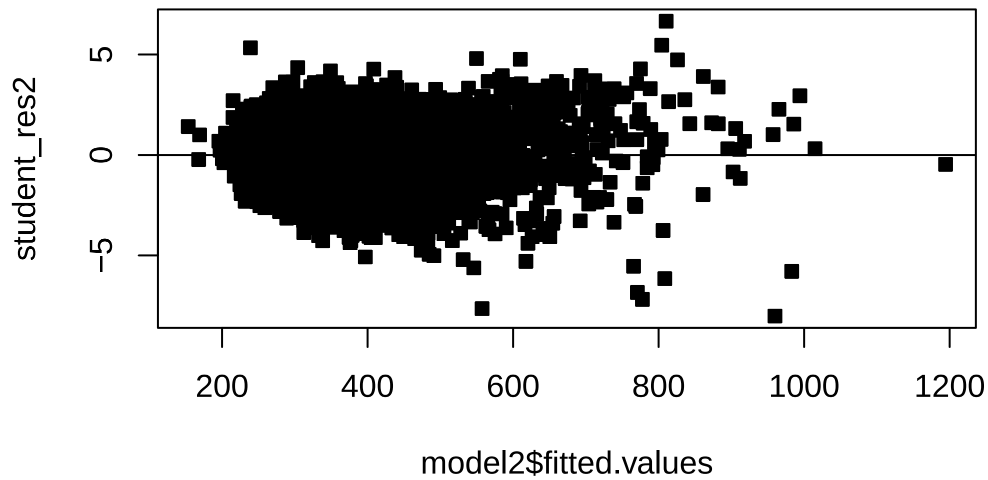

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



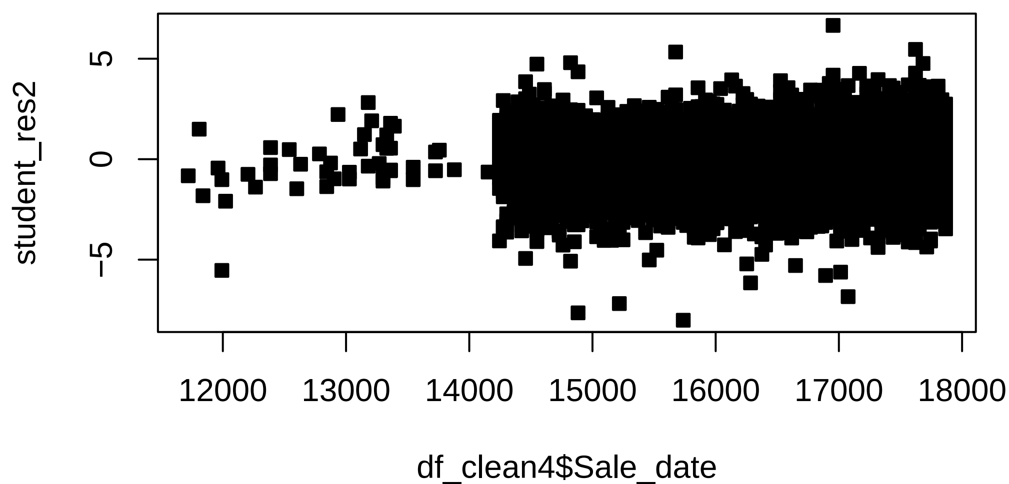
```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



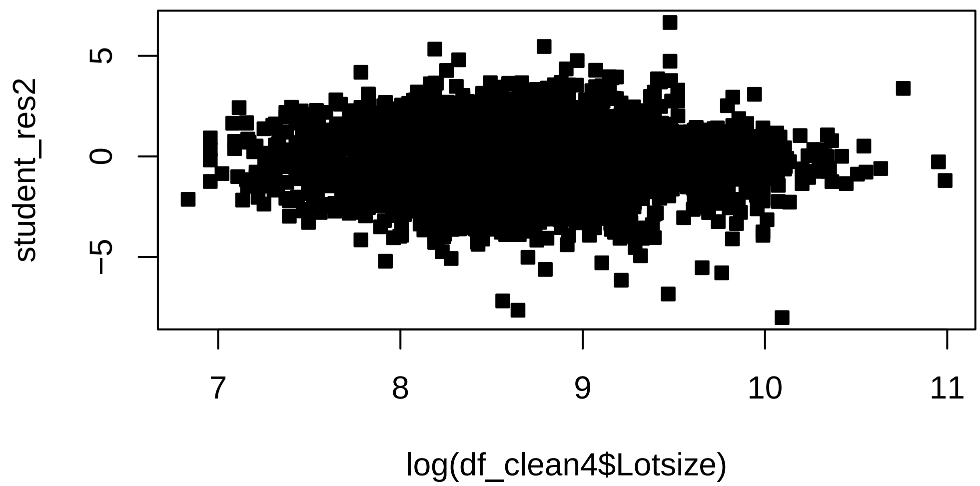

```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



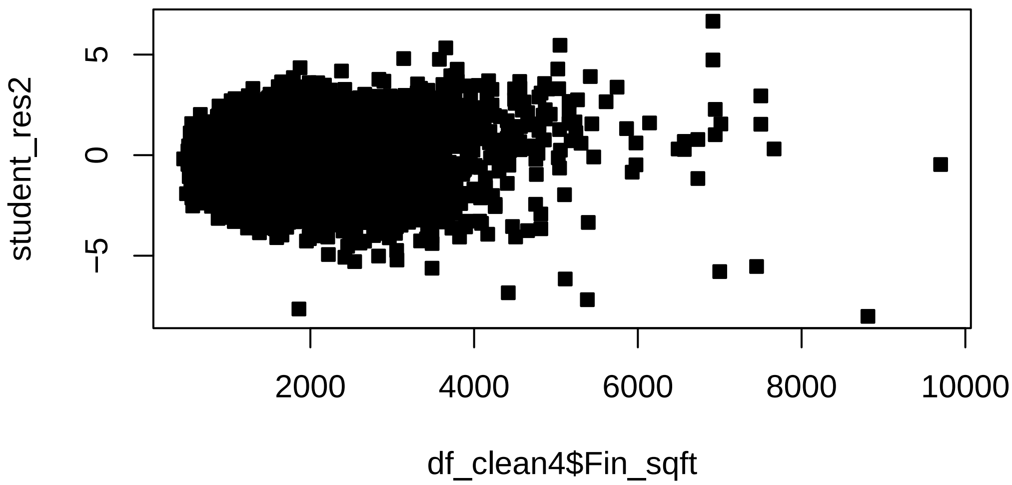

```r
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



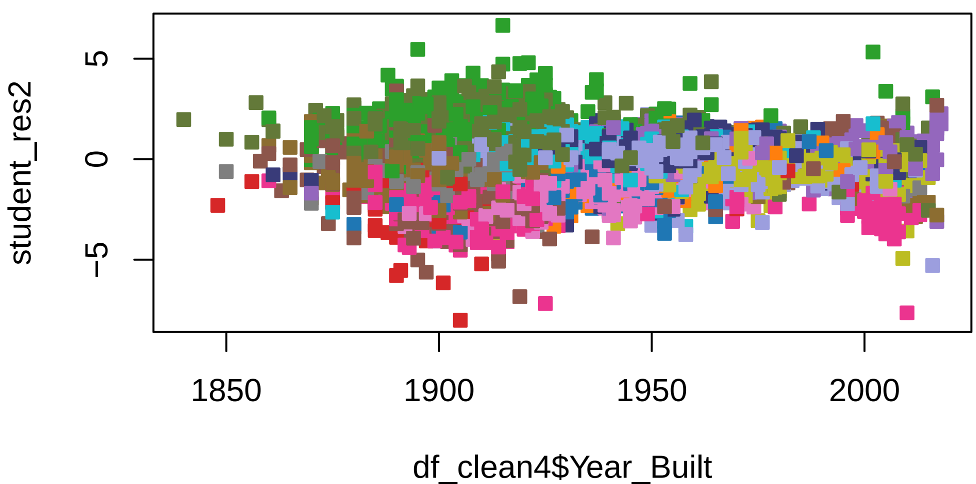
```r
plot(log(df_clean4$Lotsize) ,student_res2,pch=22,bg=1)
```




```r
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```




```
plot(df_clean4$Year_Built ,student_res2,pch=22,bg=custom_palette[df_clean4$District],col=c
```



```
df_clean4
```

	District	Extwall	Stories	Year_Built	Fin_sqft	Units	Bdrms	Fbath
1	7	Frame	2	1913	3476	>3	>8	1
2	3	Frame	2	1897	1992	>3	2	2
3	4	Frame	2	1907	2339	>3	0	1
4	4	Frame	2	1890	2329	>3	4	1
5	4	Stone	>2	1891	7450	2	7	>4
6	12	Frame	1.5	1906	2462	2	3	2
7	4	Frame	1.5	1890	2372	>3	2	2
8	11	Brick	1	1950	1149	1	3	1
9	1	Aluminum / Vinyl	1	1947	994	1	3	1
10	1	Stucco	2	1905	2938	>3	3	1
11	1	Brick	1	1951	1620	1	4	2
12	13	Brick	1	1956	986	1	3	1
13	3	Frame	2	1890	2360	>3	1	2
14	8	Aluminum / Vinyl	1	1903	1156	1	2	1
15	4	Frame	2	1895	3269	>3	6	1
16	15	Frame	2	1899	3380	>3	>8	1
17	4	Frame	2	1908	2775	>3	4	1
18	12	Frame	2	1889	2796	>3	0	2
19	12	Frame	2	1889	1930	>3	1	2
20	4	Frame	2	1913	2872	>3	3	2
21	13	Brick	2	1949	2430	>3	6	1
22	4	Frame	2	1890	3243	>3	8	1
23	6	Frame	2	1905	3346	>3	4	1
24	12	Frame	2	1900	2679	>3	7	2
25	12	Frame	2	1880	3459	>3	8	1
26	3	Stucco	2	1895	3208	>3	5	1
27	6	Block	2	1956	1920	>3	4	1
28	3	Stucco	2	1916	2800	>3	2	1
29	6	Brick	2	1908	2646	2	6	2
30	4	Frame	2	1908	3466	2	6	4
31	7	Frame	2	1913	3476	>3	>8	1
32	7	Frame	1	1932	761	1	3	1
33	6	Frame	2	1905	3346	>3	4	1
34	15	Frame	2	1899	4225	>3	>8	1
35	4	Frame	2	1908	3380	>3	4	1
36	4	Aluminum / Vinyl	2	1901	2898	2	6	4
37	8	Aluminum / Vinyl	2	1903	2880	>3	6	1
38	8	Aluminum / Vinyl	1.5	1892	2091	>3	4	1
39	4	Frame	2	1901	2770	>3	6	1

40	7	Frame	2	1913	2696	>3	4	1
41	12	Frame	2	1889	2796	>3	7	2
42	6	Block	2	1956	1920	>3	4	1
43	9	Aluminum / Vinyl	2	2007	2237	1	4	2
44	8	Aluminum / Vinyl	2	1903	3552	>3	7	1
45	8	Frame	2	1903	3552	2	7	4
46	2	Frame	1	1962	1169	1	3	2
47	1	Stone	1	1938	1188	1	2	1
48	1	Aluminum / Vinyl	1	1951	1054	1	3	1
49	1	Aluminum / Vinyl	1	1950	998	1	3	1
50	1	Aluminum / Vinyl	1	1951	988	1	3	1
51	1	Aluminum / Vinyl	2	2004	1470	1	3	2
52	1	Masonry / Frame	2	1949	2460	2	5	2
53	1	Brick	1.5	1900	4507	1	5	1
54	2	Brick	1	1954	1456	1	3	1
55	2	Aluminum / Vinyl	1.5	1962	1875	2	5	2
56	2	Aluminum / Vinyl	1	1966	1379	1	3	1
57	2	Brick	1	1962	1361	1	3	1
58	2	Brick	1	1964	1334	1	3	1
59	2	Brick	1	1962	1334	1	3	1
60	2	Brick	1	1959	1288	1	3	2
61	2	Frame	1	1970	1120	1	4	1
62	2	Frame	1	1955	864	1	3	1
63	2	Aluminum / Vinyl	1	1957	864	1	3	1
64	3	Stucco	>2	1894	3255	1	3	2
65	3	Frame	1	1910	977	1	3	1
66	3	Aluminum / Vinyl	2	1912	2152	2	4	2
67	3	Aluminum / Vinyl	1.5	1908	1656	2	4	2
68	3	Aluminum / Vinyl	1	1909	1359	1	3	2
69	3	Aluminum / Vinyl	2	1921	1248	1	3	1
70	5	Brick	1	1950	1338	1	3	1
71	5	Brick	1	1949	1284	1	3	1
72	5	Brick	1	1950	1201	1	2	1
73	5	Aluminum / Vinyl	1	1956	1176	1	4	2
74	5	Aluminum / Vinyl	1	1947	1165	1	4	1
75	5	Aluminum / Vinyl	1	1953	1144	1	3	1
76	5	Stone	1	1950	989	1	2	1
77	5	Masonry / Frame	2	1957	2234	2	6	2
78	5	Aluminum / Vinyl	1.5	1955	1995	2	5	2
79	5	Brick	1	1958	1110	1	3	1
80	6	Frame	1	1922	1029	1	2	1
81	7	Aluminum / Vinyl	1.5	1910	1729	2	6	2
82	8	Frame	2	1913	1364	1	3	1

83	8	Frame	1	1901	1256	1	3	1
85	9	Frame	2	1965	1591	1	4	1
86	9	Aluminum / Vinyl	2	1980	1455	1	3	1
87	10	Brick	1.5	1952	1435	1	2	1
88	10	Brick	1.5	1936	1276	1	2	1
89	10	Aluminum / Vinyl	1	1948	1012	1	3	1
90	10	Brick	2	1940	1556	1	4	1
91	10	Aluminum / Vinyl	1.5	1924	1697	2	3	2
92	10	Brick	1	1928	1853	1	4	1
93	10	Aluminum / Vinyl	1	1927	1025	1	3	1
94	10	Aluminum / Vinyl	2	1919	1452	1	4	1
95	10	Aluminum / Vinyl	1	1924	916	1	2	2
96	11	Brick	1.5	1941	1203	1	2	1
97	11	Aluminum / Vinyl	1	1938	1157	1	4	1
98	11	Aluminum / Vinyl	2	1992	2159	1	3	2
99	11	Brick	1.5	1937	1859	2	3	2
100	11	Brick	1	1932	1369	1	3	1
101	11	Aluminum / Vinyl	1	1961	1336	1	3	1
102	11	Aluminum / Vinyl	1	1979	1321	1	3	2
103	11	Brick	1	1958	1119	1	3	2
104	11	Brick	1	1953	1048	1	2	1
105	11	Aluminum / Vinyl	1	1949	870	1	2	1
106	13	Brick	1	1950	1188	1	3	2
107	13	Aluminum / Vinyl	1.5	1946	1122	1	2	1
108	13	Aluminum / Vinyl	1	1943	971	1	3	1
109	13	Brick	1	1966	1957	1	3	1
110	13	Brick	1	1967	1414	1	3	1
111	13	Aluminum / Vinyl	1	1959	1175	1	3	1
112	13	Brick	1	1964	1122	1	3	1
113	14	Aluminum / Vinyl	2	1954	1078	1	2	1
114	14	Aluminum / Vinyl	2	1910	2128	2	4	2
115	14	Aluminum / Vinyl	1	1924	973	1	3	1
117	14	Frame	1.5	1900	1340	1	3	2
118	14	Block	1.5	1920	1326	1	2	1
119	1	Brick	1.5	1938	1184	1	2	1
120	1	Aluminum / Vinyl	2	1929	2429	2	6	2
121	1	Brick	2	1943	1786	2	4	2
122	1	Aluminum / Vinyl	1	1923	1431	1	3	1
123	2	Brick	1	1955	1074	1	3	1
124	2	Aluminum / Vinyl	1	1956	1034	1	3	1
125	2	Aluminum / Vinyl	2	1959	1428	1	3	1
126	2	Aluminum / Vinyl	2	1960	2070	2	6	2
127	2	Brick	1	1955	1115	1	2	1

128	2	Aluminum / Vinyl	2	1970	2785	2	6	2
129	3	Brick	2	1946	2684	2	5	2
130	3	Frame	2	1887	3352	2	5	2
131	3	Brick	2	1907	2849	1	5	3
132	5	Brick	1.5	1952	1568	1	3	2
133	5	Stone	1	1947	1425	1	4	1
134	5	Stone	1	1952	1408	1	3	1
135	5	Aluminum / Vinyl	1	1951	1036	1	3	1
136	5	Aluminum / Vinyl	1	1949	978	1	4	1
137	5	Aluminum / Vinyl	2	1960	1686	1	3	1
138	5	Frame	2	1951	1540	1	3	1
139	5	Brick	1	1954	1282	1	3	1
140	5	Aluminum / Vinyl	1	1962	1268	1	3	1
141	5	Aluminum / Vinyl	1	1956	1132	1	3	1
142	5	Aluminum / Vinyl	1	1955	978	1	3	1
143	5	Masonry / Frame	2	1954	2498	3	5	3
144	6	Frame	2	1914	2425	2	6	2
145	6	Brick	2	1914	3087	2	5	2
146	6	Frame	1	1924	1696	1	4	1
147	7	Brick	1	1950	1598	1	4	2
148	7	Aluminum / Vinyl	1	1941	1272	1	3	1
149	7	Aluminum / Vinyl	1	1938	957	1	3	1
150	7	Frame	2	1920	2208	2	4	2
151	7	Brick	1	1960	1442	1	4	2
152	8	Aluminum / Vinyl	2	1921	2647	2	6	2
153	8	Brick	1	1927	1493	1	4	2
154	9	Brick	1	1958	1275	1	3	1
155	9	Aluminum / Vinyl	1	1936	1262	1	4	2
156	9	Aluminum / Vinyl	1	1972	1140	1	3	1
157	9	Aluminum / Vinyl	1	1976	906	1	3	1
158	9	Aluminum / Vinyl	1	1954	672	1	2	1
159	10	Aluminum / Vinyl	1	1940	1128	1	3	2
160	10	Brick	1.5	1936	1109	1	2	1
161	10	Aluminum / Vinyl	2	1988	2666	1	4	2
162	10	Frame	>2	1912	2600	1	4	2
163	10	Frame	1.5	1924	1665	2	3	2
164	10	Brick	1	1948	778	1	2	1
165	10	Brick	1	1926	1427	1	3	1
166	10	Stucco	1.5	1917	1398	1	3	1
167	11	Aluminum / Vinyl	1.5	1970	2032	1	5	1
168	11	Stone	1.5	1951	1940	1	3	2
169	11	Aluminum / Vinyl	1.5	1940	1197	1	3	1
170	11	Aluminum / Vinyl	1	1950	1176	1	3	1

171	11	Brick	1	1948	948	1	3	1
172	11	Aluminum / Vinyl	2	1953	1904	2	6	2
173	11	Aluminum / Vinyl	1	1943	1550	2	4	2
174	11	Brick	1	1954	1037	1	3	1
175	11	Aluminum / Vinyl	1	1928	1088	1	2	1
176	11	Brick	1.5	1937	1925	1	3	1
177	12	Aluminum / Vinyl	1	1919	1118	1	5	1
178	12	Aluminum / Vinyl	1	1921	1076	1	4	1
179	13	Aluminum / Vinyl	1	1947	1210	1	3	1
180	13	Aluminum / Vinyl	1	1940	1195	1	3	1
181	13	Brick	1	1955	1010	1	3	1
182	13	Aluminum / Vinyl	1	1939	1001	1	2	1
183	13	Aluminum / Vinyl	1	1979	1268	1	3	1
184	13	Brick	1	1967	1199	1	3	1
185	13	Aluminum / Vinyl	1	1966	1012	1	3	1
186	13	Aluminum / Vinyl	1	1953	880	1	2	1
187	13	Aluminum / Vinyl	1	1915	1424	1	3	0
188	13	Aluminum / Vinyl	1	1923	1238	1	4	1
189	14	Aluminum / Vinyl	1.5	1941	1346	1	4	1
190	14	Aluminum / Vinyl	2	1929	2695	2	4	2
191	14	Frame	2	1898	3148	1	4	3
192	14	Frame	1	1906	1896	1	4	2
193	14	Frame	2	1890	1636	1	4	2
194	14	Prem Wood	1	1920	1332	1	3	2
195	14	Fiber-Cement	1	1890	1274	1	3	2
196	14	Brick	1	1930	1196	1	3	1
197	15	Aluminum / Vinyl	2	1913	2502	2	6	2
199	15	Frame	1	1918	1434	1	3	1
200	15	Aluminum / Vinyl	1	1895	1362	1	4	1
201	1	Brick	1	1955	1259	1	4	1
202	1	Brick	2	1954	1940	2	4	2
203	1	Frame	1	1946	856	1	2	1
204	2	Brick	1	1956	1406	1	3	1
205	2	Stone	1	1955	1120	1	2	1
206	2	Aluminum / Vinyl	1	1957	951	1	3	1
207	2	Aluminum / Vinyl	2	1974	2484	2	6	2
208	3	Frame	2	1922	1852	1	3	1
209	3	Frame	1.5	1921	2101	1	4	2
210	3	Stucco	1	1915	1414	1	2	1
211	5	Brick	1.5	1953	2429	1	3	2
212	5	Brick	1	1951	1318	1	3	2
213	5	Brick	1.5	1949	1258	1	2	1
214	5	Aluminum / Vinyl	1	1950	1257	1	3	2

215	5	Aluminum / Vinyl	1	1952	934	1	3	1
216	5	Aluminum / Vinyl	1	1942	923	1	3	2
217	5	Aluminum / Vinyl	1	1956	784	1	2	1
218	5	Aluminum / Vinyl	2	1984	1686	1	3	2
219	5	Stone	2	1957	2146	2	6	2
220	5	Frame	2	1961	2092	2	6	2
221	5	Frame	1	1968	1500	1	3	1
222	5	Aluminum / Vinyl	1	1997	1262	1	3	1
223	5	Brick	1	1952	1239	1	3	1
224	5	Brick	1	1958	1232	1	3	1
225	5	Brick	1	1956	1223	1	3	1
226	5	Aluminum / Vinyl	1	1956	1154	1	3	1
227	5	Aluminum / Vinyl	1	1956	1077	1	3	1
228	5	Brick	1	1966	1059	1	3	1
229	5	Aluminum / Vinyl	1	1950	1036	1	3	1
230	5	Aluminum / Vinyl	1	1957	954	1	3	1
231	5	Frame	1	1948	833	1	2	1
232	5	Aluminum / Vinyl	2	1962	2484	2	6	2
233	5	Aluminum / Vinyl	2	1978	2451	2	6	2
234	6	Aluminum / Vinyl	2	1922	2086	2	4	2
235	6	Aluminum / Vinyl	1.5	1896	1728	2	4	2
236	6	Aluminum / Vinyl	1.5	1910	1468	2	4	2
237	6	Aluminum / Vinyl	1.5	1909	1740	1	3	2
238	6	Frame	1.5	1892	1366	1	3	1
239	6	Frame	1.5	1892	1366	1	3	1
240	6	Stone	1	1925	820	1	1	1
241	7	Masonry / Frame	2	1936	2780	2	4	2
242	7	Aluminum / Vinyl	1.5	1929	2082	2	4	2
243	7	Aluminum / Vinyl	1.5	1899	1452	2	4	1
244	7	Stone	1	1941	1118	1	2	1
245	7	Aluminum / Vinyl	1	1914	1515	1	5	1
246	8	Masonry / Frame	2	1912	2904	1	6	1
247	8	Aluminum / Vinyl	2	1883	2234	2	6	2
248	8	Aluminum / Vinyl	1.5	1904	1912	2	3	2
249	8	Aluminum / Vinyl	1	1916	1578	1	4	2
250	8	Aluminum / Vinyl	1.5	1912	1652	1	2	1
251	8	Aluminum / Vinyl	1	1910	1438	1	3	1
252	8	Aluminum / Vinyl	1	1886	1252	1	3	1
253	9	Aluminum / Vinyl	2	2008	2361	1	4	2
254	9	Aluminum / Vinyl	1	1971	1200	1	4	1
255	10	Aluminum / Vinyl	1	1950	1170	1	4	1
256	10	Aluminum / Vinyl	1.5	1930	1852	2	4	2
257	10	Brick	2	1950	1900	2	4	2

258	10	Aluminum / Vinyl	1	1926	1557	1	4	2
259	10	Frame	1	1919	1499	1	3	2
260	10	Aluminum / Vinyl	1	1925	1401	1	4	1
261	10	Frame	1	1930	1318	1	3	1
262	10	Aluminum / Vinyl	1	1927	1189	1	3	1
263	10	Aluminum / Vinyl	1	1926	1034	1	3	1
264	10	Aluminum / Vinyl	1	1940	864	1	2	1
265	11	Aluminum / Vinyl	1	1948	1188	1	3	1
266	11	Aluminum / Vinyl	1	1951	865	1	2	1
267	11	Stone	1	1948	1057	1	2	1
268	11	Frame	1	1953	998	1	3	1
269	11	Aluminum / Vinyl	1	1956	988	1	3	1
270	11	Brick	1	1950	869	1	2	1
271	11	Aluminum / Vinyl	1	1955	694	1	2	1
272	12	Aluminum / Vinyl	2	1894	1780	2	6	2
273	12	Frame	1	1924	1202	1	4	1
274	12	Aluminum / Vinyl	1	1895	1004	1	3	1
275	13	Stone	1.5	1937	1819	1	3	1
276	13	Aluminum / Vinyl	1	1941	1763	1	5	2
277	13	Masonry / Frame	2	1937	1494	1	4	1
278	13	Aluminum / Vinyl	1	1954	1476	1	3	2
279	13	Brick	1.5	1946	1406	1	2	1
280	13	Aluminum / Vinyl	1	1954	1034	1	3	1
281	13	Aluminum / Vinyl	1	1943	1006	1	3	1
282	13	Frame	2	1979	1746	1	3	2
283	13	Brick	2	1940	2076	2	4	2
284	13	Brick	1	1990	2044	1	3	2
285	13	Brick	1	1957	1308	1	4	1
286	13	Aluminum / Vinyl	1	1950	1269	1	2	1
287	13	Brick	1	1956	1102	1	2	1
288	13	Aluminum / Vinyl	1	1958	1056	1	3	1
289	13	Aluminum / Vinyl	1	1959	1051	1	3	1
290	13	Aluminum / Vinyl	1	1960	1048	1	3	1
291	13	Aluminum / Vinyl	1	1960	907	1	3	2
292	14	Brick	1.5	1956	2073	1	4	3
293	14	Stone	1	1950	1571	1	4	1
294	14	Brick	1	1947	926	1	3	1
295	14	Aluminum / Vinyl	1.5	1927	2109	2	4	3
296	14	Aluminum / Vinyl	2	1895	1968	1	5	2
297	14	Frame	2	1876	1674	2	3	2
298	14	Aluminum / Vinyl	1.5	1921	1391	2	3	2
299	14	Brick	1	1926	1548	1	4	1
300	14	Aluminum / Vinyl	1	1925	1430	1	3	1

301	14	Block	1	1947	858	1	2	1
302	14	Aluminum / Vinyl	1.5	1898	1611	1	3	2
303	14	Frame	1	1922	1334	1	4	2
304	14	Aluminum / Vinyl	1	1900	1220	1	3	2
305	15	Aluminum / Vinyl	2	1913	2902	2	6	2
306	15	Aluminum / Vinyl	1.5	1905	1596	2	3	2
307	15	Stucco	1	1916	1680	1	5	3
308	1	Aluminum / Vinyl	2	1939	1816	1	5	1
309	1	Aluminum / Vinyl	1.5	1929	1596	2	4	2
310	1	Aluminum / Vinyl	1.5	1929	1596	2	4	2
311	1	Frame	1.5	1925	1546	2	3	2
312	1	Aluminum / Vinyl	1	1927	1278	1	4	1
313	1	Aluminum / Vinyl	1	1953	1082	1	3	1
314	1	Aluminum / Vinyl	1	1926	1334	1	4	2
315	2	Aluminum / Vinyl	1	1956	1107	1	3	1
316	2	Aluminum / Vinyl	1	1942	1073	1	3	1
317	2	Aluminum / Vinyl	1	1951	862	1	2	1
318	2	Masonry / Frame	2	1956	2588	2	6	2
319	2	Aluminum / Vinyl	2	1957	2066	2	6	2
320	2	Aluminum / Vinyl	1	1965	1223	1	3	1
321	2	Aluminum / Vinyl	1	1963	1218	1	3	1
322	2	Brick	1	1947	1077	1	2	1
323	2	Aluminum / Vinyl	1	1957	988	1	3	1
324	3	Aluminum / Vinyl	2	1925	2848	2	6	2
325	3	Aluminum / Vinyl	2	1923	2724	2	6	2
326	3	Aluminum / Vinyl	1.5	1890	1570	2	2	2
327	3	Masonry / Frame	2	1909	3701	1	6	4
328	3	Frame	2	1909	2167	1	3	1
329	3	Aluminum / Vinyl	1	1890	1113	1	3	1
330	5	Stone	1	1948	1247	1	3	1
331	5	Brick	1	1949	1218	1	3	1
332	5	Aluminum / Vinyl	1	1952	1173	1	3	1
333	5	Aluminum / Vinyl	1	1952	1124	1	3	1
334	5	Aluminum / Vinyl	1.5	1941	1084	1	3	1
335	5	Frame	1.5	1941	1035	1	3	1
336	5	Aluminum / Vinyl	1	1951	954	1	3	1
337	5	Brick	1	1951	948	1	2	1
338	5	Aluminum / Vinyl	1	1952	722	1	2	1
339	5	Masonry / Frame	2	1972	2464	1	4	1
340	5	Aluminum / Vinyl	2	1954	1872	1	4	1
341	5	Aluminum / Vinyl	2	1988	1800	1	3	1
342	5	Aluminum / Vinyl	2	1948	1378	1	3	1
343	5	Brick	2	1966	2744	2	6	2

344	5	Masonry / Frame	2	1955	2192	2	6	2
345	5	Brick	1.5	1953	1930	2	4	2
346	5	Brick	1	1963	1446	1	3	1
347	5	Stone	1	1951	1280	1	2	2
348	5	Aluminum / Vinyl	1	1953	1263	1	3	1
349	5	Brick	1	1958	1219	1	3	1
350	5	Aluminum / Vinyl	1	1971	1217	1	4	1
351	5	Brick	1	1955	1200	1	3	1
352	5	Brick	1	1958	1076	1	3	2
353	5	Aluminum / Vinyl	1	1956	997	1	3	1
354	5	Aluminum / Vinyl	1	1958	963	1	3	2
355	5	Masonry / Frame	2	1967	2424	2	6	2
356	6	Frame	1.5	1902	1990	2	5	2
357	6	Brick	1	1928	1653	1	4	1
358	7	Stone	1	1935	1711	1	4	1
359	7	Brick	1.5	1937	1647	1	4	2
360	7	Brick	1	1948	1590	1	3	1
361	7	Stone	1	1947	1402	1	4	3
362	7	Aluminum / Vinyl	2	1923	2496	2	4	2
363	7	Stone	1.5	1943	1973	2	3	2
364	7	Aluminum / Vinyl	1.5	1929	1930	2	4	2
365	8	Frame	1	1885	652	1	2	1
366	8	Frame	2	1914	2540	2	6	2
368	9	Aluminum / Vinyl	1	1953	1020	1	4	1
369	9	Brick	1	1957	1217	1	3	2
370	9	Brick	1	1957	1188	1	3	1
371	9	Masonry / Frame	1	1957	1161	1	3	1
372	10	Aluminum / Vinyl	1.5	1947	1636	1	3	1
373	10	Brick	1	1945	946	1	3	2
374	10	Masonry / Frame	2	1926	2407	1	3	2
375	10	Aluminum / Vinyl	1	1900	1240	1	3	1
376	10	Stucco	1.5	1925	1759	2	3	2
377	10	Brick	1	1926	2011	1	3	1
378	10	Brick	1	1927	2005	1	4	2
379	10	Brick	1	1927	1733	1	3	1
380	10	Aluminum / Vinyl	1	1920	1620	1	4	1
381	10	Brick	1	1931	1557	1	3	1
382	10	Brick	1	1927	1214	1	2	1
383	10	Stone	1	1952	1728	1	3	1
384	10	Stone	1	1955	1289	1	3	1
385	10	Frame	1	1951	1202	1	3	1
386	10	Aluminum / Vinyl	1	1953	792	1	2	2
387	10	Brick	1	1948	788	1	2	1

388	10	Aluminum / Vinyl	1.5	1926	1223	1	4	1
389	10	Frame	1	1916	934	1	3	1
390	11	Aluminum / Vinyl	1	1941	1320	1	3	2
391	11	Aluminum / Vinyl	1	1950	1220	1	3	1
392	11	Aluminum / Vinyl	1	1954	1187	1	4	1
393	11	Aluminum / Vinyl	1	1953	1163	1	3	1
394	11	Brick	1	1958	1100	1	2	2
395	11	Brick	1	1949	863	1	2	1
396	11	Prem Wood	2	2002	2674	1	4	2
397	11	Aluminum / Vinyl	2	1949	1275	1	3	1
398	11	Brick	2	1957	2550	2	6	2
399	11	Aluminum / Vinyl	2	1957	1884	2	6	2
400	11	Frame	1	1927	1618	1	5	2
401	11	Frame	1	1954	1478	1	3	1
402	11	Frame	1	1964	1292	1	4	1
403	11	Aluminum / Vinyl	1	1959	1233	1	3	1
404	11	Brick	1	1954	1149	1	2	1
405	11	Aluminum / Vinyl	1	1961	1062	1	3	1
406	11	Aluminum / Vinyl	1	1954	972	1	3	1
407	11	Aluminum / Vinyl	1	1955	972	1	3	1
408	11	Frame	1	1951	958	1	2	1
409	11	Aluminum / Vinyl	1	1954	943	1	3	1
410	11	Aluminum / Vinyl	1	1954	906	1	3	1
411	11	Frame	1	1951	811	1	2	1
412	11	Aluminum / Vinyl	1	1956	672	1	2	1
413	13	Brick	1.5	1948	1897	1	4	1
414	13	Aluminum / Vinyl	1	1941	1365	1	4	1
415	13	Stone	1	1940	927	1	2	1
416	13	Brick	1	1939	891	1	1	1
417	13	Aluminum / Vinyl	2	1940	1869	1	3	1
418	13	Frame	1.5	1912	2151	2	6	2
419	13	Aluminum / Vinyl	1	1929	1035	1	3	1
420	13	Brick	1	1960	1541	1	3	1
421	13	Brick	1	1969	1534	1	2	2
422	13	Brick	1	1959	1257	1	3	1
423	13	Aluminum / Vinyl	1	1960	1254	1	3	1
424	13	Frame	1	1966	1176	1	3	1
425	13	Aluminum / Vinyl	1	1954	1116	1	3	1
426	13	Brick	1	1963	1092	1	3	2
427	13	Aluminum / Vinyl	1	1959	1021	1	3	1
428	13	Aluminum / Vinyl	1	1954	1008	1	3	1
429	13	Aluminum / Vinyl	1	1961	983	1	3	1
430	13	Aluminum / Vinyl	1	1950	912	1	3	1

431	13	Aluminum / Vinyl	1	1954	864	1	3	1
432	13	Aluminum / Vinyl	1	1955	672	1	2	1
433	14	Block	1	1944	1055	1	3	2
434	14	Aluminum / Vinyl	1.5	1940	1003	1	2	1
435	14	Frame	1	1948	782	1	2	1
436	14	Aluminum / Vinyl	2	1945	1360	1	3	1
437	14	Aluminum / Vinyl	2	1937	1056	1	2	1
438	14	Aluminum / Vinyl	1.5	1927	2059	2	4	2
439	14	Frame	1	1926	1412	1	3	1
440	14	Aluminum / Vinyl	1	1923	1251	1	4	1
441	14	Aluminum / Vinyl	1	1918	908	1	2	1
442	14	Aluminum / Vinyl	1	1942	848	1	2	1
443	14	Aluminum / Vinyl	2	1909	1956	1	4	1
444	14	Stucco	1.5	1916	1561	1	4	2
445	14	Aluminum / Vinyl	1.5	1909	1523	1	3	1
446	14	Aluminum / Vinyl	2	1901	1503	1	3	1
447	1	Brick	1.5	1937	1461	1	3	1
448	1	Frame	1	1956	1212	1	2	1
449	1	Frame	1	1941	1048	1	3	1
450	1	Aluminum / Vinyl	1	1942	920	1	3	1
451	1	Aluminum / Vinyl	1	1937	919	1	2	1
452	1	Brick	2	1954	1810	2	6	2
453	1	Brick	2	1943	1786	2	4	2
454	1	Aluminum / Vinyl	1	1956	1425	1	3	2
455	1	Aluminum / Vinyl	1	1927	876	1	3	1
456	1	Prem Wood	1.5	1931	2883	1	4	1
457	2	Brick	1.5	1952	1450	1	4	2
458	2	Brick	1	1956	1365	1	3	2
459	2	Aluminum / Vinyl	1.5	1963	1885	2	5	2
460	2	Aluminum / Vinyl	1	1952	969	1	2	1
461	2	Aluminum / Vinyl	1	1952	969	1	2	1
462	2	Aluminum / Vinyl	1	1952	969	1	2	1
463	2	Aluminum / Vinyl	1	1952	969	1	2	1
464	2	Aluminum / Vinyl	1	1971	936	1	3	1
465	2	Aluminum / Vinyl	1	1955	672	1	2	1
466	2	Aluminum / Vinyl	1	1930	2152	1	3	2
467	3	Frame	1	1951	1653	1	4	1
468	3	Brick	1.5	1923	2351	1	4	2
469	3	Brick	1.5	1923	2351	1	4	2
470	3	Masonry / Frame	2	1925	1984	1	4	1
471	3	Frame	2	1889	2478	2	7	2
472	3	Brick	2	1923	5438	1	5	4
473	3	Aluminum / Vinyl	1	1927	1530	1	4	2

474	3	Masonry / Frame	2	1915	3417	1	5	3
475	3	Frame	2	1899	2340	1	5	2
476	3	Aluminum / Vinyl	2	1890	2077	1	4	2
477	3	Stone	1.5	1925	1787	1	3	2
478	3	Frame	2	1901	1717	1	4	1
479	3	Frame	1	1910	1501	1	3	1
480	3	Frame	2	1904	1254	1	4	1
481	3	Aluminum / Vinyl	1	1895	1230	1	5	2
482	3	Aluminum / Vinyl	1	1923	1132	1	3	1
483	3	Aluminum / Vinyl	1.5	1897	1095	1	2	1
484	5	Stone	1	1938	1745	1	3	2
485	5	Brick	1	1952	1543	1	4	2
486	5	Brick	1	1947	1464	1	3	1
487	5	Brick	1	1951	1438	1	4	1
488	5	Brick	1.5	1949	1409	1	3	1
489	5	Aluminum / Vinyl	1	1940	1402	1	4	1
490	5	Aluminum / Vinyl	1	1952	1399	1	4	1
491	5	Brick	1	1940	1381	1	2	2
492	5	Aluminum / Vinyl	1	1952	1368	1	3	1
493	5	Aluminum / Vinyl	1	1948	1251	1	3	1
494	5	Aluminum / Vinyl	1	1950	1171	1	3	1
495	5	Aluminum / Vinyl	1	1950	1170	1	3	1
496	5	Aluminum / Vinyl	1	1950	1147	1	3	1
497	5	Aluminum / Vinyl	1	1956	1114	1	3	1
498	5	Aluminum / Vinyl	1	1953	1104	1	3	1
499	5	Aluminum / Vinyl	1	1950	1011	1	3	1
500	5	Aluminum / Vinyl	1	1955	994	1	3	2
501	5	Aluminum / Vinyl	1	1950	888	1	2	1
502	5	Brick	1	1947	883	1	2	1
503	5	Aluminum / Vinyl	1	1952	693	1	2	1
504	5	Aluminum / Vinyl	2	2008	2506	1	4	2
505	5	Masonry / Frame	2	1956	1598	1	3	1
506	5	Aluminum / Vinyl	2	1950	1572	1	3	1
507	5	Masonry / Frame	2	1938	1557	1	3	1
508	5	Stone	1	1952	1657	1	3	1
509	5	Brick	1	1958	1627	1	3	2
510	5	Aluminum / Vinyl	1	1959	1431	1	4	1
511	5	Brick	1	1959	1429	1	3	1
512	5	Aluminum / Vinyl	1	1993	1226	1	3	2
513	5	Brick	1	1956	1223	1	3	1
514	5	Brick	1	1957	1201	1	3	1
515	5	Brick	1	1959	1151	1	2	1
516	5	Aluminum / Vinyl	1	1954	1142	1	3	2

517	5	Aluminum / Vinyl	1	1957	1053	1	3	1
518	5	Frame	1	1951	1053	1	3	1
519	5	Brick	1	1951	1033	1	3	1
520	5	Brick	1	1954	1006	1	2	1
521	5	Aluminum / Vinyl	1	1956	983	1	3	1
522	5	Stone	1	1950	980	1	2	1
523	5	Frame	1	1954	962	1	3	1
524	6	Frame	2	1904	3476	2	6	2
525	6	Aluminum / Vinyl	1.5	1919	1969	1	4	2
526	6	Aluminum / Vinyl	1.5	1910	1210	1	3	1
527	6	Frame	1	1903	998	1	3	1
528	7	Block	1	1944	1014	1	3	1
529	7	Aluminum / Vinyl	2	1924	2674	2	6	2
530	7	Aluminum / Vinyl	1	1924	1497	1	4	1
531	7	Frame	1	1926	1453	1	5	1
532	7	Masonry / Frame	1	1946	1035	1	3	1
533	7	Aluminum / Vinyl	1	1947	1029	1	3	1
534	7	Masonry / Frame	2	1936	1678	1	3	1
535	8	Aluminum / Vinyl	2	2008	1672	1	3	2
536	8	Frame	1.5	1918	1809	2	3	2
537	8	Aluminum / Vinyl	1.5	1926	2095	2	5	2
538	8	Aluminum / Vinyl	1	1927	1601	1	5	1
539	8	Frame	2	1913	1364	1	3	1
540	8	Aluminum / Vinyl	1	1904	1161	1	3	1
541	8	Aluminum / Vinyl	1	1900	1074	1	3	1
542	8	Masonry / Frame	2	1914	3632	3	7	4
543	9	Aluminum / Vinyl	1	1980	1320	1	3	1
544	9	Brick	1	1959	924	1	3	1
545	9	Frame	1	1977	906	1	3	1
546	10	Stone	1.5	1937	1815	1	3	2
547	10	Masonry / Frame	1.5	1950	1505	1	3	1
548	10	Aluminum / Vinyl	1	1953	1307	1	3	1
549	10	Brick	1	1953	1255	1	3	1
550	10	Aluminum / Vinyl	1	1948	1094	1	3	1
551	10	Aluminum / Vinyl	1	1948	978	1	2	1
552	10	Aluminum / Vinyl	1	1951	920	1	2	1
553	10	Brick	1	1947	845	1	2	1
554	10	Stucco	2	1914	3591	2	6	3
555	10	Stone	2	1948	2520	2	4	2
556	10	Brick	1	1931	1741	1	4	2
557	10	Frame	1	1927	1584	1	3	2
558	10	Aluminum / Vinyl	1	1925	1225	1	3	1
559	10	Aluminum / Vinyl	1	1927	1010	1	2	1

560	10	Aluminum / Vinyl	1	1925	768	1	1	1
561	10	Aluminum / Vinyl	1	1955	1063	1	3	2
562	10	Brick	1	1957	1032	1	3	1
563	10	Aluminum / Vinyl	1	1943	810	1	2	1
564	10	Aluminum / Vinyl	2	1926	1320	1	3	2
565	11	Stone	1	1946	1278	1	2	1
566	11	Stone	1	1945	1182	1	4	1
567	11	Aluminum / Vinyl	1	1954	1103	1	4	1
568	11	Brick	1	1945	992	1	3	1
570	11	Aluminum / Vinyl	2	1964	1754	1	3	1
571	11	Stone	1.5	1937	2045	2	4	2
572	11	Brick	1	1952	1533	1	4	2
573	11	Frame	1	1960	1285	1	3	1
574	11	Aluminum / Vinyl	1	1958	1202	1	3	1
575	11	Aluminum / Vinyl	1	1959	1177	1	3	1
576	11	Brick	1	1951	1166	1	3	1
577	11	Brick	1	1956	1145	1	3	1
578	11	Aluminum / Vinyl	1	1960	1097	1	3	1
579	11	Brick	1	1958	1086	1	2	1
580	11	Aluminum / Vinyl	1	1959	1072	1	3	1
581	11	Brick	1	1956	1061	1	3	1
582	11	Brick	1	1954	1053	1	3	1
583	11	Brick	1	1956	1035	1	2	1
584	11	Brick	1	1950	1031	1	2	2
585	11	Brick	1	1950	1008	1	3	1
586	11	Aluminum / Vinyl	1	1954	955	1	3	1
587	11	Aluminum / Vinyl	1	1959	943	1	3	1
588	11	Brick	1	1955	925	1	2	1
589	11	Aluminum / Vinyl	1	1953	879	1	3	1
590	11	Aluminum / Vinyl	1	1953	879	1	3	1
591	11	Aluminum / Vinyl	1	1956	876	1	3	1
592	11	Frame	1	1952	802	1	2	1
593	11	Aluminum / Vinyl	1	1956	672	1	2	1
594	11	Aluminum / Vinyl	1	1936	1427	1	3	2
595	13	Frame	1.5	1986	2523	1	3	2
596	13	Aluminum / Vinyl	1.5	1939	1096	1	3	1
598	13	Aluminum / Vinyl	2	1929	1065	1	3	1
599	13	Aluminum / Vinyl	2	2000	2018	2	4	2
600	13	Brick	1	1982	1837	1	3	3
601	13	Brick	1	1971	1693	1	3	2
602	13	Brick	1	1959	1548	1	2	2
603	13	Aluminum / Vinyl	1	1967	1538	1	3	2
604	13	Aluminum / Vinyl	1	1975	1480	1	3	1

605	13	Aluminum / Vinyl	1	1984	1351	1	3	1
606	13	Aluminum / Vinyl	1	1950	1186	1	2	1
607	13	Aluminum / Vinyl	1	1959	1171	1	3	1
608	13	Aluminum / Vinyl	1	1967	1151	1	3	1
609	13	Brick	1	1959	1150	1	3	1
610	13	Brick	1	1960	1150	1	3	1
611	13	Brick	1	1955	1119	1	3	2
612	13	Brick	1	1959	1101	1	3	1
613	13	Aluminum / Vinyl	1	1961	981	1	3	1
615	14	Frame	1	1948	1540	1	4	1
616	14	Stone	1	1936	1246	1	3	1
617	14	Brick	1.5	1936	1222	1	2	1
618	14	Brick	1	1955	1108	1	2	1
619	14	Aluminum / Vinyl	1	1947	1102	1	3	1
620	14	Aluminum / Vinyl	1	1950	1021	1	3	1
621	14	Aluminum / Vinyl	2	1948	1378	1	3	1
622	14	Aluminum / Vinyl	1	1928	1638	2	3	2
623	14	Brick	2	1951	2260	2	4	2
624	14	Aluminum / Vinyl	1.5	1898	2343	2	5	2
625	14	Aluminum / Vinyl	1.5	1910	1894	2	3	2
626	14	Aluminum / Vinyl	1.5	1898	1748	2	4	2
627	14	Aluminum / Vinyl	1.5	1927	1715	1	4	1
628	14	Frame	1	1926	1607	1	3	1
629	14	Aluminum / Vinyl	1	1928	1477	1	3	1
630	14	Aluminum / Vinyl	1	1922	1164	1	3	1
631	14	Aluminum / Vinyl	1	1948	864	1	3	1
632	14	Aluminum / Vinyl	1	1950	808	1	2	1
633	14	Aluminum / Vinyl	1	1898	1449	1	3	2
634	14	Aluminum / Vinyl	2	1892	1240	1	2	1
635	14	Aluminum / Vinyl	1	1912	954	1	3	1
636	14	Aluminum / Vinyl	1	1922	896	1	2	1
637	15	Brick	2	1905	1894	1	4	1
638	1	Block	1.5	1950	1425	1	4	2
639	1	Aluminum / Vinyl	1	1942	1110	1	3	1
640	1	Brick	1	1951	1070	1	3	1
641	1	Brick	2	1953	2016	2	6	2
642	1	Aluminum / Vinyl	2	1956	1938	2	6	2
643	1	Aluminum / Vinyl	1.5	1937	1459	2	3	2
644	1	Brick	1	1929	1673	1	4	1
645	1	Aluminum / Vinyl	1	1925	1632	1	5	1
646	1	Masonry / Frame	1	1961	1646	1	3	2
647	1	Brick	1	1956	1270	1	3	1
648	1	Brick	1	1962	1042	1	3	2

649	1	Aluminum / Vinyl	1	1955	1040	1	3	1
650	1	Frame	1.5	1913	1186	1	3	1
651	2	Aluminum / Vinyl	1	1997	1481	1	3	2
652	2	Brick	1.5	1955	1839	1	4	2
653	2	Brick	1	1954	1553	1	4	1
654	2	Aluminum / Vinyl	1	1947	1172	1	3	1
655	2	Brick	1	1956	1096	1	2	1
656	2	Aluminum / Vinyl	1	1940	1028	1	3	1
657	2	Aluminum / Vinyl	2	2001	2321	1	3	2
658	2	Aluminum / Vinyl	2	1948	1267	1	3	1
659	2	Brick	1	1970	1334	1	3	2
660	2	Brick	1	1960	1185	1	3	1
661	2	Brick	1	1958	1159	1	3	1
662	2	Brick	1	1953	1126	1	3	1
663	2	Aluminum / Vinyl	1	1956	1058	1	3	1
664	2	Aluminum / Vinyl	1	1961	1056	1	3	1
665	2	Aluminum / Vinyl	1	1956	948	1	3	1
666	2	Aluminum / Vinyl	1	1955	936	1	3	1
667	2	Aluminum / Vinyl	1	1952	761	1	2	1
668	3	Frame	1.5	1926	1880	2	3	2
669	3	Aluminum / Vinyl	1.5	1890	1789	2	4	2
670	3	Frame	2	1906	2640	1	4	2
671	3	Frame	2	1900	2138	1	3	1
672	3	Stucco	2	1915	1923	1	4	1
673	3	Aluminum / Vinyl	1.5	1897	1364	1	3	1
675	5	Masonry / Frame	1.5	1947	1668	1	3	2
676	5	Brick	1	1950	1530	1	3	1
677	5	Aluminum / Vinyl	1	1940	1515	1	3	1
678	5	Brick	1	1949	1428	1	3	2
679	5	Brick	1	1947	1370	1	3	1
680	5	Brick	1	1948	1367	1	3	1
681	5	Aluminum / Vinyl	1	1953	1317	1	3	1
682	5	Aluminum / Vinyl	1	1951	1232	1	3	1
683	5	Aluminum / Vinyl	1	1950	1229	1	3	1
684	5	Stone	1	1948	1188	1	2	1
685	5	Aluminum / Vinyl	1	1956	1180	1	4	2
686	5	Aluminum / Vinyl	1	1952	1180	1	3	1
687	5	Brick	1	1952	1125	1	4	1
688	5	Aluminum / Vinyl	1	1951	1107	1	3	1
689	5	Frame	1	1946	1090	1	3	2
690	5	Brick	1	1955	1088	1	2	1
691	5	Stone	1	1940	1057	1	2	1
692	5	Aluminum / Vinyl	1	1953	1030	1	3	1

693	5	Aluminum / Vinyl	1	1952	1000	1	3	1
694	5	Aluminum / Vinyl	1	1949	971	1	3	1
695	5	Fiber-Cement	2	2009	2150	1	4	2
696	5	Masonry / Frame	2	1937	1674	1	3	1
697	5	Aluminum / Vinyl	2	1948	2032	2	4	2
698	5	Aluminum / Vinyl	1.5	1963	1866	2	4	2
699	5	Brick	1	1967	1665	1	3	2
700	5	Brick	1	1956	1223	1	3	1
701	5	Aluminum / Vinyl	1	1972	1204	1	4	1
702	5	Aluminum / Vinyl	1	1949	1140	1	3	1
703	5	Brick	1	1964	1090	1	3	1
704	5	Brick	1	1957	1080	1	3	2
705	5	Brick	1	1951	1056	1	2	2
706	5	Aluminum / Vinyl	1	1950	1022	1	3	1
707	5	Brick	1	1951	1016	1	3	1
708	5	Frame	1	1956	988	1	3	1
709	5	Aluminum / Vinyl	1	1956	948	1	2	1
710	5	Aluminum / Vinyl	1	1950	904	1	3	1
711	5	Aluminum / Vinyl	1	1951	833	1	2	1
712	5	Aluminum / Vinyl	1	1950	707	1	2	1
713	5	Aluminum / Vinyl	1	1950	1099	1	3	1
714	5	Masonry / Frame	2	1951	2040	2	4	2
715	6	Aluminum / Vinyl	2	1984	2052	1	4	2
716	6	Aluminum / Vinyl	2	2008	1848	1	3	1
717	6	Frame	1.5	1926	2000	2	4	2
718	6	Aluminum / Vinyl	2	1890	1998	2	4	2
720	6	Frame	1	1885	2212	1	3	1
721	7	Brick	1.5	1945	1486	1	3	1
722	7	Aluminum / Vinyl	1	1948	1099	1	3	1
723	7	Aluminum / Vinyl	1	1948	1063	1	3	1
724	7	Aluminum / Vinyl	2	1941	1346	1	3	1
725	7	Aluminum / Vinyl	2	1924	2535	2	6	2
726	7	Brick	1.5	1919	2391	2	4	2
727	7	Brick	2	1950	2394	2	4	2
728	7	Aluminum / Vinyl	1.5	1931	1279	1	3	1
729	7	Stone	1	1933	1244	1	3	2
730	8	Aluminum / Vinyl	2	1911	3331	2	7	3
731	8	Aluminum / Vinyl	1	1919	1512	1	4	1
732	8	Aluminum / Vinyl	1	1926	1427	1	3	1
733	8	Aluminum / Vinyl	1	1921	1325	1	3	1
734	8	Aluminum / Vinyl	1	1959	917	1	3	1
735	8	Aluminum / Vinyl	1.5	1910	1294	1	2	1
736	8	Aluminum / Vinyl	2	1912	2438	3	6	3

737	9	Aluminum / Vinyl	1	1955	1274	1	4	2
738	9	Aluminum / Vinyl	2	2004	2016	1	3	2
739	9	Aluminum / Vinyl	1	1930	1089	1	3	1
740	9	Aluminum / Vinyl	1	1948	720	1	2	1
741	9	Aluminum / Vinyl	1	2008	1721	1	3	2
742	9	Aluminum / Vinyl	1	1957	1211	1	3	1
743	9	Aluminum / Vinyl	1	1971	1200	1	4	1
744	9	Aluminum / Vinyl	1	1972	1200	1	4	1
745	9	Aluminum / Vinyl	1	1962	1132	1	3	2
746	9	Brick	1	1961	1117	1	3	1
747	9	Aluminum / Vinyl	1	1981	1077	1	2	2
748	9	Aluminum / Vinyl	1	1969	1063	1	3	1
749	9	Frame	1	1956	1033	1	3	1
750	9	Aluminum / Vinyl	1	1965	1006	1	3	1
751	9	Aluminum / Vinyl	1	1954	958	1	3	1
752	9	Frame	1	1980	912	1	3	2
753	9	Aluminum / Vinyl	1	1977	1512	1	3	1
754	10	Aluminum / Vinyl	1	1951	1292	1	3	2
755	10	Brick	1	1951	1536	1	3	1
756	10	Brick	1.5	1952	1435	1	2	1
757	10	Brick	1	1955	1344	1	3	1
758	10	Brick	1	1948	1336	1	3	1
759	10	Stone	1	1939	1335	1	3	1
760	10	Aluminum / Vinyl	1	1948	1206	1	3	1
761	10	Aluminum / Vinyl	1	1950	1196	1	3	1
762	10	Aluminum / Vinyl	1	1953	1184	1	2	1
763	10	Aluminum / Vinyl	1	1948	1170	1	4	1
764	10	Brick	1.5	1946	1060	1	2	1
765	10	Aluminum / Vinyl	1	1948	1047	1	3	1
766	10	Aluminum / Vinyl	1	1952	784	1	2	1
767	10	Brick	2	1928	2900	2	6	2
768	10	Aluminum / Vinyl	2	1917	2190	2	4	2
769	10	Aluminum / Vinyl	2	1925	1990	2	4	2
770	10	Frame	1.5	1913	1980	2	3	2
771	10	Masonry / Frame	2	1925	2698	2	4	2
772	10	Stone	2	1939	2183	2	5	2
773	10	Brick	1	1928	2116	1	5	2
774	10	Frame	1	1924	1637	1	4	1
775	10	Frame	1	1927	1316	1	3	2
776	10	Aluminum / Vinyl	1	1926	934	1	3	1
777	10	Frame	1	1927	926	1	2	1
778	10	Aluminum / Vinyl	1	1953	1384	1	3	1
779	10	Brick	1	1954	1303	1	3	1

780	10	Aluminum / Vinyl	1	1953	955	1	2	1
781	10	Aluminum / Vinyl	2	1900	1769	1	3	2
782	10	Frame	1	1923	1140	1	3	1
783	11	Aluminum / Vinyl	1	1962	1788	1	4	2
784	11	Brick	1	1939	1581	1	3	1
785	11	Aluminum / Vinyl	1.5	1954	1347	1	4	1
786	11	Brick	1	1954	1248	1	3	1
787	11	Aluminum / Vinyl	1	1955	1181	1	3	1
788	11	Aluminum / Vinyl	1	1947	1063	1	2	1
789	11	Aluminum / Vinyl	1	1950	1060	1	3	1
790	11	Brick	1	1951	1017	1	2	1
791	11	Aluminum / Vinyl	1	1949	925	1	3	1
792	11	Masonry / Frame	2	1951	2383	1	3	2
793	11	Aluminum / Vinyl	2	1987	2331	1	3	2
794	11	Frame	2	1984	2034	1	3	2
795	11	Masonry / Frame	2	1955	1846	1	4	2
796	11	Brick	1	1929	1865	1	3	1
797	11	Aluminum / Vinyl	1	1954	1436	1	2	1
798	11	Stone	1	1955	1407	1	3	2
799	11	Aluminum / Vinyl	1	1954	1183	1	3	1
800	11	Brick	1	1955	1153	1	3	1
801	11	Brick	1	1956	1150	1	3	1
802	11	Frame	1	1955	1147	1	3	1
803	11	Aluminum / Vinyl	1	1964	1094	1	3	1
804	11	Aluminum / Vinyl	1	1959	1073	1	3	1
805	11	Brick	1	1954	1013	1	3	1
806	11	Brick	1	1954	1013	1	3	1
807	11	Brick	1	1954	1010	1	3	1
808	11	Brick	1	1951	1008	1	3	1
809	11	Aluminum / Vinyl	1	1954	1001	1	3	1
810	11	Brick	1	1955	977	1	3	1
811	11	Aluminum / Vinyl	1	1954	941	1	3	1
812	11	Brick	1	1944	824	1	2	1
813	11	Aluminum / Vinyl	1	1952	778	1	2	1
814	11	Prem Wood	1	1964	2416	1	4	2
815	11	Aluminum / Vinyl	2	1975	2478	2	6	2
816	11	Aluminum / Vinyl	2	1963	2214	2	6	2
817	12	Aluminum / Vinyl	2	1966	1162	1	3	1
818	12	Aluminum / Vinyl	1	1903	828	1	2	1
819	12	Aluminum / Vinyl	2	1908	2316	2	6	2
820	12	Frame	1.5	1893	1728	2	4	2
821	13	Brick	1	1955	2184	1	5	2
822	13	Aluminum / Vinyl	1	1958	1408	1	4	1

823	13	Brick	1	1957	1026	1	2	1
824	13	Brick	1.5	1940	986	1	2	1
825	13	Aluminum / Vinyl	1	1923	877	1	2	2
826	13	Brick	1	1956	1829	1	4	2
827	13	Frame	1	1965	1555	1	4	1
828	13	Brick	1	1967	1427	1	4	1
829	13	Brick	1	1964	1289	1	3	1
830	13	Brick	1	1957	1168	1	3	1
831	13	Brick	1	1956	1150	1	2	2
832	13	Aluminum / Vinyl	1	1965	1124	1	3	1
833	13	Aluminum / Vinyl	1	1958	1120	1	3	1
834	13	Brick	1	1963	1116	1	3	1
835	13	Aluminum / Vinyl	1	1982	1064	1	3	1
836	13	Aluminum / Vinyl	1	1959	1053	1	3	1
837	13	Aluminum / Vinyl	1	1959	1029	1	3	1
838	13	Brick	1	1960	1025	1	3	1
839	13	Aluminum / Vinyl	1	1952	1016	1	3	1
840	13	Aluminum / Vinyl	1	1951	864	1	3	1
841	13	Aluminum / Vinyl	1	1951	783	1	2	1
842	13	Aluminum / Vinyl	1	1955	1273	1	2	1
843	14	Aluminum / Vinyl	1.5	1954	1409	1	4	1
844	14	Stucco	1	1944	1198	1	3	1
845	14	Brick	1	1948	1153	1	3	2
846	14	Aluminum / Vinyl	2	1954	1553	1	3	1
847	14	Aluminum / Vinyl	2	1938	1056	1	2	1
848	14	Aluminum / Vinyl	1.5	1927	2093	2	3	2
849	14	Aluminum / Vinyl	2	1923	2067	2	4	2
850	14	Aluminum / Vinyl	2	1905	3278	2	6	2
851	14	Aluminum / Vinyl	1.5	1906	2488	2	4	2
852	14	Aluminum / Vinyl	2	1884	2250	2	6	3
853	14	Aluminum / Vinyl	1.5	1890	1768	2	3	2
854	14	Aluminum / Vinyl	1.5	1908	1603	2	4	2
855	14	Aluminum / Vinyl	1.5	1922	1363	2	4	2
856	14	Frame	1	1933	1444	1	3	1
857	14	Brick	1	1956	1099	1	2	1
858	14	Brick	1	1959	980	1	2	1
859	14	Aluminum / Vinyl	1	1950	912	1	3	1
860	14	Brick	1	1953	892	1	2	1
861	14	Aluminum / Vinyl	1	1950	672	1	2	1
862	14	Aluminum / Vinyl	1.5	1890	2009	1	4	1
863	14	Frame	1.5	1890	1541	1	4	2
864	14	Aluminum / Vinyl	2	1925	1525	1	3	1
865	14	Aluminum / Vinyl	1	1910	1518	1	5	1

866	14	Aluminum / Vinyl	2	1909	1476	1	3	2
867	14	Aluminum / Vinyl	1.5	1905	1366	1	2	1
868	14	Aluminum / Vinyl	2	1909	1267	1	3	1
869	14	Aluminum / Vinyl	1.5	1920	1238	1	4	2
870	14	Stucco	1	1916	1135	1	2	1
871	14	Aluminum / Vinyl	1	1912	1070	1	3	1
872	14	Aluminum / Vinyl	1	1902	968	1	2	1
873	14	Aluminum / Vinyl	1	1908	892	1	3	1
874	14	Brick	1.5	1929	1639	1	3	1
875	15	Aluminum / Vinyl	2	2006	1458	1	3	1
876	15	Frame	2	1915	2616	2	6	2
877	15	Aluminum / Vinyl	2	1900	2009	2	4	2
878	15	Frame	1	1918	1835	1	5	1
879	15	Aluminum / Vinyl	1	1918	1506	1	5	1
880	1	Aluminum / Vinyl	1	1950	1071	1	3	1
881	1	Frame	2	1940	1186	1	3	1
882	1	Stone	1.5	1942	2168	2	3	2
883	1	Aluminum / Vinyl	1	1926	1169	1	3	1
884	1	Aluminum / Vinyl	1	1980	1619	1	3	2
885	1	Aluminum / Vinyl	1	1955	1082	1	3	1
886	1	Aluminum / Vinyl	1	1955	1082	1	3	1
887	1	Aluminum / Vinyl	1	1952	765	1	2	1
888	2	Aluminum / Vinyl	1	1951	1125	1	4	1
889	2	Aluminum / Vinyl	1	1962	1260	1	3	1
890	2	Brick	1	1955	1160	1	3	2
891	2	Brick	1	1956	1082	1	3	1
892	2	Brick	1	1955	1024	1	3	1
893	2	Frame	1	1961	1002	1	3	1
894	2	Aluminum / Vinyl	1	1955	950	1	3	1
895	2	Aluminum / Vinyl	1	1956	936	1	3	1
896	2	Aluminum / Vinyl	1	1957	874	1	3	1
897	2	Aluminum / Vinyl	1	1955	672	1	2	1
898	2	Aluminum / Vinyl	1	1955	1414	1	3	1
899	2	Masonry / Frame	2	1963	2281	2	6	2
900	2	Brick	1	1931	1436	1	4	1
901	3	Brick	2	1925	2934	2	4	2
902	3	Brick	1.5	1926	2555	2	5	2
903	3	Aluminum / Vinyl	1.5	1922	1641	2	3	2
904	3	Aluminum / Vinyl	1.5	1898	1846	2	5	2
905	3	Aluminum / Vinyl	1	1923	1919	1	4	2
906	3	Aluminum / Vinyl	2	1907	1993	1	3	1
907	3	Block	2	1905	1722	1	3	1
908	3	Aluminum / Vinyl	1	1889	1137	1	2	1

910	3	Brick	2	1914	3307	1	4	2
911	5	Aluminum / Vinyl	1	1941	1648	1	3	2
912	5	Aluminum / Vinyl	1	1952	1457	1	4	1
913	5	Frame	1	1950	1426	1	3	1
914	5	Brick	1	1955	1376	1	3	1
915	5	Brick	1	1949	1220	1	3	1
916	5	Aluminum / Vinyl	1	1948	1206	1	3	1
917	5	Brick	1.5	1947	1205	1	3	2
918	5	Aluminum / Vinyl	1	1952	1171	1	3	1
919	5	Stucco	1	1952	1156	1	3	2
920	5	Aluminum / Vinyl	1	1952	1153	1	3	1
921	5	Aluminum / Vinyl	1	1950	1132	1	3	1
922	5	Aluminum / Vinyl	1	1957	1130	1	4	1
923	5	Brick	1	1952	1088	1	3	1
924	5	Block	1	1940	1084	1	3	1
925	5	Aluminum / Vinyl	1	1950	1054	1	3	1
926	5	Aluminum / Vinyl	1	1951	1036	1	3	1
927	5	Brick	1	1953	1025	1	2	1
928	5	Aluminum / Vinyl	1	1950	1012	1	3	1
929	5	Aluminum / Vinyl	1	1942	981	1	3	1
930	5	Aluminum / Vinyl	1	1953	973	1	3	1
931	5	Aluminum / Vinyl	1	1950	971	1	3	1
932	5	Aluminum / Vinyl	1	1950	967	1	3	1
933	5	Aluminum / Vinyl	1	1953	963	1	3	1
934	5	Brick	1	1950	962	1	3	1
935	5	Aluminum / Vinyl	1	1950	691	1	2	1
936	5	Masonry / Frame	2	1958	2388	2	6	2
937	5	Masonry / Frame	2	1957	2330	2	6	2
938	5	Aluminum / Vinyl	1	1926	1345	1	4	1
939	5	Brick	1	1954	1744	1	2	1
940	5	Aluminum / Vinyl	1	1964	1399	1	3	1
941	5	Aluminum / Vinyl	1	1955	1361	1	3	1
942	5	Brick	1	1952	1200	1	3	2
943	5	Brick	1	1955	1161	1	3	1
944	5	Brick	1	1957	1134	1	3	1
945	5	Aluminum / Vinyl	1	1950	1130	1	3	2
946	5	Aluminum / Vinyl	1	1960	1120	1	3	1
947	5	Brick	1	1955	1068	1	2	1
948	5	Brick	1	1958	1040	1	3	2
949	5	Aluminum / Vinyl	1	1955	1022	1	3	1
950	5	Frame	1	1956	1008	1	3	1
951	5	Brick	1	1952	981	1	2	1
952	5	Stone	1	1950	980	1	2	1

953	5	Aluminum / Vinyl	1	1955	974	1	3	1
954	5	Frame	1	1955	974	1	3	1
955	5	Aluminum / Vinyl	1	1956	974	1	3	2
956	5	Aluminum / Vinyl	1	1955	948	1	3	1
957	5	Frame	1	1956	948	1	3	1
958	5	Aluminum / Vinyl	1	1957	907	1	2	1
959	5	Aluminum / Vinyl	1	1950	696	1	2	1
960	6	Aluminum / Vinyl	1	1895	798	1	2	1
961	6	Frame	2	1913	2778	2	6	2
962	6	Stucco	1.5	1914	1488	2	3	1
963	6	Aluminum / Vinyl	1.5	1925	1354	2	2	2
964	6	Frame	2	1900	2858	1	4	2
965	6	Aluminum / Vinyl	1	1900	1322	1	3	1
966	6	Frame	1	1900	1220	1	3	1
967	6	Frame	2	1912	2960	3	6	3
968	7	Stone	1	1948	1719	1	4	1
969	7	Masonry / Frame	2	1946	1455	1	3	1
970	7	Brick	2	1926	3128	2	6	2
971	7	Masonry / Frame	2	1937	2668	2	6	2
972	7	Frame	2	1912	1949	2	4	2
973	7	Brick	1	1926	1651	1	4	1
974	7	Brick	1	1952	1011	1	2	1
975	7	Aluminum / Vinyl	1.5	1915	1402	1	4	1
976	7	Aluminum / Vinyl	1	1916	1110	1	3	1
977	8	Aluminum / Vinyl	1	1885	1170	1	4	2
978	8	Brick	2	1908	2020	2	5	2
979	8	Aluminum / Vinyl	1.5	1898	1592	2	4	2
980	8	Aluminum / Vinyl	1	1890	1426	2	3	1
981	8	Aluminum / Vinyl	1	1896	1482	1	4	1
982	8	Frame	1	1898	1405	1	4	2
983	8	Aluminum / Vinyl	1	1920	1179	1	3	1
984	9	Aluminum / Vinyl	2	2001	2421	1	3	2
985	9	Aluminum / Vinyl	2	1978	1826	1	3	1
986	9	Aluminum / Vinyl	2	1996	1503	1	3	2
987	9	Frame	1	1938	711	1	2	1
988	9	Stone	1	1958	1750	1	3	1
989	9	Frame	1	1965	1227	1	3	1
990	9	Aluminum / Vinyl	1	1971	1200	1	3	1
991	9	Frame	1	1972	1200	1	4	1
992	9	Aluminum / Vinyl	1	1972	1120	1	3	1
993	9	Aluminum / Vinyl	1	1966	1107	1	3	1
994	9	Frame	1	1957	1080	1	3	1
995	9	Aluminum / Vinyl	1	1956	1064	1	3	1

996	9	Aluminum / Vinyl	1	1968	1063	1	3	1
997	9	Brick	1	1958	973	1	3	1
998	9	Brick	1	1958	927	1	3	1
999	9	Frame	1	1974	906	1	3	1
1000	10	Brick	1	1942	1624	1	3	1
1001	10	Aluminum / Vinyl	1	1952	1389	1	3	1
1002	10	Aluminum / Vinyl	1.5	1951	1353	1	3	2
1003	10	Brick	1	1947	1319	1	3	1
1004	10	Aluminum / Vinyl	1	1947	1275	1	3	1
1005	10	Frame	1	1952	1217	1	3	1
1006	10	Aluminum / Vinyl	1	1951	1119	1	3	1
1007	10	Aluminum / Vinyl	1	1947	1101	1	3	1
1008	10	Aluminum / Vinyl	1	1948	982	1	3	1
1009	10	Frame	2	1957	1980	1	3	2
1010	10	Aluminum / Vinyl	2	2004	1459	1	3	2
1011	10	Masonry / Frame	2	1926	1407	1	3	1
1012	10	Masonry / Frame	2	1929	2422	2	4	2
1013	10	Brick	1.5	1926	2365	2	4	2
1014	10	Aluminum / Vinyl	2	1928	2330	2	4	2
1015	10	Aluminum / Vinyl	1.5	1924	1830	2	4	2
1016	10	Aluminum / Vinyl	1.5	1929	1797	2	4	2
1017	10	Aluminum / Vinyl	2	1940	2012	2	4	2
1018	10	Brick	2	1952	1944	2	4	2
1019	10	Aluminum / Vinyl	2	1956	1872	2	4	2
1020	10	Aluminum / Vinyl	1	1926	1565	1	3	2
1021	10	Frame	1	1928	1452	1	4	1
1022	10	Frame	1	1928	1200	1	4	1
1023	10	Frame	1	1925	1188	1	2	1
1024	10	Aluminum / Vinyl	1	1927	1151	1	3	1
1025	10	Aluminum / Vinyl	1	1953	1384	1	3	1
1026	10	Aluminum / Vinyl	1.5	1904	1568	1	4	2
1027	10	Aluminum / Vinyl	1	1925	1448	1	3	1
1028	10	Aluminum / Vinyl	1	1905	1363	1	4	1
1029	10	Aluminum / Vinyl	1.5	1925	1215	1	3	1
1030	10	Aluminum / Vinyl	1	1910	936	1	2	2
1031	11	Stone	1	1938	1517	1	3	1
1032	11	Brick	1	1936	1412	1	3	1
1033	11	Stone	1	1942	1325	1	3	1
1034	11	Brick	1	1952	1264	1	2	1
1035	11	Brick	1	1955	1207	1	3	2
1036	11	Aluminum / Vinyl	1	1955	1193	1	4	1
1037	11	Brick	1	1953	1091	1	3	1
1038	11	Aluminum / Vinyl	1	1952	1034	1	3	1

1039	11	Aluminum / Vinyl	1	1952	983	1	3	1
1040	11	Aluminum / Vinyl	2	1954	1450	1	3	2
1041	11	Masonry / Frame	1.5	1964	2330	2	5	3
1042	11	Frame	1.5	1929	1904	2	3	2
1043	11	Aluminum / Vinyl	1	1959	1615	1	3	2
1044	11	Aluminum / Vinyl	1	1964	1334	1	3	1
1045	11	Brick	1	1957	1284	1	3	1
1046	11	Aluminum / Vinyl	1	1959	1257	1	3	1
1047	11	Block	1	1949	1244	1	2	1
1048	11	Brick	1	1958	1184	1	3	1
1049	11	Aluminum / Vinyl	1	1966	1174	1	3	1
1050	11	Brick	1	1956	1170	1	2	1
1051	11	Aluminum / Vinyl	1	1953	1158	1	3	1
1052	11	Brick	1	1955	1153	1	3	1
1053	11	Stone	1	1954	1138	1	2	1
1054	11	Aluminum / Vinyl	1	1957	1128	1	3	1
1055	11	Brick	1	1959	1120	1	3	1
1056	11	Brick	1	1954	1084	1	3	1
1057	11	Stone	1	1951	1059	1	2	1
1058	11	Brick	1	1958	1033	1	3	1
1059	11	Brick	1	1959	1031	1	3	1
1060	11	Brick	1	1959	1022	1	3	1
1061	11	Brick	1	1959	1022	1	3	1
1062	11	Aluminum / Vinyl	1	1956	1019	1	3	1
1063	11	Aluminum / Vinyl	1	1953	887	1	3	1
1064	11	Aluminum / Vinyl	1	1953	879	1	3	1
1065	11	Aluminum / Vinyl	1	1956	876	1	3	1
1066	11	Aluminum / Vinyl	1	1953	874	1	3	1
1067	11	Brick	1	1953	831	1	2	1
1068	11	Block	1	1945	742	1	2	1
1069	11	Brick	1	1969	1206	1	2	1
1070	11	Aluminum / Vinyl	2	1975	2478	2	6	2
1071	12	Aluminum / Vinyl	1	1900	1086	1	3	1
1072	12	Aluminum / Vinyl	1	1965	1104	1	3	1
1073	12	Brick	2	1922	1792	1	3	1
1074	12	Aluminum / Vinyl	1	1892	1597	1	4	1
1075	12	Aluminum / Vinyl	1	1905	1201	1	4	2
1077	13	Brick	1.5	1950	1504	1	3	2
1078	13	Stone	1	1947	1162	1	3	1
1079	13	Stucco	1	1939	1094	1	4	1
1080	13	Aluminum / Vinyl	1.5	1940	1002	1	3	1
1081	13	Brick	1	1954	883	1	2	1
1082	13	Aluminum / Vinyl	1.5	1949	1941	2	3	2

1083	13	Aluminum / Vinyl	2	1950	1502	2	4	2
1084	13	Aluminum / Vinyl	1	1926	1542	1	3	1
1085	13	Aluminum / Vinyl	1	1915	1338	1	3	1
1086	13	Aluminum / Vinyl	1	1925	1134	1	3	1
1087	13	Brick	1	1969	1837	1	3	1
1088	13	Brick	1	1960	1493	1	3	2
1089	13	Aluminum / Vinyl	1	1967	1433	1	3	1
1090	13	Brick	1	1959	1140	1	3	1
1091	13	Brick	1	1964	1116	1	3	2
1092	13	Aluminum / Vinyl	1	1981	1078	1	3	1
1093	13	Aluminum / Vinyl	1	1981	1066	1	3	1
1094	13	Aluminum / Vinyl	1	1961	1062	1	3	1
1095	13	Brick	1	1956	1051	1	3	1
1096	13	Aluminum / Vinyl	1	1960	972	1	3	1
1097	13	Aluminum / Vinyl	1	1959	956	1	3	1
1098	13	Aluminum / Vinyl	1	1960	942	1	3	1
1099	13	Aluminum / Vinyl	1	1953	900	1	3	1
1100	13	Brick	1	1955	811	1	2	1
1101	13	Frame	1	1953	745	1	2	1
1102	13	Aluminum / Vinyl	1	1947	672	1	2	1
1103	13	Aluminum / Vinyl	1	1932	1294	1	3	1
1104	14	Stone	1	1949	1407	1	2	2
1105	14	Aluminum / Vinyl	1	1947	1235	1	3	1
1106	14	Aluminum / Vinyl	1	1947	1107	1	3	1
1107	14	Aluminum / Vinyl	1	1953	1102	1	3	1
1108	14	Frame	1	1953	978	1	3	1
1109	14	Aluminum / Vinyl	1	1944	959	1	3	1
1110	14	Aluminum / Vinyl	1	1944	942	1	3	1
1111	14	Aluminum / Vinyl	1	1910	2015	1	3	1
1112	14	Aluminum / Vinyl	1	1898	810	1	2	1
1113	14	Aluminum / Vinyl	1	1898	810	1	2	1
1114	14	Aluminum / Vinyl	1	1895	796	1	2	1
1115	14	Aluminum / Vinyl	2	1926	2134	2	4	2
1116	14	Frame	1.5	1920	2062	2	4	2
1117	14	Aluminum / Vinyl	1.5	1922	1990	2	5	2
1118	14	Aluminum / Vinyl	1.5	1926	1943	2	5	2
1119	14	Stone	2	1939	2627	2	3	2
1120	14	Aluminum / Vinyl	2	1899	2525	2	6	2
1121	14	Aluminum / Vinyl	2	1892	1560	2	4	2
1122	14	Aluminum / Vinyl	2	1890	2548	2	5	2
1123	14	Brick	1	1930	2024	1	5	1
1124	14	Brick	1	1926	1711	1	4	1
1125	14	Brick	1	1929	1528	1	3	2

1126	14	Aluminum / Vinyl	1	1925	1340	1	3	1
1127	14	Brick	1	1968	1203	1	3	1
1128	14	Aluminum / Vinyl	1	1953	1056	1	2	1
1129	14	Aluminum / Vinyl	1	1931	866	1	2	1
1130	14	Aluminum / Vinyl	1	1947	791	1	2	1
1131	14	Masonry / Frame	2	1903	3896	1	5	0
1132	14	Stucco	2	1910	2248	1	5	2
1133	14	Aluminum / Vinyl	1	1914	1326	1	4	1
1134	14	Aluminum / Vinyl	1.5	1900	1057	1	3	1
1135	14	Frame	1.5	1925	765	1	1	1
1136	15	Aluminum / Vinyl	2	1996	1667	1	3	1
1137	15	Stucco	2	1914	2464	2	6	3
1138	15	Aluminum / Vinyl	1	1921	1397	1	3	1
1139	15	Frame	1	1890	949	1	3	1
1140	1	Aluminum / Vinyl	1.5	1938	1603	1	3	2
1141	1	Stucco	1	1950	1120	1	4	1
1142	1	Brick	1	1929	1420	1	3	1
1143	1	Aluminum / Vinyl	1	1929	1394	1	4	1
1144	1	Aluminum / Vinyl	1	1928	1205	1	4	1
1145	1	Aluminum / Vinyl	1	1929	798	1	2	1
1146	1	Aluminum / Vinyl	1	1958	1234	1	3	1
1147	1	Aluminum / Vinyl	1	1960	1151	1	3	1
1148	1	Aluminum / Vinyl	1	1951	720	1	2	1
1149	1	Aluminum / Vinyl	1	1951	713	1	2	1
1150	1	Stucco	1	1951	698	1	2	1
1151	1	Brick	1.5	1927	1105	1	2	1
1152	2	Aluminum / Vinyl	2	2003	2797	1	5	3
1153	2	Aluminum / Vinyl	2	2002	2706	1	4	2
1154	2	Stone	1	1952	1356	1	3	1
1155	2	Aluminum / Vinyl	1	1965	1223	1	3	1
1156	2	Aluminum / Vinyl	1	1966	1216	1	3	1
1157	2	Aluminum / Vinyl	1	1965	1149	1	3	1
1158	2	Aluminum / Vinyl	1	1960	1027	1	3	1
1159	2	Aluminum / Vinyl	1	1956	1019	1	3	1
1160	2	Brick	1	1957	947	1	2	1
1161	2	Aluminum / Vinyl	1	1958	936	1	3	1
1162	2	Aluminum / Vinyl	1	1957	909	1	3	1
1163	2	Aluminum / Vinyl	1	1957	903	1	3	1
1164	2	Aluminum / Vinyl	1	1957	873	1	2	1
1165	2	Frame	1	1955	864	1	3	1
1166	3	Brick	2	1935	2865	1	5	3
1167	3	Aluminum / Vinyl	2	1895	1806	2	4	2
1168	3	Frame	1	1916	1993	1	5	2

1169	3	Frame	2	1903	1804	1	5	1
1170	3	Aluminum / Vinyl	2	1889	2513	2	6	2
1171	5	Brick	1	1951	1642	1	3	2
1172	5	Stone	1	1951	1518	1	2	1
1173	5	Brick	1	1950	1432	1	3	1
1174	5	Brick	1.5	1937	1364	1	4	1
1175	5	Brick	1	1949	1359	1	3	1
1176	5	Aluminum / Vinyl	1	1953	1346	1	3	1
1177	5	Aluminum / Vinyl	1	1948	1341	1	4	1
1178	5	Brick	1	1951	1278	1	3	1
1179	5	Brick	1	1951	1278	1	3	1
1180	5	Frame	1	1952	1264	1	3	2
1181	5	Aluminum / Vinyl	1.5	1949	1238	1	4	1
1182	5	Aluminum / Vinyl	1	1952	1190	1	3	1
1183	5	Aluminum / Vinyl	1	1942	1165	1	2	1
1184	5	Brick	1	1951	1164	1	3	1
1185	5	Aluminum / Vinyl	1	1949	1140	1	3	1
1186	5	Brick	1	1950	1123	1	3	1
1187	5	Aluminum / Vinyl	1	1950	1070	1	3	1
1188	5	Aluminum / Vinyl	1	1948	1051	1	3	1
1189	5	Aluminum / Vinyl	1	1951	1050	1	3	1
1190	5	Aluminum / Vinyl	1	1953	1030	1	4	1
1191	5	Aluminum / Vinyl	1	1952	1014	1	3	1
1192	5	Aluminum / Vinyl	1	1954	964	1	3	1
1193	5	Aluminum / Vinyl	1	1950	906	1	3	2
1194	5	Aluminum / Vinyl	1	1955	840	1	2	1
1195	5	Fiber-Cement	2	2000	2573	1	4	3
1196	5	Aluminum / Vinyl	2	1953	1768	1	3	1
1197	5	Stone	1	1928	1889	2	3	2
1198	5	Masonry / Frame	2	1959	2783	2	6	2
1199	5	Brick	2	1952	2410	2	5	2
1200	5	Brick	2	1955	2278	2	4	2
1201	5	Brick	1	1953	1915	1	2	2
1202	5	Brick	1	1960	1281	1	3	1
1203	5	Aluminum / Vinyl	1	1958	1264	1	3	1
1204	5	Aluminum / Vinyl	1	1959	1217	1	3	1
1205	5	Aluminum / Vinyl	1	1980	1199	1	3	1
1206	5	Brick	1	1957	1188	1	3	1
1207	5	Aluminum / Vinyl	1	1955	1132	1	3	1
1208	5	Brick	1	1961	1075	1	3	1
1209	5	Aluminum / Vinyl	1	1952	1064	1	2	1
1210	5	Brick	1	1950	1032	1	3	1
1211	5	Aluminum / Vinyl	1	1956	1027	1	3	1

1212	5	Brick	1	1959	986	1	3	1
1213	5	Aluminum / Vinyl	1	1954	984	1	3	1
1214	5	Aluminum / Vinyl	1	1954	981	1	2	1
1215	5	Aluminum / Vinyl	1	1955	970	1	3	1
1216	6	Aluminum / Vinyl	1.5	1923	1896	2	3	2
1217	6	Aluminum / Vinyl	1	1925	1384	1	2	1
1218	6	Aluminum / Vinyl	1	1895	1088	1	3	1
1219	7	Stone	1	1937	1762	1	5	2
1220	7	Stone	1	1935	1397	1	3	1
1221	7	Aluminum / Vinyl	1	1938	1296	1	2	1
1222	7	Stucco	1.5	1937	1127	1	2	1
1223	7	Aluminum / Vinyl	1	1953	1008	1	2	1
1224	7	Aluminum / Vinyl	1	1927	856	1	3	1
1225	7	Masonry / Frame	2	1930	2782	2	4	2
1226	7	Stone	2	1939	2560	2	4	2
1227	7	Aluminum / Vinyl	1.5	1957	2002	2	5	2
1228	7	Stone	1	1939	951	1	2	1
1229	7	Brick	1	1950	752	1	2	1
1230	8	Aluminum / Vinyl	1	1940	896	1	2	1
1231	8	Aluminum / Vinyl	2	1908	3356	2	8	2
1232	8	Frame	2	1912	2138	2	5	2
1233	8	Aluminum / Vinyl	1.5	1916	1760	2	5	2
1234	8	Frame	1.5	1906	1365	2	3	2
1235	8	Aluminum / Vinyl	1	1928	1612	1	4	1
1236	8	Aluminum / Vinyl	1.5	1899	1767	1	3	2
1237	8	Aluminum / Vinyl	1	1902	961	1	4	1
1238	9	Aluminum / Vinyl	2	2009	2220	1	4	2
1239	9	Aluminum / Vinyl	2	1967	1602	1	4	1
1240	9	Aluminum / Vinyl	1	2005	1920	1	3	3
1241	9	Stone	1	1958	1750	1	3	1
1242	9	Aluminum / Vinyl	1	1978	1327	1	3	1
1243	9	Aluminum / Vinyl	1	1966	1227	1	3	1
1244	9	Aluminum / Vinyl	1	1981	1209	1	3	2
1245	9	Frame	1	1962	1177	1	3	1
1246	9	Aluminum / Vinyl	1	1967	1063	1	3	1
1247	9	Brick	1	1959	992	1	3	1
1248	9	Aluminum / Vinyl	2	1980	2464	2	6	2
1249	10	Brick	1.5	1939	1619	1	3	2
1250	10	Aluminum / Vinyl	1.5	1939	1573	1	3	1
1251	10	Brick	1	1953	1389	1	3	1
1252	10	Brick	1	1947	1127	1	3	1
1253	10	Stone	1	1941	1120	1	2	1
1254	10	Frame	1.5	1939	1116	1	2	1

1255	10	Brick	1	1946	1038	1	3	1
1256	10	Brick	1	1951	1033	1	3	1
1257	10	Aluminum / Vinyl	1	1952	728	1	2	1
1258	10	Masonry / Frame	2	1952	1486	1	3	1
1259	10	Aluminum / Vinyl	2	1938	1134	1	2	1
1260	10	Aluminum / Vinyl	1.5	1927	1947	2	3	2
1261	10	Brick	1.5	1929	1945	2	3	2
1262	10	Masonry / Frame	2	1952	2688	2	6	2
1263	10	Aluminum / Vinyl	1.5	1926	1708	1	4	3
1264	10	Aluminum / Vinyl	1	1924	1701	1	4	1
1265	10	Aluminum / Vinyl	1	1919	1632	1	5	2
1266	10	Brick	1	1923	1557	1	4	1
1267	10	Aluminum / Vinyl	1	1926	1424	1	3	1
1268	10	Aluminum / Vinyl	1	1926	1404	1	3	1
1269	10	Aluminum / Vinyl	1	1919	1314	1	4	1
1270	10	Aluminum / Vinyl	1	1927	1062	1	2	1
1271	10	Brick	1	1931	1053	1	2	1
1272	10	Aluminum / Vinyl	1	1919	1027	1	4	1
1273	10	Aluminum / Vinyl	1	1926	982	1	2	1
1274	10	Aluminum / Vinyl	1	1926	923	1	2	1
1275	10	Brick	1	1955	1488	1	4	1
1276	10	Brick	1	1954	1278	1	3	1
1277	10	Aluminum / Vinyl	1	1950	743	1	2	1
1278	10	Aluminum / Vinyl	1	1955	692	1	2	1
1279	10	Aluminum / Vinyl	1	1920	1304	1	3	1
1280	10	Frame	1	1924	1175	1	3	2
1281	10	Aluminum / Vinyl	1	1923	1008	1	3	1
1282	10	Brick	1	1923	880	1	2	1
1283	10	Aluminum / Vinyl	1	1921	768	1	2	1
1284	10	Stone	1.5	1931	1837	1	3	1
1285	10	Stone	1.5	1931	1837	1	3	1
1286	11	Aluminum / Vinyl	1	1952	1833	1	4	2
1287	11	Stone	1	1951	1710	1	2	1
1288	11	Aluminum / Vinyl	1	1956	1611	1	4	1
1289	11	Stone	1.5	1950	1598	1	3	1
1290	11	Aluminum / Vinyl	1	1950	1588	1	3	2
1291	11	Aluminum / Vinyl	1	1953	1325	1	3	3
1292	11	Brick	1	1949	1294	1	3	2
1293	11	Aluminum / Vinyl	1	1959	1159	1	4	1
1294	11	Aluminum / Vinyl	1	1954	1142	1	4	2
1295	11	Aluminum / Vinyl	1	1949	1032	1	3	1
1296	11	Aluminum / Vinyl	1	1942	1023	1	3	1
1297	11	Brick	1	1950	975	1	2	1

1298	11	Stone	2	1945	1858	1	3	1
1299	11	Aluminum / Vinyl	2	1956	1540	1	3	1
1300	11	Brick	2	1936	1448	1	3	1
1301	11	Aluminum / Vinyl	1.5	1968	1985	2	5	2
1302	11	Brick	1	1966	1443	1	3	1
1303	11	Aluminum / Vinyl	1	1942	1411	1	3	2
1304	11	Brick	1	1967	1383	1	3	1
1305	11	Brick	1	1958	1350	1	3	1
1306	11	Brick	1	1966	1264	1	3	1
1307	11	Aluminum / Vinyl	1	1959	1250	1	3	1
1308	11	Brick	1	1959	1225	1	2	1
1309	11	Aluminum / Vinyl	1	1962	1169	1	3	1
1310	11	Brick	1	1960	1144	1	3	1
1311	11	Brick	1	1959	1135	1	3	1
1312	11	Brick	1	1959	1131	1	3	2
1313	11	Brick	1	1959	1126	1	3	1
1314	11	Aluminum / Vinyl	1	1965	1107	1	3	1
1315	11	Aluminum / Vinyl	1	1959	1080	1	3	1
1316	11	Brick	1	1972	1078	1	1	2
1317	11	Aluminum / Vinyl	1	1962	1058	1	3	1
1318	11	Frame	1	1957	1053	1	3	1
1319	11	Brick	1	1959	1023	1	3	1
1320	11	Stone	1	1955	1015	1	3	1
1321	11	Brick	1	1953	1013	1	3	1
1322	11	Aluminum / Vinyl	1	1942	1008	1	3	1
1323	11	Aluminum / Vinyl	1	1956	973	1	2	1
1324	11	Frame	1	1953	962	1	2	1
1325	11	Aluminum / Vinyl	1	1953	956	1	3	1
1326	11	Aluminum / Vinyl	1	1960	931	1	3	1
1327	11	Brick	1	1956	923	1	3	1
1328	11	Frame	1	1951	886	1	2	1
1329	11	Aluminum / Vinyl	1	1953	864	1	3	1
1330	11	Aluminum / Vinyl	1	1956	864	1	3	1
1331	11	Aluminum / Vinyl	1	1942	794	1	2	1
1332	11	Block	1	1945	781	1	2	2
1333	11	Aluminum / Vinyl	1	1951	644	1	2	1
1334	11	Brick	1.5	1918	1616	1	2	1
1335	12	Aluminum / Vinyl	2	2006	1684	1	4	2
1336	12	Aluminum / Vinyl	1	1899	1191	1	3	2
1337	12	Aluminum / Vinyl	1	1900	1178	1	3	1
1338	12	Aluminum / Vinyl	1	1900	978	1	3	1
1339	12	Aluminum / Vinyl	1	1905	1496	2	4	2
1340	12	Aluminum / Vinyl	1.5	1873	1692	1	4	2

1341	12	Stucco	1	1903	1024	1	3	1
1343	12	Frame	2	1900	2766	3	7	3
1344	13	Brick	1	1955	1583	1	4	2
1345	13	Brick	1	1952	1400	1	4	1
1346	13	Brick	1	1950	1377	1	3	2
1347	13	Brick	1	1950	1172	1	2	1
1348	13	Aluminum / Vinyl	1	1961	1160	1	4	1
1349	13	Brick	1	1951	1142	1	3	1
1350	13	Stone	1	1950	1109	1	3	1
1351	13	Brick	1.5	1936	1078	1	2	1
1352	13	Aluminum / Vinyl	1	1947	1046	1	3	1
1353	13	Brick	1	1953	1042	1	3	1
1354	13	Aluminum / Vinyl	1	1947	834	1	2	1
1355	13	Aluminum / Vinyl	1	1906	1320	1	4	2
1356	13	Aluminum / Vinyl	1.5	1885	1143	1	2	1
1357	13	Brick	1.5	1925	2224	2	5	2
1358	13	Masonry / Frame	2	1949	1800	2	4	2
1359	13	Aluminum / Vinyl	2	1950	1417	2	3	2
1360	13	Frame	1	1927	1324	1	3	2
1361	13	Brick	1	1968	1539	1	2	2
1362	13	Aluminum / Vinyl	1	1974	1324	1	3	1
1363	13	Brick	1	1961	1196	1	3	1
1364	13	Aluminum / Vinyl	1	1963	1148	1	3	1
1365	13	Aluminum / Vinyl	1	1950	1102	1	2	1
1366	13	Brick	1	1960	1064	1	3	1
1367	13	Brick	1	1961	1036	1	3	2
1368	13	Aluminum / Vinyl	1	1959	970	1	3	1
1369	13	Aluminum / Vinyl	1	1953	744	1	2	1
1370	13	Aluminum / Vinyl	1	1950	699	1	2	1
1371	13	Aluminum / Vinyl	1	1955	672	1	2	1
1372	13	Frame	1	1929	1130	1	2	1
1373	14	Aluminum / Vinyl	1	1951	1098	1	4	1
1374	14	Brick	2	1940	1838	1	3	1
1375	14	Aluminum / Vinyl	2	2006	1682	1	3	2
1376	14	Aluminum / Vinyl	2	1941	1213	1	2	1
1377	14	Aluminum / Vinyl	2	1950	986	1	2	1
1378	14	Aluminum / Vinyl	1	1896	1127	1	3	1
1379	14	Aluminum / Vinyl	1	1898	864	1	2	1
1380	14	Aluminum / Vinyl	1.5	1930	2152	2	5	2
1381	14	Aluminum / Vinyl	1.5	1925	1873	2	4	2
1382	14	Aluminum / Vinyl	1.5	1927	1626	2	3	2
1383	14	Aluminum / Vinyl	1.5	1925	1410	2	3	2
1384	14	Block	1.5	1900	1761	2	4	2

1385	14	Aluminum / Vinyl	1.5	1921	1627	1	3	2
1386	14	Brick	1.5	1930	2062	1	4	2
1387	14	Aluminum / Vinyl	1	1924	1931	1	5	2
1388	14	Brick	1	1925	1774	1	3	1
1389	14	Aluminum / Vinyl	1.5	1920	1155	1	3	1
1390	14	Aluminum / Vinyl	1	1927	1060	1	2	1
1391	14	Aluminum / Vinyl	1	1964	1792	1	3	1
1392	14	Aluminum / Vinyl	1	1951	1056	1	3	1
1393	14	Aluminum / Vinyl	1	1969	973	1	2	1
1394	14	Aluminum / Vinyl	1	1941	934	1	2	1
1395	14	Aluminum / Vinyl	1	1941	853	1	2	1
1396	14	Aluminum / Vinyl	1	1950	725	1	2	1
1397	14	Aluminum / Vinyl	1	1948	679	1	2	1
1398	14	Stucco	1	1952	672	1	1	1
1399	14	Prem Wood	1	1905	1479	1	2	1
1400	14	Aluminum / Vinyl	1	1899	1419	1	3	2
1401	14	Aluminum / Vinyl	1.5	1922	1386	1	2	3
1402	14	Frame	1	1904	1315	1	2	1
1403	14	Aluminum / Vinyl	1	1916	1191	1	3	1
1404	14	Aluminum / Vinyl	1	1922	1060	1	3	1
1405	14	Frame	1.5	1921	1002	1	2	1
1406	14	Brick	1	1932	888	1	2	1
1407	14	Aluminum / Vinyl	1	1925	829	1	2	1
1408	14	Brick	2	1954	1769	2	4	2
1409	15	Aluminum / Vinyl	1.5	1900	1771	1	4	2
1410	15	Aluminum / Vinyl	1	1890	1425	1	4	1
1411	15	Frame	1	1891	1124	1	4	1
1412	15	Masonry / Frame	2	1928	2922	1	5	2
1413	1	Stone	1	1946	1670	1	4	1
1414	1	Aluminum / Vinyl	1	1964	1213	1	3	1
1415	1	Frame	1	1952	1144	1	3	1
1416	1	Aluminum / Vinyl	2	1924	1171	1	2	2
1417	1	Aluminum / Vinyl	1	1927	957	1	4	1
1418	2	Brick	1	1956	1090	1	3	1
1419	2	Masonry / Frame	2	1959	1635	1	3	1
1420	2	Frame	1	1925	640	1	2	1
1421	2	Brick	1.5	1958	2013	2	5	2
1422	2	Brick	1	1958	1287	1	3	1
1423	2	Aluminum / Vinyl	1	1955	1284	1	2	1
1424	2	Brick	1	1958	1215	1	3	1
1425	2	Aluminum / Vinyl	1	1960	1183	1	3	1
1426	2	Brick	1	1959	1153	1	3	1
1427	2	Frame	1	1971	948	1	3	1

1428	2	Aluminum / Vinyl	1	1958	938	1	3	1
1429	2	Aluminum / Vinyl	1	1955	936	1	3	1
1430	2	Frame	1	1955	864	1	3	1
1431	3	Fiber-Cement	2	1890	1852	1	5	3
1432	3	Frame	2	1922	2196	2	4	2
1433	3	Aluminum / Vinyl	1.5	1889	1680	2	5	2
1434	3	Frame	1.5	1901	1643	2	4	2
1435	3	Frame	1.5	1895	1557	2	4	1
1436	3	Aluminum / Vinyl	1	1926	1384	1	4	1
1437	3	Frame	2	1897	4077	1	5	3
1438	3	Stucco	2	1920	3037	1	5	3
1439	3	Brick	2	1922	2242	1	4	2
1440	3	Aluminum / Vinyl	1.5	1890	1408	1	3	2
1441	3	Aluminum / Vinyl	1.5	1891	1291	1	3	2
1442	3	Stucco	2	1911	3486	1	5	2
1443	4	Brick	2	1885	4140	2	6	2
1444	5	Brick	1	1948	1473	1	4	2
1445	5	Brick	1	1948	1274	1	3	1
1446	5	Brick	1	1953	1261	1	3	1
1447	5	Aluminum / Vinyl	1	1949	1255	1	3	1
1448	5	Aluminum / Vinyl	1	1951	1255	1	3	1
1449	5	Aluminum / Vinyl	1	1948	1142	1	3	1
1450	5	Brick	1	1949	1125	1	2	1
1451	5	Brick	1	1949	1086	1	3	1
1452	5	Aluminum / Vinyl	1	1946	1048	1	3	1
1453	5	Aluminum / Vinyl	1	1952	1005	1	3	1
1454	5	Aluminum / Vinyl	1	1941	921	1	3	1
1455	5	Masonry / Frame	2	1951	1473	1	3	1
1456	5	Masonry / Frame	2	1939	1188	1	3	1
1457	5	Brick	1	1950	2411	2	4	2
1458	5	Masonry / Frame	2	1959	2386	2	6	2
1459	5	Brick	1.5	1952	2156	2	5	2
1460	5	Stone	1	1950	1922	1	2	2
1461	5	Masonry / Frame	1	1961	1749	1	3	1
1462	5	Aluminum / Vinyl	1	2004	1672	1	3	2
1463	5	Frame	1	1956	1498	1	3	1
1464	5	Brick	1	1948	1426	1	3	1
1465	5	Brick	1	1956	1388	1	3	1
1466	5	Aluminum / Vinyl	1	1957	1384	1	4	1
1467	5	Aluminum / Vinyl	1	1963	1120	1	3	1
1468	5	Brick	1	1949	1053	1	2	1
1469	5	Aluminum / Vinyl	1	1956	1051	1	3	1
1470	5	Aluminum / Vinyl	1	1955	1050	1	3	1

1471	5	Brick	1	1951	1016	1	3	1
1472	5	Aluminum / Vinyl	1	1949	989	1	2	1
1473	5	Aluminum / Vinyl	1	1954	962	1	3	1
1474	5	Brick	1	1955	935	1	2	2
1475	5	Brick	1	1951	744	1	2	1
1476	5	Aluminum / Vinyl	1.5	1925	1019	1	3	1
1477	6	Aluminum / Vinyl	1.5	1924	1807	2	4	2
1478	6	Aluminum / Vinyl	1	1919	1313	1	2	1
1479	6	Frame	1	1925	1212	1	3	1
1480	6	Aluminum / Vinyl	2	1963	956	1	3	1
1481	7	Aluminum / Vinyl	1.5	1939	1315	1	3	1
1482	7	Brick	1	1937	941	1	3	1
1483	7	Aluminum / Vinyl	1	1949	678	1	2	1
1484	7	Stone	2	1944	2408	2	4	2
1485	7	Aluminum / Vinyl	1.5	1926	2045	2	4	2
1486	7	Aluminum / Vinyl	1.5	1913	1712	2	4	2
1487	7	Frame	1	1925	1593	1	3	1
1488	7	Brick	1	1927	1589	1	4	1
1489	7	Brick	1.5	1934	1604	1	3	1
1490	8	Aluminum / Vinyl	1	1900	1031	1	3	1
1491	8	Aluminum / Vinyl	1.5	1913	2019	2	4	2
1492	8	Frame	1.5	1918	1809	2	3	2
1493	8	Aluminum / Vinyl	1	1924	1556	1	4	2
1494	8	Stucco	1	1920	1458	1	5	1
1495	8	Frame	2	1924	1762	1	3	3
1496	8	Aluminum / Vinyl	1.5	1900	1637	1	5	2
1497	8	Aluminum / Vinyl	1	1909	1576	1	5	1
1498	8	Aluminum / Vinyl	1.5	1920	1024	1	3	1
1499	8	Stucco	1.5	1910	2816	3	6	3
1500	9	Aluminum / Vinyl	1	1957	1482	1	4	1
1501	9	Aluminum / Vinyl	1	1964	1390	1	4	1
1502	9	Aluminum / Vinyl	1	1969	1325	1	3	1
1503	9	Aluminum / Vinyl	1	1978	1320	1	3	1
1504	9	Brick	1	1957	1131	1	3	1
1505	9	Aluminum / Vinyl	1	1962	1074	1	3	1
1506	10	Frame	1	1952	1291	1	3	1
1507	10	Aluminum / Vinyl	1	1942	1247	1	3	2
1508	10	Stone	1	1945	1113	1	3	1
1509	10	Aluminum / Vinyl	1	1942	968	1	3	1
1510	10	Aluminum / Vinyl	1	1952	728	1	2	1
1511	10	Stone	2	1941	1894	1	3	1
1512	10	Brick	2	1938	1694	1	3	1
1513	10	Aluminum / Vinyl	2	1921	1657	1	3	1

1514	10	Aluminum / Vinyl	2	1926	1271	1	3	1
1515	10	Aluminum / Vinyl	1	1927	872	1	1	1
1516	10	Brick	1.5	1928	2108	2	5	2
1517	10	Brick	1	1925	1975	2	3	3
1518	10	Aluminum / Vinyl	2	1923	1918	2	4	2
1519	10	Brick	1.5	1929	1821	2	4	2
1520	10	Brick	2	1967	2104	2	4	2
1521	10	Stone	2	1945	2026	2	4	2
1522	10	Stone	2	1945	2026	2	4	2
1523	10	Aluminum / Vinyl	1	1928	1279	1	3	1
1524	10	Aluminum / Vinyl	1	1929	1253	1	3	1
1525	10	Brick	1	1922	1221	1	3	1
1526	10	Stone	1	1949	1370	1	2	1
1527	10	Stone	1	1949	1011	1	2	1
1528	10	Aluminum / Vinyl	1	1949	808	1	2	1
1529	10	Stucco	1	1916	1397	1	3	1
1530	10	Aluminum / Vinyl	1	1923	990	1	4	1
1531	10	Frame	1	1925	870	1	1	1
1532	10	Frame	2	1928	2990	3	5	3
1533	10	Brick	2	1928	3178	1	5	3
1534	10	Brick	1	1928	1413	1	2	1
1535	11	Aluminum / Vinyl	1	1963	1696	1	4	2
1536	11	Aluminum / Vinyl	1.5	1940	1467	1	3	1
1537	11	Brick	1	1952	1400	1	4	1
1538	11	Brick	1	1950	1320	1	4	1
1539	11	Stone	1	1953	1246	1	3	1
1540	11	Brick	1	1950	1225	1	3	1
1541	11	Brick	1	1938	1205	1	3	1
1542	11	Aluminum / Vinyl	1	1949	1183	1	3	1
1543	11	Aluminum / Vinyl	1	1952	1082	1	4	1
1544	11	Aluminum / Vinyl	1	1946	1036	1	3	1
1545	11	Aluminum / Vinyl	1	1953	1004	1	3	1
1546	11	Aluminum / Vinyl	1	1958	934	1	3	1
1547	11	Brick	1	1940	887	1	2	1
1548	11	Aluminum / Vinyl	1	1955	800	1	2	1
1549	11	Stone	2	1932	2942	1	4	1
1550	11	Aluminum / Vinyl	2	1985	1858	1	4	2
1551	11	Brick	2	1950	1604	1	3	1
1552	11	Stone	2	1937	1584	1	3	1
1553	11	Aluminum / Vinyl	2	1942	1288	1	3	1
1554	11	Aluminum / Vinyl	1.5	1929	1954	2	4	2
1555	11	Masonry / Frame	2	1959	2299	2	6	2
1556	11	Aluminum / Vinyl	1.5	1957	2237	2	5	2

1557	11	Brick	1.5	1956	2114	2	5	3
1558	11	Brick	1.5	1957	1616	2	4	2
1559	11	Aluminum / Vinyl	1	1920	1645	1	4	1
1560	11	Brick	1	1928	1102	1	3	1
1561	11	Brick	1	1964	1806	1	4	2
1562	11	Brick	1	1959	1468	1	3	1
1563	11	Brick	1	1958	1189	1	3	1
1564	11	Brick	1	1955	1153	1	3	1
1565	11	Aluminum / Vinyl	1	1954	1123	1	3	1
1566	11	Brick	1	1954	1112	1	3	1
1567	11	Brick	1	1955	1109	1	3	1
1568	11	Brick	1	1958	1094	1	3	1
1569	11	Brick	1	1959	1091	1	3	1
1570	11	Brick	1	1954	1085	1	3	1
1571	11	Aluminum / Vinyl	1	1950	1072	1	3	1
1572	11	Aluminum / Vinyl	1	1955	1020	1	2	1
1573	11	Brick	1	1952	1013	1	3	1
1574	11	Brick	1	1954	998	1	2	1
1575	11	Stone	1	1950	982	1	3	2
1576	11	Aluminum / Vinyl	1	1955	955	1	2	2
1577	11	Aluminum / Vinyl	1	1952	954	1	2	1
1578	11	Brick	1	1956	940	1	3	1
1579	11	Aluminum / Vinyl	1	1953	879	1	3	1
1580	11	Aluminum / Vinyl	1	1956	864	1	2	1
1581	11	Aluminum / Vinyl	1	1942	833	1	2	1
1582	11	Brick	1	1947	826	1	2	1
1583	11	Aluminum / Vinyl	1	1952	811	1	2	1
1584	11	Aluminum / Vinyl	2	1986	2524	2	4	2
1585	11	Brick	2	1944	1870	2	4	2
1586	12	Frame	2	1913	1692	2	4	2
1587	13	Aluminum / Vinyl	1	1971	1494	1	4	1
1588	13	Brick	1	1954	1619	1	4	1
1589	13	Aluminum / Vinyl	1	1966	1608	1	4	2
1590	13	Block	1	1936	1430	1	4	1
1591	13	Aluminum / Vinyl	1	1942	1242	1	3	1
1592	13	Aluminum / Vinyl	1	1935	1215	1	3	1
1593	13	Aluminum / Vinyl	1	1950	1183	1	3	1
1594	13	Brick	1	1939	1173	1	3	2
1595	13	Aluminum / Vinyl	1	1943	1148	1	3	1
1596	13	Stucco	1	1949	1057	1	3	2
1597	13	Aluminum / Vinyl	1	1958	991	1	3	1
1598	13	Aluminum / Vinyl	1.5	1950	2005	2	3	3
1599	13	Block	1	1912	1572	2	4	2

1600	13	Aluminum / Vinyl	1	1928	1446	1	3	2
1601	13	Aluminum / Vinyl	1	1977	1550	1	3	1
1602	13	Frame	1	1958	1424	1	3	2
1603	13	Aluminum / Vinyl	1	1963	1329	1	3	2
1604	13	Brick	1	1960	1201	1	2	2
1605	13	Brick	1	1956	1189	1	3	1
1606	13	Aluminum / Vinyl	1	1969	1092	1	3	1
1607	13	Frame	1	1960	965	1	3	1
1608	13	Aluminum / Vinyl	1	1950	911	1	3	1
1609	13	Aluminum / Vinyl	1	1954	870	1	2	1
1610	13	Aluminum / Vinyl	1	1947	861	1	2	1
1611	13	Aluminum / Vinyl	1	1940	852	1	2	1
1612	13	Aluminum / Vinyl	1	1947	732	1	2	1
1613	13	Aluminum / Vinyl	1.5	1926	1172	1	4	1
1614	14	Aluminum / Vinyl	1	1939	1238	1	3	2
1615	14	Aluminum / Vinyl	1	1953	1226	1	4	1
1616	14	Aluminum / Vinyl	1	1944	1168	1	3	1
1617	14	Brick	1	1944	1100	1	4	1
1618	14	Aluminum / Vinyl	1	1953	1090	1	4	1
1619	14	Aluminum / Vinyl	1	1950	1040	1	4	1
1620	14	Prem Wood	2	2007	1826	1	4	2
1621	14	Aluminum / Vinyl	2	1938	1320	1	3	1
1622	14	Aluminum / Vinyl	1.5	1900	1356	1	3	2
1623	14	Aluminum / Vinyl	2	1924	2472	2	5	2
1624	14	Brick	2	1932	2208	2	4	2
1625	14	Aluminum / Vinyl	1.5	1923	1875	2	4	2
1626	14	Aluminum / Vinyl	2	1890	2424	2	4	2
1627	14	Brick	1.5	1930	2381	2	3	2
1628	14	Aluminum / Vinyl	2	1925	1848	2	4	2
1629	14	Aluminum / Vinyl	1.5	1910	1680	2	3	2
1630	14	Aluminum / Vinyl	1	1913	1922	2	5	2
1631	14	Aluminum / Vinyl	2	1892	1814	2	4	2
1632	14	Brick	1	1925	2261	1	3	2
1633	14	Stone	1	1933	1650	1	3	1
1634	14	Aluminum / Vinyl	1	1930	1542	1	4	1
1635	14	Aluminum / Vinyl	1	1926	1434	1	3	1
1636	14	Aluminum / Vinyl	1	1910	1372	1	3	1
1637	14	Aluminum / Vinyl	1	1927	1248	1	3	1
1638	14	Brick	1	1928	1145	1	3	1
1639	14	Aluminum / Vinyl	1	1956	1003	1	3	1
1640	14	Aluminum / Vinyl	1	1943	710	1	2	1
1641	14	Frame	1	1908	1519	1	5	2
1642	14	Aluminum / Vinyl	1	1906	1470	1	4	2

1643	14	Aluminum / Vinyl	1.5	1908	1290	1	3	1
1644	14	Aluminum / Vinyl	1.5	1911	1283	1	4	1
1645	14	Aluminum / Vinyl	1	1921	1120	1	3	1
1646	14	Aluminum / Vinyl	1	1890	959	1	2	1
1647	15	Aluminum / Vinyl	1	1922	1962	1	5	2
1648	15	Aluminum / Vinyl	1	1899	1238	1	4	1
1649	1	Brick	1	1937	1466	1	3	2
1650	1	Brick	1	1941	1456	1	4	1
1651	1	Aluminum / Vinyl	1	1946	1384	1	5	1
1652	1	Aluminum / Vinyl	1	1953	1298	1	4	2
1653	1	Aluminum / Vinyl	1.5	1937	1120	1	2	1
1654	1	Aluminum / Vinyl	1.5	1936	1054	1	2	1
1655	1	Aluminum / Vinyl	2	1936	1144	1	3	1
1656	1	Aluminum / Vinyl	2	1914	1835	2	4	2
1657	1	Aluminum / Vinyl	1	1924	1340	1	4	1
1658	1	Aluminum / Vinyl	1	1926	1214	1	3	1
1659	1	Aluminum / Vinyl	1	1955	1082	1	3	1
1660	1	Aluminum / Vinyl	1	1954	963	1	3	1
1661	1	Aluminum / Vinyl	1	1953	870	1	3	1
1662	1	Aluminum / Vinyl	1	1939	795	1	2	1
1663	1	Brick	1	1929	1548	1	4	2
1664	2	Aluminum / Vinyl	1.5	1951	1428	1	4	1
1665	2	Aluminum / Vinyl	1.5	1935	1309	1	5	1
1666	2	Aluminum / Vinyl	1	1952	1073	1	4	1
1667	2	Brick	1	1947	920	1	2	1
1668	2	Aluminum / Vinyl	1	1929	1153	1	3	1
1669	2	Frame	1	1963	1408	1	3	2
1670	2	Frame	1	1966	1308	1	4	1
1671	2	Aluminum / Vinyl	1	1967	1215	1	3	1
1672	2	Aluminum / Vinyl	1	1959	1135	1	3	1
1673	2	Brick	1	1959	1116	1	3	2
1674	2	Aluminum / Vinyl	1	1946	1040	1	2	1
1675	2	Aluminum / Vinyl	1	1956	925	1	3	1
1676	2	Frame	1	1959	910	1	3	1
1677	2	Aluminum / Vinyl	1	1957	886	1	3	1
1678	2	Aluminum / Vinyl	1	1955	864	1	3	1
1679	2	Aluminum / Vinyl	1	1954	864	1	3	1
1680	3	Aluminum / Vinyl	1	1898	1296	1	3	1
1681	3	Aluminum / Vinyl	1	1895	870	1	4	2
1682	3	Aluminum / Vinyl	1.5	1925	2225	2	4	2
1683	3	Frame	2	1920	2016	2	6	2
1684	3	Aluminum / Vinyl	1.5	1917	1761	2	4	2
1685	3	Brick	1	1926	1420	1	3	1

1686	3	Brick	2	1912	3066	1	6	2
1687	3	Stucco	2	1922	1947	1	4	1
1688	3	Stucco	2	1910	1886	1	3	1
1689	3	Stucco	2	1910	1886	1	3	1
1690	3	Stucco	2	1912	1805	1	4	1
1691	3	Aluminum / Vinyl	1	1892	1080	1	3	1
1692	4	Frame	2	1908	2362	1	5	1
1693	5	Stone	1	1951	1417	1	3	1
1694	5	Brick	1	1947	1374	1	3	1
1695	5	Stone	1	1946	1272	1	3	1
1696	5	Aluminum / Vinyl	1	1946	1264	1	3	1
1697	5	Brick	1	1952	1230	1	2	1
1698	5	Brick	1	1952	1166	1	3	2
1699	5	Aluminum / Vinyl	1	1952	981	1	3	1
1700	5	Aluminum / Vinyl	1	1951	980	1	2	1
1701	5	Aluminum / Vinyl	1	1951	980	1	2	1
1702	5	Aluminum / Vinyl	1	1942	812	1	2	1
1703	5	Masonry / Frame	2	1965	2252	1	5	3
1704	5	Aluminum / Vinyl	1.5	1922	1817	2	5	2
1705	5	Brick	1	1956	2458	2	5	2
1706	5	Aluminum / Vinyl	1	1968	1552	1	3	1
1707	5	Stone	1	1956	1535	1	3	1
1708	5	Stone	1	1950	1507	1	4	2
1709	5	Brick	1	1997	1456	1	3	1
1710	5	Aluminum / Vinyl	1	1949	1328	1	2	2
1711	5	Brick	1	1961	1324	1	3	2
1712	5	Brick	1	1958	1180	1	3	1
1713	5	Brick	1	1952	1153	1	2	1
1714	5	Brick	1	1953	1142	1	3	1
1715	5	Aluminum / Vinyl	1	1957	1136	1	3	1
1716	5	Brick	1	1957	1124	1	2	1
1717	5	Brick	1	1959	1107	1	3	1
1718	5	Aluminum / Vinyl	1	1955	1064	1	3	1
1719	5	Aluminum / Vinyl	1	1955	1043	1	3	1
1720	5	Brick	1	1952	1029	1	2	1
1721	5	Brick	1	1961	1011	1	3	1
1722	5	Aluminum / Vinyl	1	1955	1008	1	3	1
1723	5	Aluminum / Vinyl	1	1955	1008	1	3	1
1724	5	Stone	1	1950	960	1	2	1
1725	5	Aluminum / Vinyl	1	1950	927	1	3	1
1726	5	Stone	1	1939	924	1	2	1
1727	5	Stone	1	1956	920	1	3	1
1728	5	Aluminum / Vinyl	1	1948	800	1	2	1

1729	5	Aluminum / Vinyl	1	1950	728	1	2	2
1730	5	Aluminum / Vinyl	2	1977	2896	2	6	2
1731	6	Aluminum / Vinyl	1	1895	934	1	2	1
1732	6	Aluminum / Vinyl	2	1922	2612	2	4	2
1733	6	Frame	2	1911	2202	2	4	2
1734	6	Aluminum / Vinyl	1.5	1916	1965	2	4	2
1735	6	Frame	1.5	1914	1693	2	4	2
1736	6	Aluminum / Vinyl	1.5	1873	1664	2	4	2
1737	6	Aluminum / Vinyl	1.5	1893	2037	1	4	2
1738	6	Aluminum / Vinyl	2	1910	1736	1	4	1
1739	7	Brick	1	1945	1731	1	4	1
1740	7	Brick	1	1951	1372	1	4	2
1741	7	Aluminum / Vinyl	1	1954	1176	1	4	1
1742	7	Aluminum / Vinyl	1	1950	1113	1	3	1
1743	7	Masonry / Frame	2	1952	2020	1	4	3
1744	7	Aluminum / Vinyl	1	1928	774	1	2	1
1745	7	Aluminum / Vinyl	1	1930	563	1	2	1
1746	7	Aluminum / Vinyl	2	1950	1932	2	4	2
1747	7	Frame	2	1913	2152	2	4	2
1748	7	Stone	1	1954	1448	1	3	1
1749	7	Aluminum / Vinyl	1	1953	912	1	3	1
1750	7	Aluminum / Vinyl	1	1920	1244	1	3	1
1751	7	Stone	2	1935	1980	1	3	1
1752	8	Aluminum / Vinyl	1	1951	1375	1	3	2
1753	8	Aluminum / Vinyl	1	1952	779	1	2	1
1754	8	Stucco	1.5	1915	2003	2	5	2
1755	8	Frame	2	1912	2528	2	6	2
1756	8	Frame	2	1907	1936	2	5	2
1757	8	Aluminum / Vinyl	1	1880	1875	2	4	2
1758	8	Stucco	1	1921	1870	1	3	2
1759	8	Aluminum / Vinyl	1.5	1920	1510	1	3	1
1760	8	Aluminum / Vinyl	1	1923	1454	1	4	2
1761	8	Aluminum / Vinyl	1	1925	1440	1	3	2
1762	8	Stucco	1	1923	1255	1	3	1
1763	8	Frame	1	1925	1237	1	3	1
1764	8	Aluminum / Vinyl	1	1926	1106	1	2	1
1765	8	Aluminum / Vinyl	1	1900	1431	1	4	1
1767	9	Aluminum / Vinyl	1	1956	1872	1	3	1
1768	9	Aluminum / Vinyl	1.5	1976	1437	1	4	2
1769	9	Aluminum / Vinyl	1	1953	1050	1	4	1
1770	9	Aluminum / Vinyl	2	2008	2208	1	4	2
1771	9	Brick	1	1964	1669	1	3	1
1772	9	Aluminum / Vinyl	1	1979	1581	1	4	2

1773	9	Aluminum / Vinyl	1	1974	1324	1	3	1
1774	9	Aluminum / Vinyl	1	1953	1309	1	3	2
1775	9	Aluminum / Vinyl	1	1956	1246	1	3	1
1776	9	Aluminum / Vinyl	1	1973	1200	1	4	1
1777	9	Frame	1	1966	1184	1	3	1
1778	9	Masonry / Frame	1	1957	1165	1	3	1
1779	9	Aluminum / Vinyl	1	1957	1155	1	3	1
1780	9	Aluminum / Vinyl	1	1992	1143	1	3	1
1781	9	Aluminum / Vinyl	1	1972	1140	1	3	1
1782	9	Frame	1	1968	1063	1	2	1
1783	9	Aluminum / Vinyl	1	1965	1033	1	3	1
1784	9	Aluminum / Vinyl	1	1965	1033	1	3	1
1785	9	Brick	1	1956	962	1	3	1
1786	9	Aluminum / Vinyl	1	1976	906	1	3	1
1787	9	Aluminum / Vinyl	1	1937	826	1	2	1
1788	10	Stone	1.5	1948	1920	1	4	2
1789	10	Brick	1.5	1932	1424	1	3	1
1790	10	Aluminum / Vinyl	1	1945	1307	1	3	2
1791	10	Aluminum / Vinyl	1	1955	1046	1	3	1
1792	10	Aluminum / Vinyl	1	1948	1012	1	4	1
1793	10	Masonry / Frame	2	1921	3268	2	6	2
1794	10	Aluminum / Vinyl	2	1927	2756	2	4	2
1795	10	Stucco	2	1927	2564	2	6	2
1796	10	Aluminum / Vinyl	2	1926	2300	2	4	2
1797	10	Aluminum / Vinyl	1.5	1920	2234	2	4	2
1798	10	Frame	1.5	1928	2076	2	5	2
1799	10	Aluminum / Vinyl	1.5	1928	1973	2	4	2
1800	10	Aluminum / Vinyl	2	1903	1834	2	4	2
1801	10	Brick	1	1923	2010	1	4	2
1802	10	Aluminum / Vinyl	1	1912	1672	1	3	2
1803	10	Aluminum / Vinyl	1	1925	1225	1	3	1
1804	10	Aluminum / Vinyl	1	1925	1188	1	4	2
1805	10	Aluminum / Vinyl	1	1926	1141	1	3	1
1806	10	Aluminum / Vinyl	1	1925	964	1	3	1
1807	10	Frame	1	1924	951	1	3	1
1808	10	Brick	1	1953	1264	1	2	1
1809	10	Aluminum / Vinyl	1	1952	1173	1	3	2
1810	10	Aluminum / Vinyl	1	1953	1036	1	3	1
1811	10	Aluminum / Vinyl	1	1951	700	1	2	1
1812	10	Aluminum / Vinyl	2	1925	1430	1	3	1
1813	10	Aluminum / Vinyl	1	1927	1415	1	3	2
1814	10	Aluminum / Vinyl	2	1918	1407	1	2	2
1815	10	Frame	1	1914	1302	1	3	1

1816	10	Aluminum / Vinyl	1	1917	1266	1	3	1
1817	10	Aluminum / Vinyl	2	1927	1197	1	3	1
1818	10	Stone	1.5	1923	2519	1	4	3
1819	11	Aluminum / Vinyl	1	1954	1105	1	3	2
1820	11	Aluminum / Vinyl	1	1943	1093	1	3	1
1821	11	Aluminum / Vinyl	1	1957	1069	1	3	2
1822	11	Stone	1	1951	1024	1	2	1
1823	11	Brick	1	1950	967	1	2	1
1824	11	Aluminum / Vinyl	1	1941	907	1	3	1
1825	11	Aluminum / Vinyl	1	1938	902	1	3	1
1826	11	Aluminum / Vinyl	1	1942	763	1	2	1
1827	11	Aluminum / Vinyl	1	1954	745	1	2	1
1828	11	Aluminum / Vinyl	2	1938	1496	1	3	1
1829	11	Masonry / Frame	2	1930	1478	1	3	2
1830	11	Block	2	1949	1056	1	2	1
1831	11	Aluminum / Vinyl	1.5	1980	1897	2	4	2
1832	11	Aluminum / Vinyl	2	1954	1728	2	6	2
1833	11	Aluminum / Vinyl	2	1954	1728	2	6	2
1834	11	Aluminum / Vinyl	2	1954	1728	2	6	2
1835	11	Brick	1	1963	1632	1	4	1
1836	11	Stone	1	1954	1497	1	2	1
1837	11	Aluminum / Vinyl	1	1965	1317	1	4	1
1838	11	Stone	1	1952	1312	1	2	1
1839	11	Aluminum / Vinyl	1	1963	1251	1	3	1
1840	11	Aluminum / Vinyl	1	1953	1207	1	4	2
1841	11	Brick	1	1958	1206	1	3	1
1842	11	Brick	1	1954	1176	1	3	1
1843	11	Brick	1	1956	1150	1	3	1
1844	11	Aluminum / Vinyl	1	1957	1147	1	3	1
1845	11	Brick	1	1959	1130	1	3	1
1846	11	Brick	1	1957	1129	1	3	1
1847	11	Brick	1	1959	1114	1	3	1
1848	11	Aluminum / Vinyl	1	1953	1069	1	3	1
1849	11	Brick	1	1956	989	1	3	1
1850	11	Aluminum / Vinyl	1	1960	988	1	3	1
1851	11	Aluminum / Vinyl	1	1955	988	1	3	1
1852	11	Aluminum / Vinyl	1	1953	987	1	3	1
1853	11	Frame	1	1953	980	1	3	1
1854	11	Aluminum / Vinyl	1	1953	960	1	3	1
1855	11	Brick	1	1953	921	1	3	1
1856	11	Aluminum / Vinyl	1	1957	914	1	2	1
1857	11	Aluminum / Vinyl	1	1953	894	1	2	1
1858	11	Frame	1	1952	879	1	3	1

1859	11	Aluminum / Vinyl	1	1953	870	1	2	1
1860	11	Aluminum / Vinyl	1	1947	850	1	2	1
1861	11	Aluminum / Vinyl	1	1951	770	1	2	1
1862	11	Aluminum / Vinyl	1	1950	734	1	2	1
1863	11	Aluminum / Vinyl	1	1953	672	1	2	1
1864	11	Aluminum / Vinyl	1	1937	788	1	2	1
1865	11	Aluminum / Vinyl	2	1972	2385	2	6	2
1866	11	Brick	1.5	1930	2016	1	4	1
1867	12	Aluminum / Vinyl	1	1924	1221	1	4	1
1868	13	Aluminum / Vinyl	1	1947	1608	1	3	1
1869	13	Brick	1	1948	1217	1	3	1
1870	13	Brick	1	1951	1120	1	3	1
1871	13	Aluminum / Vinyl	1	1952	1075	1	3	1
1872	13	Brick	1	1951	1002	1	3	1
1873	13	Brick	1	1955	986	1	2	1
1874	13	Aluminum / Vinyl	2	1965	1747	1	3	2
1875	13	Aluminum / Vinyl	2	1959	2052	2	6	2
1876	13	Brick	1.5	1961	1875	2	5	2
1877	13	Aluminum / Vinyl	1.5	1908	1643	2	3	2
1878	13	Frame	1.5	1920	1493	2	3	2
1879	13	Frame	1	1928	1083	1	2	1
1880	13	Frame	1	1938	1530	1	3	1
1881	13	Aluminum / Vinyl	1	1970	1352	1	3	1
1882	13	Brick	1	1966	1160	1	3	1
1883	13	Brick	1	1965	1157	1	3	1
1884	13	Aluminum / Vinyl	1	1958	1154	1	3	1
1885	13	Brick	1	1973	1100	1	2	1
1886	13	Frame	1	1958	1081	1	3	1
1887	13	Aluminum / Vinyl	1	1983	1064	1	3	1
1888	13	Aluminum / Vinyl	1	1958	1045	1	3	1
1889	13	Brick	1	1962	1036	1	3	1
1890	13	Frame	1	1949	843	1	2	1
1891	13	Aluminum / Vinyl	1	1953	705	1	2	1
1892	13	Frame	1.5	1910	1690	1	3	2
1893	13	Aluminum / Vinyl	1	1927	1325	1	4	1
1894	13	Aluminum / Vinyl	1	1928	1225	1	3	2
1895	13	Aluminum / Vinyl	1	1908	1093	1	3	1
1896	14	Aluminum / Vinyl	1	1947	1387	1	4	3
1897	14	Brick	1	1944	1353	1	4	2
1898	14	Block	1	1948	1163	1	3	1
1899	14	Aluminum / Vinyl	1	1951	1035	1	3	1
1900	14	Stucco	1	1951	1014	1	3	2
1901	14	Aluminum / Vinyl	1	1944	1004	1	3	1

1902	14	Frame	1	1950	885	1	3	1
1903	14	Aluminum / Vinyl	1	1947	828	1	2	1
1904	14	Stone	2	1921	2352	1	3	1
1905	14	Aluminum / Vinyl	2	1940	1426	1	4	2
1906	14	Aluminum / Vinyl	2	2004	1380	1	3	2
1907	14	Aluminum / Vinyl	2	1949	1374	1	3	1
1908	14	Aluminum / Vinyl	1.5	1927	1721	2	4	2
1909	14	Aluminum / Vinyl	1.5	1949	1533	2	3	2
1910	14	Frame	2	1906	2612	2	6	2
1911	14	Aluminum / Vinyl	1	1944	1950	2	6	2
1912	14	Aluminum / Vinyl	1.5	1928	1946	1	4	2
1913	14	Brick	1	1928	1541	1	4	1
1914	14	Aluminum / Vinyl	1	1923	1218	1	3	2
1915	14	Aluminum / Vinyl	1	1925	1149	1	3	1
1916	14	Aluminum / Vinyl	1	1953	976	1	3	1
1917	14	Aluminum / Vinyl	1	1943	803	1	2	1
1918	14	Aluminum / Vinyl	1	1947	784	1	2	1
1919	14	Frame	1	1943	689	1	2	1
1920	14	Aluminum / Vinyl	1.5	1915	1612	1	3	1
1921	14	Frame	1	1908	1428	1	5	1
1922	14	Frame	1	1901	1406	1	3	1
1923	14	Aluminum / Vinyl	1	1920	1199	1	4	1
1924	14	Aluminum / Vinyl	1	1900	1169	1	3	1
1925	14	Aluminum / Vinyl	1	1901	1107	1	4	1
1926	14	Aluminum / Vinyl	1	1915	1064	1	3	1
1927	14	Aluminum / Vinyl	1	1924	1040	1	3	2
1928	14	Aluminum / Vinyl	1	1897	1032	1	3	1
1929	14	Aluminum / Vinyl	2	1910	1980	3	4	3
1930	15	Brick	2	1916	3354	2	4	2
1931	1	Aluminum / Vinyl	1.5	1950	1324	1	4	2
1932	1	Frame	1.5	1935	1308	1	3	1
1933	1	Frame	1.5	1940	1228	1	3	1
1934	1	Aluminum / Vinyl	1.5	1940	1191	1	4	2
1935	1	Aluminum / Vinyl	1	1947	821	1	3	1
1936	1	Aluminum / Vinyl	2	1954	1536	2	4	2
1937	1	Aluminum / Vinyl	1	1960	1104	1	3	1
1938	1	Aluminum / Vinyl	1	1955	1082	1	3	1
1939	1	Aluminum / Vinyl	1	1949	868	1	2	1
1940	1	Aluminum / Vinyl	1	1949	868	1	2	1
1941	1	Aluminum / Vinyl	1	1942	796	1	2	1
1942	1	Aluminum / Vinyl	1	1941	720	1	2	2
1943	2	Brick	1	1957	1638	1	4	1
1944	2	Aluminum / Vinyl	1	1940	1385	1	4	1

1945	2	Brick	1	1955	1317	1	3	1
1946	2	Aluminum / Vinyl	1	1955	1056	1	3	1
1947	2	Aluminum / Vinyl	1	1947	975	1	4	1
1948	2	Aluminum / Vinyl	1.5	1957	1764	2	5	2
1949	2	Brick	1	1957	1242	1	3	1
1950	2	Brick	1	1960	1204	1	3	1
1951	2	Frame	1	1960	1188	1	3	1
1952	2	Frame	1	1956	1188	1	3	1
1953	2	Brick	1	1958	1177	1	3	1
1954	2	Frame	1	1971	1156	1	4	1
1955	2	Brick	1	1955	1115	1	2	1
1956	2	Brick	1	1958	1090	1	3	1
1957	2	Aluminum / Vinyl	1	1966	1083	1	3	1
1958	2	Frame	1	1955	1046	1	3	1
1959	2	Frame	1	1975	1023	1	3	1
1960	2	Brick	1	1955	1020	1	2	1
1961	2	Aluminum / Vinyl	1	1953	970	1	3	1
1962	2	Aluminum / Vinyl	1	1953	970	1	3	1
1963	2	Brick	1	1957	942	1	3	1
1964	2	Aluminum / Vinyl	1	1956	936	1	2	1
1965	2	Frame	1	1957	925	1	3	1
1966	2	Aluminum / Vinyl	1	1958	918	1	2	1
1967	2	Aluminum / Vinyl	1	1956	905	1	3	1
1968	2	Aluminum / Vinyl	1	1957	900	1	3	1
1969	2	Aluminum / Vinyl	1	1957	900	1	3	1
1970	2	Aluminum / Vinyl	1	1955	864	1	3	1
1971	2	Aluminum / Vinyl	1	1955	756	1	2	1
1972	2	Aluminum / Vinyl	1	1955	1414	1	3	1
1973	2	Aluminum / Vinyl	1	1955	1414	1	3	2
1974	2	Masonry / Frame	2	1963	2281	2	6	2
1975	3	Brick	2	1924	3595	1	7	3
1976	3	Frame	2	1924	2944	2	6	2
1977	3	Aluminum / Vinyl	1	1949	1413	2	3	2
1978	3	Stucco	2	1910	3886	2	6	3
1979	3	Frame	2	1908	1710	2	4	2
1980	3	Stone	2	1915	6917	3	6	4
1981	3	Frame	2	1898	3033	1	5	2
1982	3	Aluminum / Vinyl	2	1900	1800	1	3	2
1983	3	Frame	1	1917	1559	1	3	1
1984	3	Brick	1	1905	948	1	3	1
1985	4	Aluminum / Vinyl	1	1924	1282	1	4	1
1986	4	Aluminum / Vinyl	2	1890	2133	1	5	2
1987	4	Frame	2	1904	1772	1	4	1

1988	5	Brick	1	1954	1622	1	4	2
1989	5	Brick	1.5	1956	1589	1	3	1
1990	5	Brick	1	1953	1584	1	4	1
1991	5	Stone	1.5	1948	1558	1	3	1
1992	5	Brick	1	1953	1512	1	4	1
1993	5	Brick	1	1955	1501	1	3	1
1994	5	Stone	1.5	1948	1432	1	3	1
1995	5	Stone	1	1950	1415	1	3	1
1996	5	Masonry / Frame	1	1952	1352	1	3	2
1997	5	Aluminum / Vinyl	1	1949	1293	1	3	2
1998	5	Frame	1	1952	1292	1	3	1
1999	5	Brick	1	1955	1242	1	3	1
2000	5	Aluminum / Vinyl	1	1954	1219	1	3	1
2001	5	Aluminum / Vinyl	1	1955	1200	1	4	1
2002	5	Aluminum / Vinyl	1	1952	1120	1	3	1
2003	5	Brick	1	1946	1111	1	3	1
2004	5	Brick	1.5	1947	1109	1	3	1
2005	5	Aluminum / Vinyl	1	1950	1092	1	3	1
2006	5	Aluminum / Vinyl	1	1953	1035	1	4	1
2007	5	Aluminum / Vinyl	1	1952	1034	1	3	1
2008	5	Aluminum / Vinyl	1	1951	933	1	3	1
2009	5	Aluminum / Vinyl	1	1957	784	1	2	1
2010	5	Aluminum / Vinyl	2	2008	2473	1	4	2
2011	5	Brick	2	1957	2422	2	6	2
2012	5	Brick	1	1964	2065	2	4	2
2013	5	Aluminum / Vinyl	2	1953	2016	2	4	2
2014	5	Aluminum / Vinyl	1	2009	2482	1	3	2
2015	5	Aluminum / Vinyl	1	1960	1595	1	3	2
2016	5	Brick	1	1957	1509	1	3	1
2017	5	Brick	1	1969	1360	1	3	1
2018	5	Stone	1	1956	1342	1	2	2
2019	5	Aluminum / Vinyl	1	1959	1287	1	3	1
2020	5	Aluminum / Vinyl	1	1955	1256	1	3	2
2021	5	Aluminum / Vinyl	1	1969	1228	1	3	1
2022	5	Aluminum / Vinyl	1	1954	1176	1	3	1
2023	5	Brick	1	1956	1144	1	3	1
2024	5	Brick	1	1958	1120	1	3	1
2025	5	Frame	1	1952	1106	1	2	1
2026	5	Aluminum / Vinyl	1	1955	1077	1	3	2
2027	5	Brick	1	1958	1058	1	3	1
2028	5	Aluminum / Vinyl	1	1956	1044	1	3	1
2029	5	Brick	1	1956	1040	1	2	1
2030	5	Frame	1	1950	1032	1	3	1

2031	5	Brick	1	1956	1025	1	3	1
2032	5	Brick	1	1955	1019	1	3	1
2033	5	Aluminum / Vinyl	1	1956	999	1	3	1
2034	5	Brick	1	1947	936	1	2	1
2035	5	Brick	1	1947	863	1	2	1
2036	5	Aluminum / Vinyl	1	1949	772	1	2	1
2037	5	Aluminum / Vinyl	1	1949	707	1	2	1
2038	5	Aluminum / Vinyl	1.5	1926	1083	1	2	1
2039	5	Masonry / Frame	2	1965	2222	2	6	2
2040	6	Aluminum / Vinyl	2	2008	1536	1	3	1
2041	6	Frame	1	1880	1134	1	3	2
2042	6	Frame	2	1922	2626	2	6	2
2043	6	Aluminum / Vinyl	1.5	1926	1919	2	4	2
2044	6	Frame	1.5	1926	1737	2	4	2
2045	6	Brick	2	1914	2628	2	5	2
2046	6	Aluminum / Vinyl	2	1906	2488	2	4	2
2047	6	Frame	1	1927	1547	1	3	2
2048	6	Aluminum / Vinyl	1	1925	1718	1	4	1
2049	6	Frame	1.5	1902	1589	1	4	1
2050	6	Aluminum / Vinyl	1	1911	1392	1	5	1
2051	6	Frame	1	1904	944	1	2	1
2052	6	Frame	2	1904	3482	3	7	3
2053	7	Brick	1.5	1940	1612	1	3	1
2054	7	Aluminum / Vinyl	1.5	1943	1499	1	3	1
2055	7	Brick	1.5	1939	1488	1	4	1
2056	7	Aluminum / Vinyl	1	1941	1446	1	4	1
2057	7	Brick	1	1946	1349	1	4	2
2058	7	Stone	1	1941	1311	1	3	1
2059	7	Masonry / Frame	2	1950	1970	1	4	1
2060	7	Stone	2	1940	2692	2	4	2
2061	7	Aluminum / Vinyl	1.5	1922	1613	1	4	1
2062	7	Aluminum / Vinyl	1	1928	1475	1	3	1
2063	7	Aluminum / Vinyl	1	1927	1344	1	3	1
2064	7	Aluminum / Vinyl	1	1927	1344	1	3	1
2065	7	Aluminum / Vinyl	1	1927	1264	1	3	1
2066	7	Aluminum / Vinyl	1	1924	1191	1	3	1
2067	7	Aluminum / Vinyl	1	1928	1170	1	2	1
2068	7	Aluminum / Vinyl	1	1950	1101	1	2	2
2069	7	Brick	1	1955	1074	1	2	1
2070	8	Aluminum / Vinyl	1	1891	1080	1	2	2
2071	8	Aluminum / Vinyl	2	1913	2772	2	6	2
2072	8	Aluminum / Vinyl	2	1905	2684	2	8	2
2073	8	Frame	2	1923	2388	2	4	2

2074	8	Frame	1.5	1904	2063	2	4	2
2075	8	Aluminum / Vinyl	2	1884	1632	2	4	2
2076	8	Brick	1	1923	1589	1	3	1
2077	8	Aluminum / Vinyl	1	1924	843	1	2	1
2078	8	Aluminum / Vinyl	2	1912	1675	1	3	2
2079	8	Aluminum / Vinyl	1.5	1916	1549	1	2	1
2080	8	Aluminum / Vinyl	1	1900	1188	1	4	1
2081	8	Aluminum / Vinyl	1	1898	1008	1	3	1
2082	9	Aluminum / Vinyl	1	1954	1084	1	3	1
2083	9	Aluminum / Vinyl	1	1953	1020	1	4	1
2084	9	Aluminum / Vinyl	2	2001	2868	1	4	2
2085	9	Aluminum / Vinyl	2	2009	2350	1	4	2
2086	9	Aluminum / Vinyl	2	1954	1728	2	6	2
2087	9	Aluminum / Vinyl	1	1930	1223	1	3	1
2088	9	Frame	1	1966	1420	1	3	1
2089	9	Aluminum / Vinyl	1	1975	1227	1	3	1
2090	9	Brick	1	1963	1191	1	3	1
2091	9	Aluminum / Vinyl	1	1958	1175	1	3	1
2092	9	Aluminum / Vinyl	1	1980	1077	1	3	1
2093	9	Aluminum / Vinyl	1	1963	1074	1	3	1
2094	9	Aluminum / Vinyl	1	1957	1059	1	3	1
2095	9	Brick	1	1957	1050	1	3	2
2096	9	Aluminum / Vinyl	1	1964	1038	1	3	1
2097	9	Brick	1	1963	1032	1	3	1
2098	9	Aluminum / Vinyl	1	1976	906	1	3	1
2099	9	Masonry / Frame	1	1959	1598	1	4	1
2100	10	Aluminum / Vinyl	1	1942	1055	1	3	1
2101	10	Aluminum / Vinyl	1	1946	1663	1	3	1
2102	10	Aluminum / Vinyl	1	1952	1389	1	3	1
2103	10	Aluminum / Vinyl	1.5	1950	1323	1	5	2
2104	10	Aluminum / Vinyl	1	1953	1185	1	4	1
2105	10	Aluminum / Vinyl	1	1953	1110	1	4	1
2106	10	Brick	2	1929	3219	2	6	2
2107	10	Frame	2	1914	2752	2	6	3
2108	10	Aluminum / Vinyl	2	1920	2556	2	4	2
2109	10	Masonry / Frame	2	1927	2528	2	4	2
2110	10	Frame	2	1914	2294	2	4	2
2111	10	Aluminum / Vinyl	1.5	1959	1812	2	3	2
2112	10	Brick	1	1923	2589	1	4	2
2113	10	Brick	1	1930	2368	1	4	1
2114	10	Frame	1	1919	1768	1	4	2
2115	10	Aluminum / Vinyl	1	1928	1470	1	4	2
2116	10	Aluminum / Vinyl	1	1926	1417	1	3	1

2117	10	Aluminum / Vinyl	1	1925	1416	1	3	2
2118	10	Aluminum / Vinyl	1	1924	1395	1	3	1
2119	10	Aluminum / Vinyl	1	1930	1345	1	4	1
2120	10	Aluminum / Vinyl	1	1923	1274	1	2	1
2121	10	Aluminum / Vinyl	1	1923	921	1	2	1
2122	10	Aluminum / Vinyl	1	1953	1384	1	3	2
2123	10	Frame	1	1956	936	1	2	1
2124	10	Aluminum / Vinyl	1	1942	924	1	2	1
2125	10	Aluminum / Vinyl	1	1946	667	1	2	1
2126	10	Aluminum / Vinyl	1	1914	1593	1	3	2
2127	10	Frame	1.5	1896	1492	1	3	2
2128	10	Aluminum / Vinyl	1	1921	1080	1	2	1
2129	10	Frame	1	1926	990	1	3	1
2130	11	Fiber-Cement	1	2008	4405	1	6	4
2131	11	Stone	1	1948	1767	1	4	1
2132	11	Aluminum / Vinyl	1	1970	1548	1	4	2
2133	11	Aluminum / Vinyl	1	1951	1424	1	3	1
2134	11	Brick	1	1942	1316	1	3	1
2135	11	Aluminum / Vinyl	1	1959	1159	1	4	1
2136	11	Aluminum / Vinyl	1	1949	1107	1	3	1
2137	11	Brick	1	1942	980	1	2	1
2138	11	Aluminum / Vinyl	2	1958	2233	2	6	2
2139	11	Aluminum / Vinyl	2	1954	1728	2	6	2
2140	11	Aluminum / Vinyl	1	1928	1188	1	3	2
2141	11	Frame	1	1954	1364	1	3	1
2142	11	Brick	1	1953	1313	1	3	1
2143	11	Aluminum / Vinyl	1	1948	1270	1	2	1
2144	11	Brick	1	1967	1235	1	3	1
2145	11	Brick	1	1960	1201	1	3	1
2146	11	Frame	1	1953	1173	1	3	1
2147	11	Brick	1	1956	1161	1	3	1
2148	11	Brick	1	1966	1158	1	3	1
2149	11	Brick	1	1953	1140	1	2	1
2150	11	Brick	1	1959	1118	1	3	1
2151	11	Aluminum / Vinyl	1	1965	1107	1	3	1
2152	11	Aluminum / Vinyl	1	1964	1105	1	3	1
2153	11	Brick	1	1956	1093	1	2	1
2154	11	Aluminum / Vinyl	1	1958	1080	1	3	2
2155	11	Aluminum / Vinyl	1	1949	1072	1	2	1
2156	11	Aluminum / Vinyl	1	1958	1048	1	3	1
2157	11	Brick	1	1955	1013	1	3	1
2158	11	Brick	1	1959	1011	1	3	1
2159	11	Brick	1	1954	999	1	2	1

2160	11	Brick	1	1952	980	1	2	1
2161	11	Aluminum / Vinyl	1	1953	966	1	3	1
2162	11	Brick	1	1959	965	1	3	1
2163	11	Brick	1	1956	947	1	3	1
2164	11	Frame	1	1955	944	1	3	1
2165	11	Brick	1	1955	925	1	3	1
2166	11	Aluminum / Vinyl	1	1960	925	1	3	1
2167	11	Aluminum / Vinyl	1	1960	896	1	3	2
2168	11	Aluminum / Vinyl	1	1953	882	1	3	1
2169	11	Aluminum / Vinyl	1	1953	876	1	3	1
2170	11	Aluminum / Vinyl	1	1953	874	1	3	1
2171	11	Frame	1	1953	864	1	3	1
2172	11	Aluminum / Vinyl	1	1938	756	1	2	1
2173	12	Frame	2	1966	1821	1	4	1
2174	12	Frame	2	1910	1980	2	4	2
2175	12	Aluminum / Vinyl	2	1885	1968	2	4	2
2176	12	Aluminum / Vinyl	1	1893	1699	1	4	2
2177	12	Aluminum / Vinyl	1.5	1906	1600	2	4	2
2178	12	Frame	1.5	1910	1402	1	4	1
2180	13	Aluminum / Vinyl	1	1974	1789	1	4	2
2181	13	Brick	1	1948	1528	1	4	1
2182	13	Brick	1	1953	1252	1	3	1
2183	13	Brick	1	1948	1249	1	4	1
2184	13	Aluminum / Vinyl	1	1950	1143	1	3	2
2185	13	Aluminum / Vinyl	1	1951	1093	1	4	1
2186	13	Brick	1	1951	1090	1	2	1
2187	13	Brick	1	1946	1076	1	3	1
2188	13	Aluminum / Vinyl	1	1942	1074	1	4	1
2189	13	Aluminum / Vinyl	1	1949	830	1	2	1
2190	13	Aluminum / Vinyl	2	1998	2259	1	3	2
2191	13	Frame	2	1979	1943	1	3	2
2192	13	Stone	2	1936	1386	1	2	1
2193	13	Aluminum / Vinyl	1.5	1936	821	1	3	1
2194	13	Aluminum / Vinyl	2	1957	2209	2	6	2
2195	13	Brick	1.5	1951	2082	2	4	2
2196	13	Aluminum / Vinyl	1	1955	1209	1	3	1
2197	13	Stone	1	1939	1177	1	2	1
2198	13	Aluminum / Vinyl	1	1981	1156	1	3	1
2199	13	Brick	1	1960	1148	1	3	1
2200	13	Brick	1	1961	1090	1	3	1
2201	13	Brick	1	1962	1037	1	3	1
2202	13	Stone	1	1950	1026	1	2	1
2203	13	Aluminum / Vinyl	1	1960	998	1	3	1

2204	13	Aluminum / Vinyl	1	1949	912	1	3	1
2205	13	Aluminum / Vinyl	1	1951	840	1	2	1
2206	13	Aluminum / Vinyl	1	1950	812	1	2	1
2207	13	Frame	1	1942	745	1	2	1
2208	13	Aluminum / Vinyl	1	1953	743	1	2	1
2209	13	Aluminum / Vinyl	1	1943	707	1	2	1
2210	13	Brick	2	1929	2236	1	3	1
2211	13	Aluminum / Vinyl	2	1966	2132	2	6	2
2212	14	Aluminum / Vinyl	1	1953	1182	1	4	1
2213	14	Brick	1	1956	945	1	3	1
2214	14	Aluminum / Vinyl	2	1939	1144	1	2	1
2215	14	Aluminum / Vinyl	1	1890	1256	1	4	1
2216	14	Aluminum / Vinyl	1	1906	1225	1	3	1
2217	14	Frame	1	1900	1027	1	4	1
2218	14	Aluminum / Vinyl	1.5	1926	1688	2	4	2
2219	14	Aluminum / Vinyl	2	1978	2054	2	6	2
2220	14	Aluminum / Vinyl	2	1920	2670	2	5	2
2221	14	Aluminum / Vinyl	2	1915	2268	2	4	2
2222	14	Frame	2	1915	1832	2	4	2
2223	14	Brick	1.5	1926	1644	1	4	1
2224	14	Brick	1	1930	1618	1	4	2
2225	14	Aluminum / Vinyl	1	1921	1185	1	3	1
2226	14	Brick	1	1902	1048	1	3	1
2227	14	Brick	1	1953	1115	1	3	1
2228	14	Aluminum / Vinyl	1	1955	936	1	3	1
2229	14	Aluminum / Vinyl	1	1943	689	1	2	1
2230	14	Aluminum / Vinyl	1.5	1890	1759	1	5	1
2231	14	Aluminum / Vinyl	2	1924	1663	1	4	1
2232	14	Frame	1	1918	1025	1	2	1
2233	14	Aluminum / Vinyl	1	1920	880	1	2	1
2234	14	Aluminum / Vinyl	1	1900	810	1	2	1
2235	14	Aluminum / Vinyl	1	1921	775	1	2	1
2237	14	Aluminum / Vinyl	2	1953	1256	1	3	1
2238	15	Aluminum / Vinyl	2	2006	1458	1	3	1
2239	15	Frame	2	1921	2984	2	8	2
2240	15	Aluminum / Vinyl	2	1914	2536	2	6	2
2241	15	Aluminum / Vinyl	2	1914	2536	2	6	2
2242	15	Aluminum / Vinyl	2	1914	2532	2	6	2
2243	15	Aluminum / Vinyl	2	1910	2182	2	4	2
2244	15	Aluminum / Vinyl	1.5	1899	1749	2	4	3
2245	15	Aluminum / Vinyl	2	1922	2008	1	3	2
2246	15	Frame	1.5	1920	1548	1	4	1
2247	1	Aluminum / Vinyl	1.5	1940	1367	1	3	1

2248	1	Brick	1	1937	1288	1	4	1
2249	1	Aluminum / Vinyl	1	1954	1194	1	4	1
2250	1	Frame	1	1953	1096	1	4	1
2251	1	Frame	1	1940	944	1	4	1
2252	1	Aluminum / Vinyl	1	1943	884	1	3	1
2253	1	Aluminum / Vinyl	1	1925	894	1	2	1
2254	1	Aluminum / Vinyl	2	1918	2440	2	6	2
2255	1	Aluminum / Vinyl	1	1966	1578	1	3	1
2256	1	Aluminum / Vinyl	1	2007	1439	1	3	2
2257	1	Aluminum / Vinyl	1	1951	938	1	3	1
2258	1	Frame	1	1952	831	1	3	1
2259	2	Aluminum / Vinyl	1	1955	1253	1	4	2
2260	2	Aluminum / Vinyl	1	1953	1118	1	4	1
2261	2	Frame	1	1954	1053	1	3	1
2262	2	Aluminum / Vinyl	2	1968	1465	1	4	1
2263	2	Frame	2	1960	1387	1	3	1
2264	2	Frame	2	1956	2200	2	6	2
2265	2	Aluminum / Vinyl	2	1960	2070	2	6	2
2266	2	Aluminum / Vinyl	2	1958	1879	2	4	2
2267	2	Aluminum / Vinyl	2	1939	1650	2	4	2
2268	2	Aluminum / Vinyl	1	1930	1545	2	3	2
2269	2	Aluminum / Vinyl	1	1955	1465	1	4	1
2270	2	Brick	1	1956	1243	1	3	1
2271	2	Brick	1	1954	1046	1	3	1
2272	2	Aluminum / Vinyl	1	1958	1039	1	3	1
2273	2	Aluminum / Vinyl	1	1948	985	1	2	1
2274	2	Frame	1	1958	973	1	3	1
2275	2	Aluminum / Vinyl	1	1955	963	1	3	1
2276	2	Aluminum / Vinyl	1	1958	936	1	3	1
2277	2	Aluminum / Vinyl	1	1958	936	1	3	1
2278	2	Aluminum / Vinyl	1	1958	918	1	3	2
2279	2	Aluminum / Vinyl	1	1955	864	1	3	1
2280	2	Aluminum / Vinyl	1	1951	723	1	2	1
2281	2	Frame	1	1955	672	1	2	1
2282	3	Frame	2	1906	1352	1	4	1
2283	3	Masonry / Frame	2	1916	3476	2	6	2
2284	3	Aluminum / Vinyl	2	1923	2546	2	6	2
2285	3	Brick	1.5	1931	2234	2	4	2
2286	3	Frame	1.5	1926	1880	2	3	2
2287	3	Aluminum / Vinyl	1	1900	1525	2	2	2
2288	3	Frame	2	1904	2899	1	4	2
2289	3	Frame	2	1899	2350	1	4	1
2290	3	Aluminum / Vinyl	1.5	1900	1549	1	4	1

2292	4	Aluminum / Vinyl	2	2007	1879	1	4	2
2293	4	Frame	2	1900	3046	2	6	2
2294	4	Brick	2	1921	2806	2	4	2
2295	5	Brick	1	1952	1502	1	3	1
2296	5	Aluminum / Vinyl	1	1949	1279	1	4	1
2297	5	Aluminum / Vinyl	1	1952	1098	1	4	2
2298	5	Aluminum / Vinyl	2	1997	1976	1	3	2
2299	5	Aluminum / Vinyl	2	1940	1396	1	3	1
2300	5	Brick	1	1955	3025	2	5	2
2301	5	Brick	2	1959	2598	2	6	3
2302	5	Aluminum / Vinyl	1.5	1965	2479	2	6	2
2303	5	Frame	1	1968	1500	1	3	1
2304	5	Brick	1	1957	1335	1	3	1
2305	5	Brick	1	1957	1335	1	3	1
2306	5	Aluminum / Vinyl	1	1993	1226	1	3	2
2307	5	Brick	1	1964	1191	1	3	2
2308	5	Brick	1	1952	1064	1	2	1
2309	5	Aluminum / Vinyl	1	1955	948	1	3	1
2310	5	Aluminum / Vinyl	1	1952	901	1	2	1
2311	6	Aluminum / Vinyl	2	1907	2028	2	4	2
2312	6	Aluminum / Vinyl	1.5	1890	1622	1	4	1
2313	7	Brick	1	1951	1709	1	4	2
2314	7	Aluminum / Vinyl	1	1936	1628	1	5	1
2315	7	Stone	1	1947	1416	1	3	1
2316	7	Aluminum / Vinyl	1.5	1925	2064	2	4	2
2317	7	Frame	2	1927	2068	2	4	2
2318	7	Aluminum / Vinyl	1.5	1906	1816	2	4	2
2319	7	Frame	1.5	1894	1678	2	4	2
2320	7	Aluminum / Vinyl	1	1926	1392	1	4	1
2321	7	Aluminum / Vinyl	1	1920	1218	1	3	1
2322	7	Brick	1.5	1931	1717	1	4	1
2323	7	Frame	1	1910	1386	1	4	1
2324	7	Aluminum / Vinyl	1.5	1928	1164	1	4	1
2325	7	Stone	1	1933	1816	1	3	1
2326	8	Frame	2	1922	1824	1	4	1
2327	8	Frame	2	1922	1824	1	4	1
2328	8	Aluminum / Vinyl	2	1971	1169	1	4	1
2329	8	Frame	1.5	1924	1963	2	5	3
2330	8	Aluminum / Vinyl	2	1908	2300	2	6	2
2331	8	Frame	2	1894	1904	2	4	2
2332	8	Aluminum / Vinyl	1.5	1926	1819	2	4	2
2333	8	Aluminum / Vinyl	1.5	1903	1603	2	4	2
2334	8	Aluminum / Vinyl	1	1890	1999	1	5	2

2335	8	Aluminum / Vinyl	1.5	1910	1843	1	3	1
2336	8	Aluminum / Vinyl	2	1911	1649	1	5	1
2337	8	Aluminum / Vinyl	1	1880	1260	1	4	2
2338	8	Frame	1	1898	1200	1	3	1
2341	8	Stucco	2	1880	2779	3	6	2
2342	9	Aluminum / Vinyl	1	1948	637	1	2	1
2343	9	Masonry / Frame	1	1959	1610	1	4	3
2344	9	Aluminum / Vinyl	1	1973	1596	1	4	1
2345	9	Aluminum / Vinyl	1	1978	1467	1	3	1
2346	9	Aluminum / Vinyl	1	1965	1227	1	3	1
2347	9	Stone	1	1949	1086	1	2	2
2348	9	Aluminum / Vinyl	1	1951	960	1	3	1
2349	9	Aluminum / Vinyl	1	1951	792	1	2	1
2350	9	Aluminum / Vinyl	1	1951	792	1	2	1
2351	9	Masonry / Frame	2	1971	3500	3	>8	3
2352	10	Masonry / Frame	2	1932	1772	1	3	1
2353	10	Aluminum / Vinyl	1	1910	944	1	3	1
2354	10	Aluminum / Vinyl	2	1922	2903	2	6	2
2355	10	Frame	2	1928	2236	2	4	2
2356	10	Aluminum / Vinyl	2	1924	1924	2	4	2
2357	10	Brick	2	1912	3048	2	6	2
2358	10	Brick	1.5	1938	2150	2	4	2
2359	10	Frame	1.5	1924	1413	2	3	2
2360	10	Aluminum / Vinyl	1	1924	1865	1	4	1
2361	10	Aluminum / Vinyl	1	1926	982	1	3	1
2362	10	Aluminum / Vinyl	1	1926	982	1	3	1
2363	10	Aluminum / Vinyl	1	1926	916	1	3	2
2364	10	Aluminum / Vinyl	1	1951	936	1	3	1
2365	10	Aluminum / Vinyl	1	1955	864	1	3	1
2366	10	Aluminum / Vinyl	1	1940	772	1	2	1
2367	10	Stucco	2	1916	2266	1	4	2
2368	10	Aluminum / Vinyl	1	1925	616	1	2	1
2369	11	Brick	1.5	1959	1850	1	4	1
2370	11	Frame	1	1952	1077	1	4	1
2371	11	Stucco	1	1939	980	1	3	1
2372	11	Brick	2	1962	3044	2	6	2
2373	11	Aluminum / Vinyl	2	1953	1554	2	4	2
2374	11	Frame	1	1953	1176	1	3	1
2375	11	Aluminum / Vinyl	1	1955	1012	1	3	2
2376	11	Brick	1	1959	1010	1	3	1
2377	11	Brick	1	1957	967	1	2	1
2378	11	Brick	1	1956	967	1	3	1
2379	11	Aluminum / Vinyl	1	1954	867	1	2	1

2380	11	Aluminum / Vinyl	1	1949	846	1	2	1
2381	11	Aluminum / Vinyl	1	1954	735	1	2	1
2382	12	Aluminum / Vinyl	1	1906	1870	2	5	2
2383	12	Aluminum / Vinyl	1	1923	1001	1	2	1
2384	13	Aluminum / Vinyl	1	1948	1516	1	4	2
2385	13	Aluminum / Vinyl	1	1947	1144	1	4	1
2386	13	Brick	1	1946	994	1	3	1
2387	13	Brick	1	1949	942	1	2	1
2388	13	Aluminum / Vinyl	2	1985	2084	1	3	2
2389	13	Aluminum / Vinyl	1	1918	759	1	3	1
2390	13	Frame	2	1956	2098	2	5	2
2391	14	Aluminum / Vinyl	1	1953	1153	1	3	1
2392	14	Brick	1	1953	1125	1	4	1
2393	14	Aluminum / Vinyl	2	1940	1590	1	3	1
2394	14	Stucco	2	1945	1006	1	2	1
2395	14	Frame	2	1918	1964	2	4	2
2396	14	Frame	2	1924	1848	2	4	2
2397	14	Aluminum / Vinyl	1	1926	1758	1	4	1
2398	14	Frame	1.5	1925	1658	1	4	2
2399	14	Aluminum / Vinyl	1	1926	1567	1	4	1
2400	14	Aluminum / Vinyl	1	1926	1350	1	3	1
2401	14	Aluminum / Vinyl	1	1953	704	1	2	1
2402	14	Stucco	1	1913	1594	1	4	2
2403	14	Aluminum / Vinyl	1.5	1911	1526	1	4	2
2404	14	Aluminum / Vinyl	1.5	1907	1298	1	3	1
2405	14	Aluminum / Vinyl	1	1915	951	1	3	1
2406	15	Frame	1	1895	1032	1	2	1
2407	15	Aluminum / Vinyl	2	1917	2590	2	6	2
2408	15	Frame	2	1908	2491	2	6	2
2409	15	Frame	2	1908	2491	2	6	2
2410	15	Aluminum / Vinyl	2	1922	2378	2	4	2
2411	15	Aluminum / Vinyl	2	1901	2320	2	6	2
2412	15	Aluminum / Vinyl	2	1911	2068	2	4	2
2413	15	Aluminum / Vinyl	1	1921	1107	1	4	1
2414	1	Stone	1	1935	1779	1	5	2
2415	1	Brick	1	1963	1529	1	3	1
2416	3	Brick	2	1902	4050	1	7	4
2417	3	Aluminum / Vinyl	2	1896	2644	2	5	2
2418	3	Masonry / Frame	2	1901	4899	1	5	>4
2419	3	Brick	2	1910	2694	1	5	2
2420	3	Stucco	2	1908	2582	1	3	2
2421	5	Brick	1	1957	2735	1	5	2
2422	5	Stucco	1	1946	1134	1	3	1

2423	5	Aluminum / Vinyl	1	1948	1048	1	2	1
2424	5	Aluminum / Vinyl	1	1952	1008	1	3	1
2425	5	Stone	2	1947	2430	2	4	2
2426	5	Brick	1	1958	1404	1	3	1
2427	5	Aluminum / Vinyl	1	1952	1096	1	3	1
2428	6	Aluminum / Vinyl	2	2004	2204	1	3	2
2429	6	Aluminum / Vinyl	2	1897	2025	2	6	2
2430	6	Frame	1.5	1913	1528	2	4	2
2431	7	Stone	1	1936	1364	1	3	1
2432	7	Aluminum / Vinyl	2	1946	1456	1	3	1
2433	7	Brick	1	1929	1333	1	3	1
2434	8	Aluminum / Vinyl	1.5	1915	1688	1	3	1
2435	8	Aluminum / Vinyl	2	1900	2727	3	8	3
2436	10	Brick	1	1953	1222	1	4	1
2437	10	Aluminum / Vinyl	1	1920	608	1	1	1
2438	10	Aluminum / Vinyl	1.5	1926	1594	2	3	2
2439	11	Aluminum / Vinyl	1	1953	1318	1	3	1
2440	11	Aluminum / Vinyl	1	1937	917	1	2	1
2441	11	Masonry / Frame	2	1974	2071	1	4	2
2442	11	Aluminum / Vinyl	2	1959	1912	2	6	2
2443	11	Frame	1	1937	1282	1	3	1
2444	11	Stone	1	1954	1278	1	2	1
2445	11	Aluminum / Vinyl	1	1972	1152	1	3	1
2446	11	Brick	1	1959	1137	1	3	1
2447	11	Aluminum / Vinyl	1	1960	1042	1	3	2
2448	11	Aluminum / Vinyl	1	1963	1000	1	3	1
2449	11	Brick	1	1956	982	1	2	2
2450	11	Aluminum / Vinyl	1	1942	826	1	2	1
2451	12	Aluminum / Vinyl	1	1883	611	1	1	1
2452	12	Frame	2	1905	2032	2	6	2
2453	13	Aluminum / Vinyl	1	1958	1447	1	4	2
2454	13	Aluminum / Vinyl	1	1953	1286	1	4	2
2455	13	Aluminum / Vinyl	1	1940	1152	1	3	1
2456	13	Brick	1	1948	818	1	2	1
2457	13	Aluminum / Vinyl	1.5	1928	1860	2	3	2
2458	13	Brick	1	1960	1150	1	3	1
2459	14	Aluminum / Vinyl	1	1949	771	1	2	1
2460	14	Brick	1.5	1927	2362	2	3	2
2461	14	Brick	1.5	1926	2204	2	4	2
2462	14	Frame	1.5	1921	2708	1	4	1
2463	14	Brick	1	1956	1082	1	3	1
2464	14	Aluminum / Vinyl	2	1905	1458	1	3	1
2465	14	Aluminum / Vinyl	2	1903	1402	1	4	1

2466	1	Frame	1	1925	909	1	2	1
2467	1	Brick	1	1952	1620	1	3	1
2468	1	Stucco	1	1951	1242	1	3	1
2469	2	Aluminum / Vinyl	1	1951	1164	1	3	1
2470	3	Frame	1	1900	912	1	3	1
2471	3	Aluminum / Vinyl	2	1913	1632	1	4	2
2472	3	Stucco	2	1924	2163	1	4	2
2473	3	Aluminum / Vinyl	1	1900	1963	1	4	2
2474	5	Fiber-Cement	1.5	2003	3229	1	4	3
2475	5	Brick	1	1948	1369	1	3	1
2476	5	Aluminum / Vinyl	1	1952	1221	1	3	1
2477	5	Brick	1.5	1947	1109	1	3	1
2478	5	Aluminum / Vinyl	1	1956	976	1	2	1
2479	5	Masonry / Frame	2	1948	2058	2	4	2
2480	5	Brick	1.5	1956	2003	2	5	2
2481	5	Stone	1	1953	1749	1	3	1
2482	5	Frame	1	1953	1489	1	3	1
2483	5	Brick	1	1956	1223	1	3	2
2484	5	Aluminum / Vinyl	1	1960	1183	1	4	1
2485	5	Aluminum / Vinyl	1	1959	1171	1	3	1
2486	5	Aluminum / Vinyl	1	1956	1148	1	3	1
2487	5	Aluminum / Vinyl	1	1955	1084	1	3	1
2488	5	Aluminum / Vinyl	1	1955	948	1	3	1
2489	5	Brick	1	1955	937	1	3	1
2490	6	Aluminum / Vinyl	2	1893	1884	2	4	2
2491	6	Aluminum / Vinyl	1	1925	1509	1	3	1
2492	6	Aluminum / Vinyl	1	1916	1474	1	3	1
2493	6	Aluminum / Vinyl	1	1895	1485	1	3	2
2494	6	Aluminum / Vinyl	1	1895	1433	1	5	1
2495	7	Brick	1	1938	1554	1	3	2
2496	7	Aluminum / Vinyl	1.5	1941	1424	1	3	1
2497	7	Aluminum / Vinyl	1	1952	1116	1	2	1
2498	8	Frame	2	1924	2500	2	6	2
2499	8	Frame	1.5	1915	2229	2	5	2
2500	8	Aluminum / Vinyl	2	1910	2032	2	4	2
2501	8	Aluminum / Vinyl	1	1928	1612	1	3	1
2502	8	Aluminum / Vinyl	2	1900	1685	1	4	1
2503	8	Frame	1.5	1908	1431	1	3	1
2504	8	Aluminum / Vinyl	1	1918	984	1	2	1
2505	9	Aluminum / Vinyl	2	2005	2236	1	4	2
2506	9	Aluminum / Vinyl	1	1957	1211	1	3	1
2507	10	Aluminum / Vinyl	1	1953	1185	1	4	1
2508	10	Frame	1	1905	1323	1	2	2

2509	10	Aluminum / Vinyl	2	1929	2617	2	5	2
2510	10	Aluminum / Vinyl	2	1924	2498	2	4	2
2511	10	Frame	2	1927	2087	2	4	2
2512	10	Aluminum / Vinyl	1.5	1968	2066	2	5	2
2513	10	Aluminum / Vinyl	1	1919	1743	1	4	1
2514	10	Aluminum / Vinyl	1	1926	1431	1	4	2
2515	10	Aluminum / Vinyl	1.5	1927	1265	1	3	1
2516	10	Brick	1	1958	1218	1	2	1
2517	10	Aluminum / Vinyl	1	1949	1056	1	3	1
2518	10	Aluminum / Vinyl	1	1954	1008	1	2	1
2519	10	Frame	1.5	1926	1208	1	3	1
2520	11	Brick	1	1953	1210	1	3	2
2521	11	Aluminum / Vinyl	1	1937	1432	1	3	2
2522	11	Aluminum / Vinyl	2	1954	1728	2	6	2
2523	11	Frame	2	1955	1586	2	4	2
2524	11	Brick	1	1957	1175	1	3	1
2525	11	Brick	1	1961	1102	1	2	2
2526	11	Aluminum / Vinyl	1	1956	995	1	3	1
2527	11	Aluminum / Vinyl	1	1955	890	1	3	1
2528	11	Aluminum / Vinyl	1	1939	775	1	2	1
2529	12	Frame	1	1892	1452	1	3	1
2530	13	Stone	1	1948	1378	1	4	2
2531	13	Aluminum / Vinyl	1	1936	1141	1	2	1
2532	13	Aluminum / Vinyl	1	1925	865	1	2	1
2533	13	Aluminum / Vinyl	1	1977	1493	1	3	1
2534	13	Aluminum / Vinyl	1	1960	985	1	3	1
2535	13	Aluminum / Vinyl	1	1948	804	1	2	1
2536	14	Aluminum / Vinyl	1	1938	1354	1	4	1
2537	14	Aluminum / Vinyl	2	1968	2112	2	6	2
2538	14	Aluminum / Vinyl	1	1926	1587	1	4	2
2539	14	Aluminum / Vinyl	1	1930	1431	1	3	3
2540	14	Aluminum / Vinyl	1	1923	974	1	3	1
2541	14	Aluminum / Vinyl	2	1880	1712	1	4	1
2542	14	Aluminum / Vinyl	1	1896	1320	1	3	2
2543	14	Aluminum / Vinyl	1	1900	1160	1	3	1
2544	15	Aluminum / Vinyl	2	2009	1860	1	4	3
2545	15	Aluminum / Vinyl	2	2001	1456	1	3	2
2546	1	Masonry / Frame	2	1946	1564	1	3	1
2547	1	Aluminum / Vinyl	1.5	1929	2184	2	4	2
2548	1	Aluminum / Vinyl	1	1928	1145	1	3	1
2549	1	Brick	1	1931	1028	1	3	2
2550	1	Frame	1	1925	908	1	3	1
2551	2	Brick	1	1946	1037	1	3	1

2552	2	Aluminum / Vinyl	1.5	1967	2628	2	6	4
2553	2	Frame	1	1964	1416	1	3	2
2554	2	Aluminum / Vinyl	1	1970	1148	1	4	1
2555	2	Aluminum / Vinyl	1	1957	1040	1	3	1
2556	2	Aluminum / Vinyl	1	1952	969	1	2	1
2557	2	Aluminum / Vinyl	1	1955	963	1	3	1
2558	2	Brick	1	1958	888	1	2	1
2559	2	Frame	1	1957	742	1	2	1
2560	2	Aluminum / Vinyl	1	1955	1514	1	4	1
2561	2	Masonry / Frame	1	1961	1485	1	4	1
2562	3	Aluminum / Vinyl	1.5	1925	2020	2	4	2
2563	3	Aluminum / Vinyl	2	1906	1870	2	4	3
2564	3	Brick	1	1923	1557	1	3	1
2565	3	Brick	2	1903	3874	1	5	3
2566	3	Aluminum / Vinyl	2	1897	2932	1	4	2
2567	3	Brick	2	1925	2329	1	4	2
2568	3	Aluminum / Vinyl	2	1899	2219	1	4	1
2569	3	Stucco	2	1912	2104	1	5	2
2570	3	Stucco	2	1920	1992	1	4	1
2571	5	Aluminum / Vinyl	1	1949	1515	1	3	1
2572	5	Aluminum / Vinyl	1	1951	1512	1	4	2
2573	5	Brick	1	1950	1512	1	4	1
2574	5	Brick	1.5	1937	1413	1	2	1
2575	5	Frame	1	1952	1292	1	3	1
2576	5	Brick	1	1955	1231	1	3	1
2577	5	Stone	1	1948	1213	1	3	1
2578	5	Aluminum / Vinyl	1	1951	1211	1	3	1
2579	5	Aluminum / Vinyl	1	1951	1170	1	3	2
2580	5	Brick	1	1955	1026	1	2	1
2581	5	Brick	2	1939	1472	1	3	1
2582	5	Aluminum / Vinyl	1	1989	1833	1	3	2
2583	5	Brick	1	1962	1265	1	3	1
2584	5	Brick	1	1954	1237	1	3	1
2585	5	Aluminum / Vinyl	1	1971	1230	1	3	1
2586	5	Brick	1	1957	1134	1	3	1
2587	5	Aluminum / Vinyl	1	1957	1120	1	3	1
2588	5	Brick	1	1955	1120	1	3	1
2589	5	Aluminum / Vinyl	1	1956	1019	1	3	1
2590	5	Aluminum / Vinyl	1	1962	946	1	3	1
2591	5	Aluminum / Vinyl	1	1954	872	1	3	1
2592	6	Aluminum / Vinyl	1	1941	1577	1	4	1
2593	6	Aluminum / Vinyl	1.5	1904	1737	2	4	2
2594	6	Aluminum / Vinyl	1.5	1904	1737	2	4	2

2595	7	Brick	1	1952	1324	1	3	1
2596	7	Aluminum / Vinyl	1	1947	1133	1	3	1
2597	7	Brick	1	1946	1088	1	3	1
2598	7	Aluminum / Vinyl	1	1946	996	1	3	2
2599	8	Aluminum / Vinyl	1	1883	994	1	2	1
2600	8	Aluminum / Vinyl	1.5	1928	1868	2	4	2
2601	8	Frame	2	1915	2947	2	>8	2
2602	8	Masonry / Frame	1.5	1935	1644	1	3	1
2603	9	Aluminum / Vinyl	1	1969	1453	1	3	1
2604	9	Aluminum / Vinyl	1	1966	1385	1	3	1
2605	9	Brick	1	1957	1222	1	3	1
2606	9	Aluminum / Vinyl	1	1978	1219	1	3	1
2607	9	Brick	1	1957	1050	1	3	1
2608	9	Aluminum / Vinyl	1	1956	946	1	3	1
2609	9	Brick	1	1958	924	1	3	1
2610	10	Brick	1.5	1937	1496	1	4	1
2611	10	Aluminum / Vinyl	1.5	1947	1481	1	5	2
2612	10	Block	1.5	1941	1415	1	3	1
2613	10	Aluminum / Vinyl	1	1947	1012	1	3	2
2614	10	Aluminum / Vinyl	1.5	1900	1342	1	2	1
2615	10	Brick	2	1954	3018	2	6	2
2616	10	Aluminum / Vinyl	1.5	1914	1822	2	4	2
2617	10	Aluminum / Vinyl	1	1920	1727	1	4	1
2618	10	Brick	1	1923	1659	1	3	2
2619	10	Aluminum / Vinyl	1	1923	1604	1	4	2
2620	10	Frame	1	1920	1569	1	4	2
2621	10	Frame	1	1924	1529	1	3	2
2622	10	Aluminum / Vinyl	1	1928	1470	1	4	2
2623	10	Aluminum / Vinyl	1	1924	1081	1	3	1
2624	10	Stucco	1.5	1916	1422	1	3	1
2625	11	Aluminum / Vinyl	1	1953	1411	1	4	1
2626	11	Aluminum / Vinyl	1	1943	1399	1	3	1
2627	11	Brick	1	1955	1151	1	3	2
2628	11	Aluminum / Vinyl	1	1944	1004	1	3	1
2629	11	Masonry / Frame	2	1955	1846	1	4	2
2630	11	Aluminum / Vinyl	1.5	1955	2083	2	5	2
2631	11	Aluminum / Vinyl	1	1952	1588	1	3	1
2632	11	Brick	1	1955	1208	1	3	2
2633	11	Aluminum / Vinyl	1	1960	1171	1	2	1
2634	11	Brick	1	1955	1150	1	3	1
2635	11	Aluminum / Vinyl	1	1959	1113	1	3	1
2636	11	Aluminum / Vinyl	1	1959	1093	1	3	1
2637	11	Aluminum / Vinyl	1	1955	999	1	2	1

2638	11	Aluminum / Vinyl	1	1953	971	1	3	1
2639	11	Aluminum / Vinyl	1	1954	943	1	3	1
2640	11	Brick	1	1956	938	1	2	1
2641	11	Aluminum / Vinyl	1	1950	833	1	2	1
2642	11	Aluminum / Vinyl	1	1945	717	1	2	1
2643	12	Aluminum / Vinyl	2	1891	2371	2	6	2
2644	12	Frame	1.5	1900	1804	2	4	2
2645	13	Prem Wood	1.5	1979	2074	1	4	3
2646	13	Aluminum / Vinyl	1.5	1940	1215	1	2	1
2647	13	Aluminum / Vinyl	2	1969	1973	1	3	1
2648	13	Brick	2	1960	2642	2	6	3
2649	13	Brick	1.5	1948	2081	2	4	2
2650	13	Brick	1.5	1958	2056	2	5	2
2651	13	Stone	1	1951	1541	1	3	2
2652	13	Brick	1	1963	1175	1	3	1
2653	13	Aluminum / Vinyl	1	1968	1092	1	3	1
2654	13	Aluminum / Vinyl	1	1960	981	1	3	1
2655	13	Aluminum / Vinyl	1	1950	922	1	3	1
2656	13	Aluminum / Vinyl	1	1954	720	1	2	1
2657	14	Aluminum / Vinyl	1	1954	1128	1	3	1
2658	14	Brick	1	1940	1109	1	4	1
2659	14	Aluminum / Vinyl	1	1951	1092	1	4	1
2660	14	Aluminum / Vinyl	2	1954	1403	1	3	1
2661	14	Aluminum / Vinyl	1	1923	1556	2	3	2
2662	14	Brick	1.5	1954	2416	2	6	2
2663	14	Aluminum / Vinyl	1	1922	1452	1	3	2
2664	14	Frame	1	1918	1394	1	3	1
2665	14	Masonry / Frame	1	1959	1539	1	3	1
2666	15	Frame	2	1895	1642	2	4	2
2667	15	Frame	1	1920	1725	1	4	2
2668	15	Aluminum / Vinyl	1	2005	1344	1	3	1
2669	1	Aluminum / Vinyl	1	1951	1274	1	4	1
2670	1	Aluminum / Vinyl	1	1948	1248	1	3	1
2671	1	Aluminum / Vinyl	1	1951	802	1	2	1
2672	1	Frame	1	1926	1347	1	3	1
2673	1	Aluminum / Vinyl	1	1960	1130	1	3	1
2674	1	Aluminum / Vinyl	1	1960	1104	1	3	1
2675	1	Masonry / Frame	1	1956	1045	1	3	2
2676	1	Aluminum / Vinyl	1	1954	1024	1	1	1
2677	1	Brick	1	1929	1422	1	4	2
2678	2	Brick	1	1950	1630	1	3	2
2679	2	Aluminum / Vinyl	1	1954	1299	1	3	1
2680	2	Brick	2	1959	2469	2	6	2

2681	2	Frame	1	1962	1538	1	3	2
2682	2	Aluminum / Vinyl	1	1965	1312	1	3	1
2683	2	Aluminum / Vinyl	1	1967	1205	1	3	1
2684	2	Stone	1	1961	1196	1	3	1
2685	2	Aluminum / Vinyl	1	1955	1175	1	3	1
2686	2	Aluminum / Vinyl	1	1974	1122	1	3	1
2687	2	Masonry / Frame	1	1948	1032	1	3	1
2688	2	Brick	1	1955	1024	1	3	1
2689	2	Aluminum / Vinyl	1	1955	950	1	3	1
2690	2	Aluminum / Vinyl	1	1959	936	1	3	1
2691	2	Brick	1	1956	936	1	3	1
2692	2	Aluminum / Vinyl	1	1951	912	1	3	1
2694	2	Aluminum / Vinyl	1	1966	2092	1	3	1
2695	2	Aluminum / Vinyl	1	1961	1510	1	4	1
2696	3	Aluminum / Vinyl	2	1922	2755	2	6	2
2697	3	Aluminum / Vinyl	2	1896	3128	2	8	2
2698	3	Aluminum / Vinyl	1	1891	2111	2	4	2
2699	3	Frame	2	1906	2105	1	3	3
2700	3	Frame	2	1901	1915	1	4	1
2701	3	Masonry / Frame	2	1926	1757	1	3	1
2702	3	Frame	2	1921	1694	1	3	1
2703	3	Frame	1	1900	1036	1	1	1
2704	4	Block	2	1893	1810	1	4	1
2705	5	Brick	1	1950	1729	1	4	2
2706	5	Brick	1	1951	1507	1	3	2
2707	5	Brick	1	1949	1477	1	4	1
2708	5	Frame	1	1950	1315	1	3	1
2709	5	Aluminum / Vinyl	1.5	1940	1312	1	3	1
2710	5	Aluminum / Vinyl	1	1949	1288	1	5	2
2711	5	Brick	1	1949	1263	1	3	1
2712	5	Aluminum / Vinyl	1	1956	1257	1	4	2
2713	5	Aluminum / Vinyl	1	1957	1225	1	3	1
2714	5	Stone	1	1949	1224	1	3	1
2715	5	Brick	1	1950	1156	1	2	1
2716	5	Aluminum / Vinyl	1	1952	1142	1	4	1
2717	5	Aluminum / Vinyl	1.5	1930	1919	2	3	2
2718	5	Stone	1	1953	1863	1	3	2
2719	5	Stone	1	1954	1760	1	2	1
2720	5	Brick	1	1967	1632	1	3	1
2721	5	Aluminum / Vinyl	1	1985	1465	1	3	2
2722	5	Frame	1	1959	1220	1	3	1
2723	5	Brick	1	1959	1219	1	3	1
2724	5	Aluminum / Vinyl	1	1971	1212	1	4	1

2725	5	Frame	1	1956	1130	1	3	1
2726	5	Aluminum / Vinyl	1	1962	1102	1	3	1
2727	5	Aluminum / Vinyl	1	1957	963	1	3	1
2728	5	Aluminum / Vinyl	1	1957	963	1	3	1
2729	5	Aluminum / Vinyl	1	1954	962	1	3	1
2730	5	Brick	1	1947	952	1	2	2
2731	5	Brick	1	1947	952	1	2	2
2732	5	Aluminum / Vinyl	1	1950	874	1	2	1
2733	5	Aluminum / Vinyl	1	1951	772	1	2	1
2734	5	Aluminum / Vinyl	1	1952	698	1	2	1
2735	5	Aluminum / Vinyl	1	1924	1086	1	3	1
2736	6	Brick	1.5	1927	2027	2	4	2
2737	6	Frame	1.5	1900	1635	2	4	2
2738	6	Aluminum / Vinyl	1	1925	1267	1	3	2
2739	6	Aluminum / Vinyl	1	1926	1243	1	3	1
2740	6	Aluminum / Vinyl	2	1888	1996	1	4	2
2741	7	Aluminum / Vinyl	1	1942	1307	1	3	1
2742	7	Brick	1.5	1937	1231	1	2	1
2743	7	Brick	1	1950	1227	1	3	1
2744	7	Stone	1	1947	1192	1	3	1
2745	7	Masonry / Frame	2	1951	2173	1	3	1
2746	7	Aluminum / Vinyl	2	1928	2410	2	6	2
2747	7	Aluminum / Vinyl	1.5	1926	2372	2	4	2
2748	7	Aluminum / Vinyl	2	1920	2354	2	5	2
2749	7	Frame	1	1927	1642	2	3	2
2750	7	Block	1	1926	797	1	3	1
2751	7	Aluminum / Vinyl	2	1900	1208	1	3	1
2752	8	Block	1	1946	1047	1	2	1
2753	8	Aluminum / Vinyl	1	1897	1003	1	3	1
2754	8	Aluminum / Vinyl	1.5	1929	2013	2	5	2
2755	8	Aluminum / Vinyl	2	1912	1848	2	4	2
2756	8	Brick	1	1921	1849	1	5	1
2757	8	Brick	1.5	1925	2004	1	4	1
2758	8	Aluminum / Vinyl	1.5	1906	1431	1	3	1
2759	8	Aluminum / Vinyl	1	1890	1429	1	3	1
2760	8	Aluminum / Vinyl	1	1910	1398	1	4	1
2761	8	Aluminum / Vinyl	1	1905	1088	1	3	1
2762	8	Aluminum / Vinyl	1	1925	1052	1	2	1
2763	9	Brick	1	1949	1147	1	3	1
2764	9	Aluminum / Vinyl	1	1969	1318	1	4	1
2765	9	Aluminum / Vinyl	1	1957	1188	1	3	1
2766	9	Aluminum / Vinyl	1	1958	1071	1	3	1
2767	9	Aluminum / Vinyl	1	1957	1018	1	3	1

2768	9	Aluminum / Vinyl	1	1981	912	1	3	2
2769	10	Stone	1	1948	1516	1	3	2
2770	10	Brick	1	1953	1368	1	3	1
2771	10	Brick	1	1952	1222	1	3	1
2772	10	Aluminum / Vinyl	1	1947	1211	1	3	1
2773	10	Aluminum / Vinyl	1	1947	1046	1	3	1
2774	10	Aluminum / Vinyl	1	1955	964	1	3	1
2775	10	Frame	2	1913	2091	1	4	1
2776	10	Frame	2	1923	1655	1	4	1
2777	10	Masonry / Frame	2	1948	1597	1	3	1
2778	10	Aluminum / Vinyl	2	1922	1390	1	3	2
2779	10	Aluminum / Vinyl	2	1926	1291	1	3	1
2780	10	Aluminum / Vinyl	2	1951	1260	1	3	1
2781	10	Masonry / Frame	2	1951	1151	1	3	1
2782	10	Aluminum / Vinyl	2	1926	2618	2	4	2
2783	10	Aluminum / Vinyl	2	1922	2460	2	6	2
2784	10	Aluminum / Vinyl	2	1959	1962	2	6	2
2785	10	Brick	2	1944	1722	2	4	2
2786	10	Aluminum / Vinyl	1	1924	1514	1	3	1
2787	10	Aluminum / Vinyl	1	1926	1270	1	3	1
2788	10	Aluminum / Vinyl	1	1925	1268	1	4	2
2789	10	Aluminum / Vinyl	1	1929	1219	1	3	1
2790	10	Aluminum / Vinyl	1	1926	948	1	2	1
2791	10	Brick	1	1953	1189	1	3	1
2792	10	Aluminum / Vinyl	1	1955	864	1	3	1
2793	10	Aluminum / Vinyl	1	1947	728	1	2	1
2794	10	Aluminum / Vinyl	1	1926	1142	1	3	1
2795	10	Stucco	1	1925	932	1	2	1
2796	11	Brick	1.5	1937	1751	1	3	1
2797	11	Brick	1	1953	1363	1	4	1
2798	11	Aluminum / Vinyl	1	1952	1012	1	3	1
2799	11	Aluminum / Vinyl	1	1942	984	1	3	1
2800	11	Fiber-Cement	2	2006	3027	1	4	2
2801	11	Masonry / Frame	2	1963	2328	1	4	2
2802	11	Aluminum / Vinyl	2	2009	1320	1	3	0
2803	11	Aluminum / Vinyl	1	1922	1364	1	3	1
2804	11	Brick	1	1955	1359	1	2	1
2805	11	Stone	1	1952	1193	1	2	1
2806	11	Brick	1	1954	1168	1	3	1
2807	11	Brick	1	1956	1148	1	3	2
2808	11	Brick	1	1959	1131	1	3	1
2809	11	Brick	1	1961	1127	1	2	1
2810	11	Aluminum / Vinyl	1	1957	1106	1	2	1

2811	11	Aluminum / Vinyl	1	1953	1104	1	3	1
2812	11	Aluminum / Vinyl	1	1964	1074	1	3	1
2813	11	Aluminum / Vinyl	1	1966	1033	1	3	1
2814	11	Aluminum / Vinyl	1	1963	1026	1	3	1
2815	11	Aluminum / Vinyl	1	1956	1019	1	3	1
2816	11	Aluminum / Vinyl	1	1954	1000	1	3	1
2817	11	Aluminum / Vinyl	1	1957	958	1	3	1
2818	11	Aluminum / Vinyl	1	1953	943	1	3	1
2819	11	Aluminum / Vinyl	1	1952	914	1	3	1
2820	11	Aluminum / Vinyl	1	1954	904	1	3	1
2821	11	Brick	1	1950	895	1	2	1
2822	11	Aluminum / Vinyl	1	1952	880	1	3	1
2823	11	Aluminum / Vinyl	1	1953	879	1	3	1
2824	11	Aluminum / Vinyl	1	1953	879	1	3	1
2825	11	Aluminum / Vinyl	1	1953	879	1	3	1
2826	11	Aluminum / Vinyl	1	1953	876	1	3	1
2827	11	Aluminum / Vinyl	1	1956	876	1	3	1
2828	11	Aluminum / Vinyl	1	1953	864	1	3	1
2829	11	Aluminum / Vinyl	1	1952	811	1	2	1
2830	11	Aluminum / Vinyl	1	1949	706	1	2	1
2831	11	Aluminum / Vinyl	1	1953	672	1	2	1
2832	11	Frame	1	1899	1426	1	3	1
2833	11	Aluminum / Vinyl	1	1925	1301	1	3	2
2834	11	Aluminum / Vinyl	1	1927	1028	1	2	1
2835	11	Brick	2	1956	2806	2	6	2
2836	11	Stone	1.5	1939	1235	1	3	1
2837	12	Frame	2	1914	1768	2	4	2
2838	12	Frame	1	1901	1248	1	3	1
2839	12	Aluminum / Vinyl	1	1912	1180	1	3	1
2840	12	Aluminum / Vinyl	1	1896	1037	1	3	0
2841	12	Aluminum / Vinyl	1	1885	996	1	3	1
2842	12	Aluminum / Vinyl	2	1912	1628	3	4	3
2843	13	Brick	1	1947	1238	1	3	2
2844	13	Brick	1	1951	1200	1	3	1
2845	13	Brick	1	1950	1134	1	3	1
2846	13	Aluminum / Vinyl	1	1940	1078	1	2	2
2847	13	Aluminum / Vinyl	1	1952	1053	1	3	1
2848	13	Aluminum / Vinyl	1	1948	981	1	3	1
2849	13	Aluminum / Vinyl	2	1939	1198	1	2	1
2850	13	Aluminum / Vinyl	2	1940	1193	1	2	1
2851	13	Aluminum / Vinyl	1	1928	1588	2	3	2
2852	13	Brick	2	1960	2076	2	6	2
2853	13	Aluminum / Vinyl	1	1927	1229	1	3	2

2854	13	Aluminum / Vinyl	1	1974	1473	1	3	1
2855	13	Brick	1	1964	1140	1	3	1
2856	13	Brick	1	1967	1129	1	3	1
2857	13	Brick	1	1948	993	1	2	1
2858	13	Aluminum / Vinyl	1	1950	796	1	2	1
2859	13	Aluminum / Vinyl	1	1947	672	1	2	1
2860	14	Aluminum / Vinyl	1	1940	1173	1	3	1
2861	14	Aluminum / Vinyl	1	1953	652	1	2	1
2862	14	Masonry / Frame	2	1925	1580	1	3	1
2863	14	Aluminum / Vinyl	2	1939	1372	1	3	1
2864	14	Aluminum / Vinyl	1	1901	1122	1	4	1
2865	14	Aluminum / Vinyl	2	1924	2148	2	4	2
2866	14	Aluminum / Vinyl	1.5	1927	1890	2	3	2
2867	14	Aluminum / Vinyl	1.5	1916	1648	2	3	2
2868	14	Frame	1.5	1925	1632	2	4	2
2869	14	Aluminum / Vinyl	1	1900	1623	2	5	2
2870	14	Aluminum / Vinyl	1.5	1913	1764	1	5	1
2871	14	Frame	1.5	1929	1173	1	3	1
2872	14	Aluminum / Vinyl	1	1923	1150	1	4	2
2873	14	Aluminum / Vinyl	1	1929	1021	1	2	1
2874	14	Aluminum / Vinyl	1	1944	768	1	2	1
2875	14	Aluminum / Vinyl	1.5	1886	1728	1	4	1
2876	14	Aluminum / Vinyl	1	1900	1431	1	4	2
2877	14	Frame	1	1922	1178	1	3	1
2878	14	Aluminum / Vinyl	1	1928	1065	1	3	1
2879	14	Aluminum / Vinyl	1	1884	1050	1	4	2
2880	14	Frame	1	1928	864	1	1	1
2881	15	Brick	2	1924	2224	1	4	1
2882	15	Aluminum / Vinyl	2	1907	2138	2	4	2
2883	15	Aluminum / Vinyl	2	1875	1656	2	4	2
2884	15	Aluminum / Vinyl	1	1924	748	1	1	1
2886	1	Stone	1	1940	1488	1	4	1
2887	1	Brick	1	1952	1256	1	4	1
2888	1	Aluminum / Vinyl	1	1951	1193	1	4	2
2889	1	Aluminum / Vinyl	1	1941	879	1	3	1
2890	1	Frame	1	1952	1247	1	3	1
2891	1	Aluminum / Vinyl	1	1926	1142	1	3	1
2892	1	Brick	2	1927	1822	1	3	1
2893	2	Stone	2	1934	1806	1	5	1
2894	2	Masonry / Frame	2	1957	2323	2	6	2
2895	2	Aluminum / Vinyl	2	1956	2200	2	6	2
2896	2	Brick	1.5	1956	2077	2	4	2
2897	2	Brick	1	1955	1177	1	3	1

2898	2	Aluminum / Vinyl	1	1967	1072	1	3	1
2899	2	Brick	1	1959	1016	1	3	1
2900	2	Aluminum / Vinyl	1	1955	950	1	3	2
2901	2	Aluminum / Vinyl	1	1955	950	1	3	1
2902	2	Frame	1	1957	925	1	3	1
2903	3	Frame	1	1890	1184	1	3	1
2904	3	Aluminum / Vinyl	1	1880	917	1	2	1
2905	3	Aluminum / Vinyl	2	1912	3350	2	7	3
2906	3	Aluminum / Vinyl	2	1912	1548	1	3	1
2907	4	Stucco	2	1908	2584	1	5	1
2908	5	Brick	1.5	1941	2003	1	4	2
2909	5	Stone	1.5	1949	1758	1	4	2
2910	5	Stone	1	1946	1397	1	3	1
2911	5	Stone	1	1947	1389	1	4	1
2912	5	Brick	1	1956	1258	1	4	2
2913	5	Aluminum / Vinyl	1.5	1954	1192	1	4	2
2914	5	Brick	1	1950	1158	1	3	1
2915	5	Aluminum / Vinyl	1	1953	1102	1	3	2
2916	5	Aluminum / Vinyl	1	1950	980	1	3	1
2917	5	Aluminum / Vinyl	1	1948	858	1	2	1
2918	5	Aluminum / Vinyl	2	1953	1422	1	3	1
2919	5	Aluminum / Vinyl	2	1950	3191	2	6	2
2920	5	Brick	2	1959	2478	2	6	2
2921	5	Brick	2	1959	2478	2	6	2
2922	5	Brick	2	1952	2455	2	4	2
2923	5	Prem Wood	1	2001	2073	1	3	2
2924	5	Brick	1	1955	1475	1	3	2
2925	5	Brick	1	1954	1465	1	2	1
2926	5	Brick	1	1958	1415	1	2	1
2927	5	Brick	1	1956	1382	1	3	1
2928	5	Aluminum / Vinyl	1	1969	1331	1	3	1
2929	5	Brick	1	1960	1235	1	2	2
2930	5	Aluminum / Vinyl	1	1971	1204	1	4	1
2931	5	Brick	1	1956	1150	1	3	1
2932	5	Aluminum / Vinyl	1	1949	1136	1	3	1
2933	5	Aluminum / Vinyl	1	1956	1132	1	3	1
2934	5	Brick	1	1960	1127	1	3	1
2935	5	Aluminum / Vinyl	1	1957	1107	1	3	1
2936	5	Aluminum / Vinyl	1	1955	1099	1	3	1
2937	5	Brick	1	1958	1084	1	3	1
2938	5	Aluminum / Vinyl	1	1948	1027	1	2	1
2939	5	Aluminum / Vinyl	1	1955	1008	1	3	1
2940	5	Aluminum / Vinyl	1	1959	994	1	3	1

2941	5	Aluminum / Vinyl	1	1955	993	1	3	1
2942	5	Brick	1	1956	978	1	3	1
2943	5	Brick	1	1956	936	1	3	1
2944	5	Aluminum / Vinyl	1	1950	871	1	3	1
2945	5	Aluminum / Vinyl	1	1951	789	1	2	1
2946	5	Aluminum / Vinyl	1	1959	1557	1	3	1
2947	6	Fiber-Cement	2	2004	2146	1	3	2
2948	6	Frame	1	1880	1120	1	1	2
2949	6	Frame	2	1890	2428	1	4	2
2950	6	Aluminum / Vinyl	1.5	1910	1645	1	5	1
2951	6	Frame	1	1890	1488	1	3	1
2952	6	Aluminum / Vinyl	1.5	1913	1391	1	3	1
2953	6	Frame	1.5	1880	1350	1	2	1
2954	6	Aluminum / Vinyl	1	1923	1276	1	4	1
2955	6	Aluminum / Vinyl	1	1901	1228	1	4	1
2956	7	Stone	1.5	1934	1755	1	3	1
2957	7	Stone	1	1947	1441	1	3	2
2958	7	Frame	1	1936	1001	1	2	1
2959	7	Aluminum / Vinyl	1	1922	589	1	1	1
2960	7	Brick	2	1958	2284	2	6	2
2961	7	Aluminum / Vinyl	1	1922	1743	1	4	2
2962	7	Aluminum / Vinyl	1	1900	990	1	3	1
2963	8	Aluminum / Vinyl	2	1900	2278	2	6	2
2964	8	Frame	1.5	1918	1636	2	3	2
2965	8	Aluminum / Vinyl	1	1923	1088	1	4	1
2966	8	Brick	1	1955	1188	1	3	1
2967	8	Frame	2	1916	2076	1	3	1
2968	8	Aluminum / Vinyl	1	1892	1523	1	4	2
2969	8	Aluminum / Vinyl	1	1892	1523	1	4	2
2970	8	Aluminum / Vinyl	1.5	1904	1386	1	3	1
2971	8	Aluminum / Vinyl	1.5	1904	1386	1	3	1
2972	9	Fiber-Cement	1	2010	2153	1	3	2
2973	9	Aluminum / Vinyl	1	1968	2127	1	4	2
2974	9	Aluminum / Vinyl	1	1974	1584	1	3	1
2975	9	Aluminum / Vinyl	1	1969	1142	1	2	1
2976	9	Frame	1	1959	1092	1	3	2
2977	9	Aluminum / Vinyl	1	1956	946	1	3	1
2978	9	Aluminum / Vinyl	2	1980	2464	2	6	2
2979	10	Stone	1	1940	1524	1	3	1
2980	10	Aluminum / Vinyl	1	1952	1221	1	4	1
2981	10	Brick	1	1950	1220	1	3	1
2982	10	Aluminum / Vinyl	1	1946	1093	1	3	2
2983	10	Brick	2	1940	1709	1	3	2

2984	10	Block	2	1948	1639	1	3	2
2985	10	Aluminum / Vinyl	2	1951	1514	1	2	1
2986	10	Aluminum / Vinyl	1	1928	1014	1	2	1
2987	10	Aluminum / Vinyl	1	1914	787	1	2	1
2988	10	Brick	2	1922	3609	2	6	2
2989	10	Aluminum / Vinyl	1.5	1925	1891	1	3	2
2990	10	Aluminum / Vinyl	1	1924	1209	1	2	1
2991	10	Aluminum / Vinyl	1	1924	1132	1	3	1
2992	10	Frame	1	1926	1073	1	3	1
2993	10	Aluminum / Vinyl	1	1922	1013	1	2	1
2994	10	Aluminum / Vinyl	1	1918	972	1	4	1
2995	10	Aluminum / Vinyl	1	1955	969	1	3	1
2996	10	Aluminum / Vinyl	1	1954	965	1	3	1
2997	10	Brick	1	1953	784	1	2	1
2998	10	Frame	1	1950	728	1	2	1
2999	10	Masonry / Frame	2	1917	2713	1	5	2
3000	10	Aluminum / Vinyl	2	1893	1741	1	3	1
3001	10	Aluminum / Vinyl	1.5	1900	1685	1	5	1
3002	10	Aluminum / Vinyl	1.5	1922	1392	1	3	2
3003	10	Frame	1	1900	1327	1	3	2
3004	10	Aluminum / Vinyl	1	1919	1063	1	3	1
3005	10	Frame	1	1927	970	1	3	1
3006	10	Stone	2	1956	3630	3	7	4
3007	10	Brick	2	1930	2681	1	3	1
3008	10	Brick	1.5	1931	1560	1	3	1
3009	11	Brick	1.5	1951	2384	1	5	2
3010	11	Aluminum / Vinyl	1	1956	1942	1	3	1
3011	11	Aluminum / Vinyl	1.5	1942	1254	1	2	1
3012	11	Aluminum / Vinyl	1	1952	1129	1	3	1
3013	11	Aluminum / Vinyl	1	1953	1073	1	3	1
3014	11	Aluminum / Vinyl	1	1953	996	1	3	1
3015	11	Aluminum / Vinyl	2	1984	1902	1	3	2
3016	11	Aluminum / Vinyl	2	1957	1378	1	3	1
3017	11	Brick	2	1955	2924	2	5	2
3018	11	Brick	2	1957	2268	2	6	2
3019	11	Aluminum / Vinyl	2	1956	1948	2	6	2
3020	11	Aluminum / Vinyl	1	1984	1597	1	3	3
3021	11	Brick	1	1967	1173	1	3	1
3022	11	Brick	1	1955	1153	1	3	2
3023	11	Brick	1	1959	1130	1	3	1
3024	11	Stone	1	1954	1120	1	3	1
3025	11	Brick	1	1954	1058	1	3	1
3026	11	Aluminum / Vinyl	1	1956	1019	1	3	1

3027	11	Brick	1	1953	1013	1	3	1
3028	11	Aluminum / Vinyl	1	1961	985	1	3	1
3029	11	Aluminum / Vinyl	1	1956	958	1	2	1
3030	11	Aluminum / Vinyl	1	1955	955	1	3	1
3031	11	Brick	1	1956	934	1	3	1
3032	11	Brick	1.5	1918	1479	1	4	1
3033	12	Frame	1	1885	1528	1	5	1
3034	12	Frame	1	1900	1232	1	3	2
3035	12	Aluminum / Vinyl	1	1895	948	1	3	1
3036	12	Aluminum / Vinyl	1	1905	2530	2	7	2
3037	12	Aluminum / Vinyl	2	1900	1814	2	4	1
3039	13	Aluminum / Vinyl	1	1950	1132	1	4	1
3040	13	Stone	1.5	1940	1098	1	2	1
3041	13	Stone	1	1946	968	1	3	1
3042	13	Aluminum / Vinyl	1	1953	840	1	2	1
3043	13	Aluminum / Vinyl	1	1940	816	1	3	1
3044	13	Aluminum / Vinyl	2	1946	1378	1	3	1
3045	13	Aluminum / Vinyl	1	1926	1348	1	4	1
3046	13	Frame	1	1964	1726	1	3	1
3047	13	Aluminum / Vinyl	1	1938	1395	1	3	2
3048	13	Aluminum / Vinyl	1	1965	1144	1	3	1
3049	13	Frame	1	1961	1098	1	3	1
3050	13	Aluminum / Vinyl	1	1966	1036	1	3	1
3051	13	Aluminum / Vinyl	1	1961	981	1	3	2
3052	13	Aluminum / Vinyl	1	1959	927	1	3	1
3053	13	Aluminum / Vinyl	1	1959	921	1	3	2
3054	13	Aluminum / Vinyl	1	1958	905	1	3	1
3055	13	Aluminum / Vinyl	1	1954	872	1	3	1
3056	13	Stone	1	1953	825	1	2	1
3057	13	Aluminum / Vinyl	1	1946	679	1	2	1
3058	13	Aluminum / Vinyl	1	1950	651	1	2	1
3059	14	Aluminum / Vinyl	1.5	1949	1765	1	3	1
3060	14	Aluminum / Vinyl	1	1956	1203	1	2	1
3061	14	Stucco	1	1944	1188	1	2	2
3062	14	Brick	1	1950	1095	1	3	1
3063	14	Brick	1	1948	828	1	2	1
3064	14	Aluminum / Vinyl	2	1948	1378	1	3	1
3065	14	Aluminum / Vinyl	1	1901	1130	1	3	1
3066	14	Aluminum / Vinyl	1	1895	640	1	3	1
3067	14	Brick	1.5	1926	2358	2	5	2
3068	14	Masonry / Frame	2	1928	1942	2	3	3
3069	14	Brick	2	1939	1906	2	4	2
3070	14	Frame	2	1910	2172	2	5	2

3071	14	Aluminum / Vinyl	1.5	1918	1786	2	5	2
3072	14	Aluminum / Vinyl	1.5	1912	1294	2	2	2
3073	14	Aluminum / Vinyl	1	1890	1516	2	3	2
3074	14	Aluminum / Vinyl	1	1905	1360	2	4	1
3075	14	Aluminum / Vinyl	1	1926	1894	1	4	2
3076	14	Aluminum / Vinyl	1.5	1925	1691	1	4	2
3077	14	Frame	1	1926	1622	1	3	2
3078	14	Frame	1	1923	1414	1	4	2
3079	14	Frame	1	1926	1410	1	4	1
3080	14	Frame	1	1927	1331	1	3	1
3081	14	Aluminum / Vinyl	1	1926	1325	1	2	1
3082	14	Stucco	1	1944	1399	1	3	2
3083	14	Aluminum / Vinyl	1	1953	1100	1	3	1
3084	14	Aluminum / Vinyl	1	1953	1100	1	3	1
3085	14	Aluminum / Vinyl	1	1911	1823	1	3	2
3086	14	Aluminum / Vinyl	1.5	1900	1556	1	4	2
3087	14	Aluminum / Vinyl	1	1908	1549	1	4	1
3088	14	Frame	1.5	1885	1336	1	3	2
3089	14	Aluminum / Vinyl	1.5	1888	1179	1	3	1
3090	15	Aluminum / Vinyl	2	2004	2197	1	3	2
3091	15	Aluminum / Vinyl	>2	2006	1884	1	2	2
3092	15	Frame	1	1922	1419	1	3	1
3093	15	Aluminum / Vinyl	1	1970	1023	1	3	1
3094	1	Brick	2	1953	1388	1	4	1
3095	1	Stone	1	1941	1276	1	3	1
3096	1	Brick	1	1946	1173	1	3	1
3097	1	Brick	1	1953	1058	1	3	1
3098	1	Brick	1	1953	1058	1	3	1
3099	1	Brick	1	1949	744	1	2	1
3100	1	Aluminum / Vinyl	1	1905	820	1	2	1
3101	1	Aluminum / Vinyl	1.5	1928	2012	2	4	2
3102	1	Masonry / Frame	2	1951	1692	2	4	2
3103	1	Aluminum / Vinyl	1.5	1920	1188	2	3	2
3104	1	Frame	1	1928	1624	1	4	1
3105	1	Aluminum / Vinyl	1	2007	1439	1	3	2
3106	1	Frame	1	1952	1144	1	3	1
3107	1	Brick	1	1952	998	1	2	1
3108	1	Brick	1	1949	816	1	1	1
3109	2	Brick	1	1959	1416	1	4	2
3110	2	Brick	1	1951	1350	1	2	1
3111	2	Aluminum / Vinyl	1	1955	1211	1	3	1
3112	2	Brick	1.5	1940	1104	1	2	1
3113	2	Aluminum / Vinyl	1	1955	1068	1	4	1

3114	2	Brick	2	1959	2662	2	6	2
3115	2	Brick	1	1959	1206	1	3	1
3116	2	Brick	1	1955	1177	1	3	1
3117	2	Frame	1	1972	1144	1	4	1
3118	2	Aluminum / Vinyl	1	1955	1120	1	3	1
3119	2	Aluminum / Vinyl	1	1956	1041	1	3	1
3120	2	Aluminum / Vinyl	1	1976	1022	1	3	1
3121	2	Brick	1	1955	1020	1	3	1
3122	2	Aluminum / Vinyl	1	1955	950	1	3	1
3123	2	Aluminum / Vinyl	1	1956	936	1	3	1
3124	2	Aluminum / Vinyl	1	1950	930	1	2	1
3125	2	Aluminum / Vinyl	1	1957	925	1	3	1
3126	2	Aluminum / Vinyl	1	1956	905	1	3	1
3127	3	Frame	2	1923	2210	1	4	2
3128	3	Masonry / Frame	2	1925	1994	1	3	1
3129	3	Aluminum / Vinyl	2	1890	1813	1	4	2
3130	3	Aluminum / Vinyl	2	1890	2973	2	6	2
3131	3	Frame	1.5	1890	2507	2	4	2
3132	3	Aluminum / Vinyl	2	1900	1642	2	4	2
3133	3	Brick	1	1929	1999	1	4	2
3134	3	Frame	1.5	1920	1980	1	4	1
3135	3	Aluminum / Vinyl	1.5	1922	1970	1	3	1
3136	3	Frame	1	1914	1692	1	4	1
3137	3	Frame	1.5	1889	1494	1	3	1
3138	3	Frame	1	1900	1049	1	3	1
3139	4	Stucco	1	1909	1040	1	4	2
3140	5	Aluminum / Vinyl	1	1976	1784	1	4	3
3141	5	Brick	1.5	1948	1748	1	4	2
3142	5	Stone	1.5	1947	1646	1	4	2
3143	5	Brick	1	1950	1600	1	4	2
3144	5	Aluminum / Vinyl	1.5	1939	1553	1	3	2
3145	5	Aluminum / Vinyl	1	1954	1510	1	3	1
3146	5	Brick	1	1947	1322	1	3	1
3147	5	Aluminum / Vinyl	1	1952	1215	1	3	1
3148	5	Aluminum / Vinyl	1	1952	1168	1	4	2
3149	5	Aluminum / Vinyl	1	1952	1080	1	4	2
3150	5	Brick	1	1940	1071	1	3	1
3151	5	Fiber-Cement	2	2008	3535	1	5	3
3152	5	Aluminum / Vinyl	2	2009	2195	1	4	2
3153	5	Brick	2	1946	1498	1	3	1
3154	5	Masonry / Frame	2	1955	2079	2	4	2
3155	5	Brick	1	1954	2129	1	2	1
3156	5	Brick	1	1977	1358	1	3	2

3157	5	Brick	1	1960	1254	1	3	1
3158	5	Brick	1	1955	1237	1	3	2
3159	5	Aluminum / Vinyl	1	1952	1188	1	4	1
3160	5	Brick	1	1954	1177	1	3	1
3161	5	Brick	1	1957	1115	1	3	1
3162	5	Masonry / Frame	1	1956	1060	1	3	1
3163	5	Aluminum / Vinyl	1	1953	1052	1	3	1
3164	5	Aluminum / Vinyl	1	1956	1019	1	3	1
3165	5	Aluminum / Vinyl	1	1956	1019	1	3	1
3166	5	Aluminum / Vinyl	1	1955	1008	1	3	1
3167	5	Aluminum / Vinyl	1	1962	999	1	3	1
3168	5	Aluminum / Vinyl	1	1953	888	1	3	2
3169	5	Aluminum / Vinyl	1	1953	864	1	3	1
3170	5	Aluminum / Vinyl	1	1924	1235	1	3	1
3171	5	Aluminum / Vinyl	1	1981	1970	1	3	2
3172	6	Frame	1	1924	1368	1	4	1
3174	6	Aluminum / Vinyl	2	1992	2788	2	6	2
3175	7	Stone	1	1936	1716	1	4	2
3176	7	Brick	1.5	1938	1492	1	2	1
3177	7	Brick	1.5	1939	1488	1	4	1
3178	7	Brick	1.5	1938	1408	1	2	1
3179	7	Frame	1	1962	1162	1	4	1
3180	7	Aluminum / Vinyl	1	1952	1075	1	3	1
3181	7	Aluminum / Vinyl	2	1924	2882	2	7	2
3182	7	Aluminum / Vinyl	1.5	1927	1725	2	3	2
3183	7	Brick	1	1928	1721	1	4	1
3184	7	Aluminum / Vinyl	1	1915	840	1	3	1
3185	7	Brick	1	1950	1270	1	4	2
3186	7	Stone	1	1947	1206	1	2	1
3187	7	Brick	2	1931	1772	1	3	1
3188	8	Prem Wood	1.5	1913	1245	1	3	2
3189	8	Aluminum / Vinyl	1	1924	1905	1	5	1
3190	8	Aluminum / Vinyl	1	1928	1614	1	4	2
3191	8	Brick	1	1928	1551	1	4	1
3192	8	Frame	1	1957	1026	1	3	1
3193	8	Aluminum / Vinyl	1	1913	1239	1	3	2
3194	8	Aluminum / Vinyl	1	1912	934	1	3	1
3195	9	Masonry / Frame	1.5	1925	1377	1	3	1
3196	9	Aluminum / Vinyl	1	2005	1932	1	3	2
3197	9	Aluminum / Vinyl	1	2010	1709	1	3	2
3198	9	Aluminum / Vinyl	1	1958	1515	1	3	1
3199	9	Aluminum / Vinyl	1	1972	1219	1	3	1
3200	9	Aluminum / Vinyl	1	1956	996	1	3	1

3201	9	Aluminum / Vinyl	1	1957	991	1	4	2
3202	10	Stone	1	1936	1807	1	4	2
3203	10	Brick	1	1948	1108	1	3	1
3204	10	Aluminum / Vinyl	1	1957	1100	1	3	2
3205	10	Brick	2	1937	1727	1	3	1
3206	10	Masonry / Frame	2	1955	1675	1	3	2
3207	10	Aluminum / Vinyl	2	1926	1271	1	3	1
3208	10	Aluminum / Vinyl	1	1925	760	1	3	1
3209	10	Aluminum / Vinyl	2	1926	2139	2	5	2
3210	10	Aluminum / Vinyl	1.5	1925	2337	1	5	2
3211	10	Aluminum / Vinyl	1	1918	1873	1	3	2
3212	10	Aluminum / Vinyl	1	1924	1685	1	4	2
3213	10	Aluminum / Vinyl	1	1925	1273	1	3	1
3214	10	Brick	1	1941	1516	1	2	1
3215	10	Brick	1	1951	912	1	2	2
3216	10	Brick	1	1926	2036	1	3	2
3217	10	Aluminum / Vinyl	1	1911	2028	1	3	2
3218	10	Aluminum / Vinyl	1.5	1927	1451	1	4	2
3219	10	Brick	1.5	1923	1304	1	3	1
3220	11	Stone	1.5	1938	1434	1	3	1
3221	11	Aluminum / Vinyl	1	1954	1037	1	3	1
3222	11	Brick	1	1949	800	1	2	1
3223	11	Masonry / Frame	2	1941	1720	1	3	1
3224	11	Stone	2	1937	1652	1	3	1
3225	11	Aluminum / Vinyl	1.5	1955	1916	2	5	2
3226	11	Aluminum / Vinyl	2	1954	1728	2	6	2
3227	11	Brick	1	1931	1177	1	2	1
3228	11	Stone	1	1950	1392	1	2	2
3229	11	Aluminum / Vinyl	1	1968	1275	1	4	1
3230	11	Brick	1	1961	1192	1	3	1
3231	11	Stone	1	1949	1127	1	3	1
3232	11	Brick	1	1956	1112	1	3	1
3233	11	Brick	1	1958	1112	1	3	1
3234	11	Frame	1	1964	1094	1	3	1
3235	11	Aluminum / Vinyl	1	1961	1058	1	3	2
3236	11	Brick	1	1954	1013	1	3	2
3237	11	Aluminum / Vinyl	1	1953	971	1	2	2
3238	11	Aluminum / Vinyl	1	1951	884	1	3	1
3239	11	Aluminum / Vinyl	1	1955	876	1	3	1
3240	11	Brick	1	1950	839	1	2	1
3241	11	Brick	1	1924	1154	1	3	1
3242	11	Stone	1.5	1939	1565	1	4	2
3243	12	Frame	1.5	1909	1265	1	3	1

3244	12	Aluminum / Vinyl	1	1860	740	1	1	1
3245	12	Frame	1.5	1914	2408	2	5	3
3246	12	Aluminum / Vinyl	1	1905	1516	1	4	1
3247	13	Brick	1	1949	1312	1	3	1
3248	13	Frame	1	1951	1284	1	4	2
3249	13	Aluminum / Vinyl	1	1941	1213	1	3	1
3250	13	Aluminum / Vinyl	1	1948	1114	1	3	2
3251	13	Aluminum / Vinyl	1	1947	1046	1	2	1
3252	13	Aluminum / Vinyl	1	1956	780	1	2	1
3253	13	Aluminum / Vinyl	2	1967	1638	1	4	1
3254	13	Stucco	1.5	1920	1741	2	5	2
3255	13	Aluminum / Vinyl	1.5	1926	1474	1	4	2
3256	13	Brick	1	1967	1475	1	3	1
3257	13	Brick	1	1969	1409	1	3	1
3258	13	Aluminum / Vinyl	1	1970	1177	1	3	1
3259	13	Aluminum / Vinyl	1	1960	1030	1	3	1
3260	13	Aluminum / Vinyl	1	1956	943	1	3	1
3261	14	Brick	1	1953	1285	1	3	1
3262	14	Aluminum / Vinyl	1	1947	1226	1	3	1
3263	14	Aluminum / Vinyl	1	1938	996	1	3	1
3264	14	Brick	2	1951	2196	1	3	2
3265	14	Aluminum / Vinyl	2	1924	1356	1	3	1
3266	14	Aluminum / Vinyl	2	1949	968	1	2	1
3267	14	Aluminum / Vinyl	1.5	1921	1295	1	4	2
3268	14	Aluminum / Vinyl	2	1926	1785	2	4	2
3269	14	Aluminum / Vinyl	1.5	1926	1644	2	3	2
3270	14	Aluminum / Vinyl	2	1891	2482	2	6	2
3271	14	Brick	2	1876	2274	2	4	3
3272	14	Aluminum / Vinyl	2	1913	1634	2	4	2
3273	14	Frame	1.5	1926	1510	1	4	1
3274	14	Aluminum / Vinyl	1.5	1916	1327	1	3	2
3275	14	Aluminum / Vinyl	1	1921	1238	1	3	1
3276	14	Aluminum / Vinyl	1.5	1918	1160	1	3	1
3277	14	Aluminum / Vinyl	1	1926	1058	1	3	1
3278	14	Aluminum / Vinyl	1	1925	1029	1	1	1
3279	14	Aluminum / Vinyl	1	1941	840	1	2	1
3280	14	Aluminum / Vinyl	1	1948	826	1	2	1
3281	14	Aluminum / Vinyl	1	1942	826	1	2	1
3282	14	Aluminum / Vinyl	1	1951	795	1	2	1
3283	14	Aluminum / Vinyl	2	1891	2514	1	3	2
3284	14	Frame	2	1905	2017	1	3	2
3285	14	Frame	1.5	1889	1382	1	3	1
3286	14	Aluminum / Vinyl	1	1913	1062	1	2	3

3287	15	Frame	2	1890	2898	2	5	3
3288	15	Frame	2	1904	2662	2	6	2
3289	1	Stucco	1.5	1926	2251	2	4	2
3290	1	Aluminum / Vinyl	1.5	1914	1921	2	5	2
3291	1	Aluminum / Vinyl	1	1964	1213	1	3	2
3292	1	Aluminum / Vinyl	1	1954	963	1	3	2
3293	1	Frame	1	1949	768	1	2	1
3294	2	Brick	1	1954	1267	1	3	1
3295	2	Aluminum / Vinyl	1	1955	1240	1	4	1
3296	2	Stone	1	1963	1392	1	4	1
3297	2	Brick	1	1957	1274	1	3	2
3298	2	Brick	1	1957	1147	1	3	1
3299	2	Aluminum / Vinyl	1	1958	1040	1	3	1
3300	2	Aluminum / Vinyl	1	1961	960	1	3	1
3301	2	Aluminum / Vinyl	1	1956	936	1	2	1
3302	2	Aluminum / Vinyl	1	1952	925	1	2	1
3303	2	Aluminum / Vinyl	1	1955	838	1	2	1
3304	3	Aluminum / Vinyl	1.5	1894	1774	2	3	2
3305	3	Aluminum / Vinyl	1.5	1911	1559	2	4	3
3306	3	Aluminum / Vinyl	1	1890	2053	2	3	2
3307	3	Aluminum / Vinyl	1	1926	1126	1	3	1
3308	3	Masonry / Frame	2	1909	2156	1	4	1
3309	3	Aluminum / Vinyl	2	1893	1543	1	3	1
3310	4	Frame	2	1909	2014	1	4	1
3311	5	Aluminum / Vinyl	1	1950	1156	1	3	2
3312	5	Aluminum / Vinyl	1	1951	1139	1	3	1
3313	5	Aluminum / Vinyl	1	1952	1102	1	3	1
3314	5	Aluminum / Vinyl	2	1969	1709	1	4	2
3315	5	Aluminum / Vinyl	2	1954	1415	1	3	1
3316	5	Masonry / Frame	2	1951	2016	2	4	2
3317	5	Aluminum / Vinyl	1.5	1960	1841	2	5	2
3318	5	Aluminum / Vinyl	1	1966	1332	1	4	1
3319	5	Brick	1	1956	1101	1	2	1
3320	5	Aluminum / Vinyl	1	1952	768	1	2	1
3321	5	Aluminum / Vinyl	2	1970	2405	2	6	2
3322	6	Aluminum / Vinyl	1.5	1923	1452	2	3	2
3323	7	Aluminum / Vinyl	1	1953	1283	1	4	1
3324	7	Aluminum / Vinyl	1	1941	1188	1	4	1
3325	7	Masonry / Frame	2	1924	1888	1	4	1
3326	7	Stone	2	1944	1884	2	4	2
3327	7	Frame	2	1942	1722	2	4	2
3328	7	Aluminum / Vinyl	1	1932	1204	1	3	1
3329	8	Aluminum / Vinyl	2	1925	2088	2	4	2

3330	8	Aluminum / Vinyl	1.5	1915	1872	2	3	2
3331	8	Aluminum / Vinyl	2	1915	2484	2	6	2
3332	8	Aluminum / Vinyl	2	1915	2450	2	5	2
3333	8	Frame	2	1905	1652	2	5	2
3334	8	Frame	1	1919	1619	1	4	1
3335	8	Aluminum / Vinyl	1	1928	1250	1	4	1
3336	8	Brick	1.5	1924	2298	1	3	0
3337	8	Frame	1	1900	2085	1	5	2
3338	8	Aluminum / Vinyl	1.5	1889	1697	1	4	2
3339	8	Aluminum / Vinyl	2	1916	1532	1	3	1
3340	8	Frame	1	1900	1096	1	3	1
3341	8	Frame	1	1911	1022	1	3	1
3342	9	Brick	1	1957	1308	1	3	1
3343	9	Aluminum / Vinyl	1	1959	1117	1	3	1
3344	9	Aluminum / Vinyl	1	1959	1024	1	3	1
3345	9	Aluminum / Vinyl	1	1954	958	1	3	1
3346	10	Brick	2	1929	2122	1	3	2
3347	10	Masonry / Frame	2	1953	1657	1	3	1
3348	10	Masonry / Frame	2	1925	2600	2	4	2
3349	10	Aluminum / Vinyl	1	1921	1824	1	4	3
3350	10	Frame	1	1917	1636	1	4	1
3351	10	Aluminum / Vinyl	1	1927	1454	1	5	2
3352	10	Aluminum / Vinyl	1	1926	1398	1	3	1
3353	10	Aluminum / Vinyl	1	1927	1269	1	3	1
3354	10	Stone	1	1955	1519	1	2	1
3355	10	Aluminum / Vinyl	1	1956	1045	1	2	1
3356	10	Aluminum / Vinyl	1	1949	746	1	2	1
3357	10	Aluminum / Vinyl	1	1937	616	1	2	1
3358	10	Aluminum / Vinyl	1.5	1896	1492	1	3	2
3359	10	Aluminum / Vinyl	1	1925	1185	1	3	1
3360	10	Aluminum / Vinyl	1	1917	1113	1	3	1
3361	11	Frame	1.5	1955	1340	1	3	1
3362	11	Brick	1	1949	1210	1	3	1
3363	11	Aluminum / Vinyl	1.5	1997	2404	2	5	3
3364	11	Aluminum / Vinyl	2	1953	1904	2	6	2
3365	11	Brick	1	1956	1260	1	3	2
3366	11	Brick	1	1966	1170	1	3	1
3367	11	Brick	1	1956	1150	1	3	1
3368	11	Aluminum / Vinyl	1	1956	1073	1	3	2
3369	11	Aluminum / Vinyl	1	1954	986	1	3	1
3370	11	Aluminum / Vinyl	1	1954	972	1	3	1
3371	11	Brick	1	1956	947	1	3	1
3372	13	Brick	1	1980	1945	1	3	2

3373	13	Brick	1	1953	1224	1	4	1
3374	13	Brick	1	1959	1203	1	3	1
3375	13	Brick	1	1953	1200	1	3	1
3376	13	Aluminum / Vinyl	1	1939	1187	1	3	1
3377	13	Brick	1	1955	1091	1	4	1
3378	13	Aluminum / Vinyl	2	1998	2196	1	3	2
3379	13	Aluminum / Vinyl	1.5	1943	1240	2	4	2
3380	13	Aluminum / Vinyl	1	1926	1310	1	4	1
3381	13	Aluminum / Vinyl	1	1936	1101	1	3	2
3382	13	Aluminum / Vinyl	1	1958	1243	1	2	1
3383	13	Aluminum / Vinyl	1	1913	1524	1	4	1
3384	13	Brick	2	1954	2588	3	4	3
3385	13	Aluminum / Vinyl	1.5	1904	1722	3	3	3
3386	14	Brick	1	1952	1515	1	4	1
3387	14	Block	1	1944	995	1	3	1
3388	14	Brick	1	1951	977	1	3	1
3389	14	Aluminum / Vinyl	1	1950	776	1	2	1
3390	14	Aluminum / Vinyl	1	1890	733	1	2	1
3391	14	Masonry / Frame	2	1919	3721	2	4	3
3392	14	Aluminum / Vinyl	1	1922	1724	1	5	1
3393	14	Brick	1	1929	1618	1	4	1
3394	14	Frame	1	1916	1584	1	4	2
3395	14	Frame	1	1926	1417	1	3	1
3396	14	Aluminum / Vinyl	1	1927	1038	1	3	1
3397	14	Brick	1	1955	840	1	2	2
3398	14	Brick	1	1953	763	1	2	2
3399	14	Brick	2	1929	1629	1	3	1
3400	14	Aluminum / Vinyl	1	1910	1560	1	3	1
3401	14	Frame	1.5	1920	1339	1	3	1
3402	14	Aluminum / Vinyl	1	1921	1139	1	3	2
3403	15	Frame	2	1970	1350	1	4	1
3404	15	Frame	2	1923	2754	2	4	2
3405	15	Frame	2	1923	2232	2	4	2
3406	15	Aluminum / Vinyl	2	1923	2217	2	4	2
3407	15	Frame	2	1913	2189	2	4	2
3408	1	Stone	1	1939	1346	1	3	1
3409	1	Aluminum / Vinyl	1	1951	1158	1	3	1
3410	1	Brick	1	1936	1078	1	3	1
3411	1	Aluminum / Vinyl	1	1951	1002	1	3	2
3412	1	Aluminum / Vinyl	1	1979	1215	1	3	1
3413	1	Stone	1	1936	1047	1	2	1
3414	1	Stone	1	1936	1047	1	2	1
3415	1	Aluminum / Vinyl	1	1954	983	1	3	1

3416	1	Masonry / Frame	1	1959	1698	1	3	2
3417	2	Brick	1	1960	1220	1	3	1
3418	2	Aluminum / Vinyl	1	1958	1039	1	3	1
3419	2	Aluminum / Vinyl	1	1954	980	1	2	1
3420	2	Frame	1	1955	950	1	3	1
3421	2	Brick	1	1953	825	1	2	1
3422	3	Aluminum / Vinyl	1.5	1926	1877	2	4	2
3423	3	Aluminum / Vinyl	2	1911	2310	2	4	2
3424	3	Brick	2	1928	3768	1	5	3
3425	3	Frame	2	1901	2419	1	5	2
3426	3	Aluminum / Vinyl	2	1902	2383	1	4	1
3427	3	Aluminum / Vinyl	2	1906	2185	1	6	1
3428	3	Brick	2	1921	3140	1	3	2
3429	5	Aluminum / Vinyl	1.5	1953	1434	1	3	1
3430	5	Aluminum / Vinyl	1	1952	1381	1	3	2
3431	5	Aluminum / Vinyl	1	1949	1180	1	3	1
3432	5	Aluminum / Vinyl	1	1949	1170	1	3	2
3433	5	Brick	1	1946	1128	1	3	1
3434	5	Aluminum / Vinyl	1	1948	1049	1	3	1
3435	5	Aluminum / Vinyl	1	1951	728	1	2	1
3436	5	Aluminum / Vinyl	2	2010	2473	1	4	2
3437	5	Brick	1.5	1952	2092	2	3	2
3438	5	Aluminum / Vinyl	1.5	1958	1814	2	5	2
3439	5	Stone	1	1955	2290	1	3	2
3440	5	Aluminum / Vinyl	1	2010	2222	1	3	2
3441	5	Brick	1	1956	1157	1	3	1
3442	5	Brick	1	1959	1153	1	3	1
3443	6	Fiber-Cement	2	2003	1848	1	3	2
3444	6	Aluminum / Vinyl	2	1914	2422	2	4	3
3445	6	Frame	2	1906	1935	2	4	2
3446	6	Aluminum / Vinyl	1	1900	1107	1	3	1
3447	7	Stucco	1.5	1942	1827	1	3	1
3448	7	Brick	1	1940	1198	1	2	1
3449	7	Masonry / Frame	2	1938	1411	1	3	1
3450	7	Aluminum / Vinyl	2	1972	1134	1	4	1
3451	7	Frame	2	1925	2184	2	4	2
3452	7	Aluminum / Vinyl	1	1946	744	1	2	1
3453	8	Aluminum / Vinyl	1	1890	1415	1	2	2
3454	8	Aluminum / Vinyl	1.5	1920	1855	2	4	2
3455	8	Brick	2	1959	2624	2	4	2
3456	8	Aluminum / Vinyl	1.5	1926	1819	2	4	2
3457	8	Aluminum / Vinyl	2	1890	1408	2	4	2
3458	8	Aluminum / Vinyl	1	1896	1482	1	4	1

3459	8	Aluminum / Vinyl	1.5	1917	1357	1	2	1
3460	8	Aluminum / Vinyl	1	1920	1322	1	3	2
3461	9	Aluminum / Vinyl	2	1957	1940	2	6	2
3462	9	Masonry / Frame	1	1959	1610	1	4	3
3463	9	Aluminum / Vinyl	1	1971	1210	1	4	1
3464	9	Aluminum / Vinyl	1	1957	1149	1	3	1
3465	10	Stone	1.5	1941	1889	1	4	2
3466	10	Aluminum / Vinyl	1	1946	1030	1	3	2
3467	10	Brick	2	1935	1692	1	4	2
3468	10	Masonry / Frame	2	1953	1657	1	3	1
3469	10	Block	2	1916	2130	2	6	2
3470	10	Brick	1	1931	1874	1	4	1
3471	10	Brick	1	1921	1857	1	4	1
3472	10	Frame	1	1920	1654	1	4	1
3473	10	Aluminum / Vinyl	1	1922	1495	1	4	1
3474	10	Aluminum / Vinyl	1	1929	1324	1	3	1
3475	10	Aluminum / Vinyl	1	1925	1152	1	3	1
3476	10	Brick	1	1951	762	1	2	1
3477	10	Aluminum / Vinyl	1	1925	768	1	3	1
3478	11	Stone	1	1946	1691	1	4	1
3479	11	Stone	1	1939	1320	1	3	1
3480	11	Brick	1	1951	1309	1	3	1
3481	11	Brick	1	1949	1079	1	3	1
3482	11	Masonry / Frame	2	1936	1376	1	3	1
3483	11	Aluminum / Vinyl	2	1953	1554	2	4	2
3484	11	Brick	1.5	1939	1868	2	4	2
3485	11	Aluminum / Vinyl	1.5	1920	1844	1	5	2
3486	11	Frame	1	1956	1490	1	5	1
3487	11	Aluminum / Vinyl	1	1977	1334	1	3	2
3488	11	Masonry / Frame	1	1959	1319	1	3	1
3489	11	Aluminum / Vinyl	1	1959	1190	1	3	1
3490	11	Brick	1	1958	1114	1	3	1
3491	11	Frame	1	1955	1008	1	3	1
3492	11	Aluminum / Vinyl	1	1954	972	1	3	1
3493	12	Stucco	1.5	1900	1630	2	4	2
3494	12	Aluminum / Vinyl	1	1903	1030	1	3	1
3495	13	Brick	1.5	1969	1718	1	4	2
3496	13	Aluminum / Vinyl	1	1951	1304	1	4	1
3497	13	Aluminum / Vinyl	1	1939	1301	1	2	2
3498	13	Brick	1	1949	1158	1	3	1
3499	13	Brick	1.5	1938	1148	1	2	2
3500	13	Aluminum / Vinyl	2	1941	1252	1	3	1
3501	13	Aluminum / Vinyl	2	1953	1082	1	2	1

3502	13	Brick	2	1955	2160	2	4	2
3503	13	Brick	1.5	1947	2019	2	3	2
3504	13	Brick	1	1955	1032	1	3	1
3505	13	Aluminum / Vinyl	1	1960	891	1	3	1
3506	14	Brick	1	1953	1342	1	4	1
3507	14	Aluminum / Vinyl	1	1941	826	1	3	2
3508	14	Brick	1.5	1939	2042	2	4	2
3509	14	Brick	2	1942	2016	2	4	2
3510	14	Brick	1	1926	1902	1	4	2
3511	14	Aluminum / Vinyl	1	1931	1669	1	4	2
3512	14	Brick	1	1929	1407	1	3	2
3513	14	Aluminum / Vinyl	1	1927	1359	1	4	1
3514	14	Frame	1	1922	1324	1	3	1
3515	14	Aluminum / Vinyl	1	1927	1198	1	3	1
3516	14	Aluminum / Vinyl	1	1926	1029	1	2	1
3517	14	Aluminum / Vinyl	1	1942	791	1	2	1
3518	14	Aluminum / Vinyl	1.5	1925	1449	1	4	1
3519	15	Frame	2	1915	2316	2	4	2
3520	15	Aluminum / Vinyl	2	1893	2080	2	4	2
3521	15	Brick	1	1921	2287	1	3	2
3522	15	Aluminum / Vinyl	2	1913	1674	1	5	1
3523	15	Frame	1	1914	1624	1	3	1
3524	1	Brick	1	1949	1059	1	3	1
3525	1	Aluminum / Vinyl	2	1966	1649	1	4	1
3526	1	Brick	1	1925	1469	1	4	2
3527	1	Aluminum / Vinyl	1	1931	794	1	2	1
3528	2	Aluminum / Vinyl	1	1941	1109	1	4	1
3529	2	Stone	1	1962	1280	1	3	1
3530	2	Aluminum / Vinyl	1	1966	1077	1	3	1
3531	2	Aluminum / Vinyl	1	1961	1053	1	3	1
3532	2	Stone	1	1951	964	1	3	1
3533	2	Aluminum / Vinyl	1	1955	912	1	3	1
3534	3	Aluminum / Vinyl	1	1946	700	1	2	1
3535	3	Frame	1.5	1914	2072	2	3	2
3536	3	Frame	1	1926	1422	1	3	1
3537	3	Aluminum / Vinyl	1	1922	1392	1	4	1
3538	3	Aluminum / Vinyl	2	1908	2156	1	4	1
3539	3	Aluminum / Vinyl	2	1897	1530	1	3	1
3540	5	Frame	1	1953	1494	1	3	1
3541	5	Brick	1	1949	1260	1	3	1
3542	5	Brick	1	1947	801	1	2	1
3543	5	Aluminum / Vinyl	2	2007	1984	1	3	2
3544	5	Aluminum / Vinyl	2	1968	1631	1	4	1

3545	5	Frame	2	1959	1484	1	4	1
3546	5	Aluminum / Vinyl	2	1963	1200	1	3	1
3547	5	Brick	1	1963	2235	2	5	2
3548	5	Brick	1.5	1956	1818	2	4	2
3549	5	Aluminum / Vinyl	1	1971	1499	1	3	1
3550	5	Frame	1	1956	1276	1	3	2
3551	5	Brick	1	1960	1237	1	3	2
3552	5	Aluminum / Vinyl	1	1964	1212	1	3	1
3553	5	Block	1	1942	1170	1	2	2
3554	5	Aluminum / Vinyl	1	1949	1056	1	3	1
3555	5	Brick	1	1958	1036	1	2	1
3556	5	Aluminum / Vinyl	1	1950	1000	1	2	1
3557	5	Aluminum / Vinyl	1	1955	948	1	3	1
3558	5	Aluminum / Vinyl	1	1949	624	1	2	1
3559	5	Aluminum / Vinyl	1	1993	2240	1	4	3
3560	6	Frame	2	1923	2092	2	2	2
3561	6	Frame	1	1925	1212	1	3	1
3562	6	Aluminum / Vinyl	2	1904	1610	1	6	1
3563	7	Brick	1	1950	1271	1	3	1
3564	7	Brick	2	1958	2774	2	6	4
3565	7	Aluminum / Vinyl	1.5	1923	2024	1	5	2
3566	7	Aluminum / Vinyl	1.5	1923	2024	1	5	2
3567	8	Masonry / Frame	2	1928	2200	2	4	2
3568	8	Frame	1	1924	1248	1	4	1
3569	9	Aluminum / Vinyl	1	1952	1480	1	3	2
3570	9	Aluminum / Vinyl	1	1956	1192	1	4	1
3571	9	Aluminum / Vinyl	2	2004	2561	1	4	2
3572	9	Aluminum / Vinyl	2	1987	1737	1	3	1
3573	9	Brick	1	1957	1282	1	3	1
3574	9	Aluminum / Vinyl	1	1975	1227	1	3	2
3575	9	Brick	1	1958	1112	1	3	2
3576	9	Brick	1	1958	1100	1	3	1
3577	9	Frame	1	1962	1045	1	3	1
3578	9	Aluminum / Vinyl	1	1957	948	1	3	1
3579	10	Brick	1	1957	1892	1	5	2
3580	10	Aluminum / Vinyl	1	1950	1037	1	4	1
3581	10	Aluminum / Vinyl	2	1969	2094	2	6	2
3582	10	Brick	1.5	1949	1529	2	3	2
3583	10	Brick	1	1927	1874	1	4	2
3584	10	Aluminum / Vinyl	1	1926	1055	1	3	1
3585	10	Aluminum / Vinyl	1	1949	945	1	2	1
3586	10	Frame	1.5	1925	1406	1	3	2
3587	10	Aluminum / Vinyl	1	1920	1099	1	3	1

3588	11	Brick	1	1954	1645	1	4	1
3589	11	Brick	1	1945	1243	1	3	1
3590	11	Frame	1	1958	1211	1	4	1
3591	11	Brick	1.5	1938	1164	1	3	1
3592	11	Aluminum / Vinyl	1	1949	1086	1	3	1
3593	11	Aluminum / Vinyl	1	1954	952	1	3	1
3594	11	Masonry / Frame	2	1959	1682	1	3	1
3595	11	Aluminum / Vinyl	2	1958	2249	2	6	2
3596	11	Aluminum / Vinyl	1.5	1955	1587	2	4	2
3597	11	Aluminum / Vinyl	1	1966	1853	1	3	2
3598	11	Aluminum / Vinyl	1	1953	1285	1	3	1
3599	11	Frame	1	1947	1254	1	3	2
3600	11	Stone	1	1950	1251	1	2	2
3601	11	Frame	1	1957	1158	1	3	1
3602	11	Brick	1	1958	1114	1	3	1
3604	11	Brick	1	1956	1025	1	3	1
3605	11	Frame	1	1972	1944	1	3	2
3606	11	Aluminum / Vinyl	2	1968	3214	3	7	3
3607	12	Aluminum / Vinyl	1	1880	950	1	3	1
3608	12	Aluminum / Vinyl	2	1914	2328	2	5	2
3609	12	Frame	2	1890	2297	2	6	2
3610	13	Aluminum / Vinyl	1	1945	1184	1	3	1
3611	13	Aluminum / Vinyl	1	1943	1006	1	3	1
3612	13	Prem Wood	2	1974	1980	1	5	1
3613	13	Aluminum / Vinyl	1.5	1928	1734	2	4	2
3614	13	Aluminum / Vinyl	1	1950	956	1	3	1
3615	13	Brick	1	1955	811	1	2	1
3616	13	Aluminum / Vinyl	1	1954	768	1	2	1
3617	13	Aluminum / Vinyl	1	1952	705	1	2	1
3618	13	Aluminum / Vinyl	2	1928	1234	1	3	1
3619	13	Aluminum / Vinyl	1.5	1924	1073	1	3	1
3620	14	Aluminum / Vinyl	1	1953	1056	1	3	1
3621	14	Aluminum / Vinyl	1	1953	784	1	2	1
3622	14	Aluminum / Vinyl	2	1909	2746	2	6	2
3623	14	Aluminum / Vinyl	1.5	1900	1566	2	4	2
3624	14	Frame	1	1900	1233	1	2	1
3625	14	Frame	1	1928	1094	1	2	1
3626	14	Aluminum / Vinyl	1	1924	1028	1	2	1
3627	15	Aluminum / Vinyl	1	2000	1430	1	2	1
3628	15	Aluminum / Vinyl	2	1969	1042	1	3	1
3629	15	Aluminum / Vinyl	1	1922	760	1	1	1
3630	15	Stucco	2	1916	2964	2	6	2
3631	15	Aluminum / Vinyl	2	1909	2350	2	6	2

3632	15	Brick	1.5	1924	2435	1	5	2
3633	15	Brick	1	1920	1960	1	3	2
3634	15	Aluminum / Vinyl	1.5	1890	1358	1	4	1
3635	1	Brick	1	1950	1001	1	3	1
3636	1	Aluminum / Vinyl	2	1967	2010	2	6	2
3637	1	Frame	1	1963	1633	1	3	1
3638	1	Aluminum / Vinyl	1	1974	1503	1	3	1
3639	1	Aluminum / Vinyl	1	1955	1082	1	3	1
3640	1	Aluminum / Vinyl	1	1947	1041	1	3	1
3641	1	Aluminum / Vinyl	1	1951	713	1	2	1
3642	2	Frame	1	1953	1268	1	4	2
3643	2	Aluminum / Vinyl	1	1955	1176	1	4	1
3644	2	Frame	1	1935	740	1	2	1
3645	2	Brick	1	1959	1173	1	3	1
3646	2	Aluminum / Vinyl	1	1956	982	1	3	1
3647	2	Brick	1	1957	942	1	3	1
3648	3	Frame	2	2003	2793	1	3	2
3649	3	Brick	2	1906	4098	1	6	3
3650	3	Brick	1	1927	1686	1	3	2
3651	3	Frame	1	1904	968	1	3	1
3652	5	Frame	1.5	1958	1664	1	4	1
3653	5	Brick	1	1952	1642	1	4	1
3654	5	Brick	1	1948	1276	1	3	1
3655	5	Brick	1	1949	1245	1	3	1
3656	5	Aluminum / Vinyl	1	1953	1005	1	3	2
3657	5	Brick	1	1956	941	1	2	1
3658	5	Aluminum / Vinyl	2	1950	1344	1	3	2
3659	5	Masonry / Frame	2	1956	2282	2	5	2
3660	5	Aluminum / Vinyl	1	1960	1384	1	3	2
3661	5	Brick	1	1959	1287	1	3	1
3662	5	Aluminum / Vinyl	1	1956	1206	1	3	1
3663	5	Aluminum / Vinyl	1	1957	997	1	3	1
3664	6	Aluminum / Vinyl	2	2004	1564	1	4	2
3665	6	Aluminum / Vinyl	2	1911	2196	2	4	2
3666	6	Aluminum / Vinyl	2	1905	2332	2	4	2
3667	6	Aluminum / Vinyl	1	1885	983	1	3	1
3668	7	Aluminum / Vinyl	1	1929	1223	1	3	1
3669	7	Aluminum / Vinyl	1	1923	1222	1	3	1
3670	8	Aluminum / Vinyl	2	1910	2022	2	4	2
3671	8	Aluminum / Vinyl	1	1925	1547	1	4	2
3672	8	Stucco	1	1913	1262	1	4	1
3673	9	Aluminum / Vinyl	2	1957	1940	2	6	2
3674	9	Aluminum / Vinyl	1.5	1957	1749	2	4	3

3675	9	Aluminum / Vinyl	1	2009	2041	1	3	2
3676	9	Frame	1	1961	1727	1	3	1
3677	9	Aluminum / Vinyl	1	1968	1556	1	3	1
3678	9	Aluminum / Vinyl	1	1956	1132	1	3	1
3679	9	Frame	1	1971	1120	1	4	1
3680	9	Aluminum / Vinyl	1	1971	1087	1	4	1
3681	9	Block	1	1946	948	1	2	1
3682	10	Aluminum / Vinyl	1	1926	760	1	2	1
3683	10	Brick	2	1926	2953	2	4	2
3684	10	Aluminum / Vinyl	2	1924	2462	2	4	2
3685	10	Aluminum / Vinyl	1	1925	1627	1	4	2
3686	10	Brick	1	1927	1371	1	3	1
3687	10	Aluminum / Vinyl	1	1951	1188	1	3	2
3688	10	Brick	1	1953	961	1	3	1
3689	10	Aluminum / Vinyl	1	1951	824	1	2	1
3690	10	Stucco	1	1916	1617	1	5	3
3691	10	Aluminum / Vinyl	1	1930	821	1	2	1
3692	10	Stone	1	1933	1974	1	5	2
3693	11	Brick	1	1951	1213	1	3	1
3694	11	Brick	1	1952	1188	1	3	1
3695	11	Aluminum / Vinyl	1	1958	1125	1	4	1
3696	11	Aluminum / Vinyl	1	1957	1108	1	3	1
3697	11	Frame	1	1928	1188	1	3	1
3698	11	Brick	1	1966	1437	1	3	1
3699	11	Brick	1	1963	1290	1	3	1
3700	11	Aluminum / Vinyl	1	1959	1228	1	3	1
3701	11	Brick	1	1963	1216	1	3	1
3702	11	Brick	1	1957	1152	1	3	2
3703	11	Aluminum / Vinyl	1	1962	1092	1	3	1
3704	11	Aluminum / Vinyl	1	1959	1086	1	3	1
3705	11	Brick	1	1954	1023	1	2	1
3706	11	Brick	1	1960	1014	1	3	1
3707	11	Stucco	1	1954	972	1	3	1
3708	11	Brick	1	1955	934	1	3	1
3709	11	Aluminum / Vinyl	1	1953	879	1	3	1
3710	11	Brick	1	1939	858	1	2	1
3711	11	Aluminum / Vinyl	1	1953	672	1	2	1
3712	12	Aluminum / Vinyl	1	1923	1172	1	2	1
3713	13	Brick	1	1953	1080	1	2	1
3714	13	Masonry / Frame	2	1973	2216	1	4	2
3715	13	Masonry / Frame	2	1974	1957	1	3	1
3716	13	Aluminum / Vinyl	1.5	1895	1630	2	4	2
3717	13	Brick	1	1955	1155	1	3	2

3718	13	Brick	1	1959	1133	1	3	1
3719	14	Aluminum / Vinyl	1	1940	1307	1	4	1
3720	14	Aluminum / Vinyl	1	1953	1034	1	3	1
3721	14	Aluminum / Vinyl	1	1895	960	1	3	1
3722	14	Brick	1.5	1925	2343	2	5	3
3723	14	Aluminum / Vinyl	2	1973	2138	2	6	2
3724	14	Frame	2	1956	1529	2	4	2
3725	14	Frame	2	1904	2545	2	6	2
3726	14	Brick	2	1871	2502	2	6	2
3727	14	Frame	2	1910	2288	2	4	2
3728	14	Aluminum / Vinyl	2	1910	2274	2	5	2
3729	14	Frame	1.5	1914	1874	2	4	2
3730	14	Aluminum / Vinyl	1	1926	1317	1	3	1
3731	14	Aluminum / Vinyl	1	1922	990	1	3	1
3732	14	Aluminum / Vinyl	2	1907	2138	1	5	1
3733	14	Aluminum / Vinyl	1	1915	1526	1	4	1
3734	15	Aluminum / Vinyl	2	2010	1860	1	4	3
3735	15	Aluminum / Vinyl	2	1917	2945	2	6	2
3736	15	Frame	2	1893	2140	2	4	2
3737	15	Aluminum / Vinyl	2	1891	1958	2	4	2
3738	15	Frame	1	1920	1476	1	4	1
3739	15	Aluminum / Vinyl	1	1970	1101	1	3	1
3740	15	Aluminum / Vinyl	1	1895	847	1	2	1
3741	1	Aluminum / Vinyl	1	1960	964	1	3	1
3742	1	Frame	2	1926	1559	1	4	1
3743	2	Aluminum / Vinyl	1.5	1955	1402	1	4	1
3744	2	Brick	1	1953	1389	1	3	1
3745	2	Aluminum / Vinyl	1	1958	1266	1	3	2
3746	2	Aluminum / Vinyl	1	1955	1176	1	2	1
3747	2	Aluminum / Vinyl	1	1952	914	1	3	1
3748	2	Aluminum / Vinyl	1.5	1958	1535	2	3	2
3749	2	Aluminum / Vinyl	1	1963	2060	1	5	2
3750	2	Aluminum / Vinyl	1	1955	864	1	3	1
3751	3	Brick	2	1902	4277	1	6	3
3752	3	Brick	2	1915	3618	1	3	2
3753	3	Frame	2	1890	3025	1	5	2
3754	3	Stucco	2	1894	2348	1	4	1
3755	5	Brick	1	1954	1504	1	4	2
3756	5	Aluminum / Vinyl	1	1941	1434	1	4	2
3757	5	Frame	1	1957	1216	1	4	1
3758	5	Brick	1	1947	1168	1	3	1
3759	5	Masonry / Frame	2	1954	2523	1	3	2
3760	5	Masonry / Frame	2	1948	1629	1	3	1

3761	5	Masonry / Frame	2	1939	1393	1	3	1
3762	5	Brick	2	1959	2623	2	6	2
3763	5	Brick	1	1954	1686	1	2	1
3764	5	Brick	1	1963	1293	1	3	1
3765	5	Brick	1	1965	1288	1	3	2
3766	6	Aluminum / Vinyl	2	1923	2266	2	4	2
3767	6	Aluminum / Vinyl	2	1892	2948	1	5	1
3768	7	Brick	1	1939	1542	1	3	2
3769	7	Brick	2	1940	2431	1	4	2
3770	7	Masonry / Frame	2	1939	1511	1	3	1
3771	7	Masonry / Frame	2	1930	2323	2	4	2
3772	7	Frame	1.5	1929	1451	2	4	3
3773	7	Stone	1	1950	1262	1	2	1
3774	7	Brick	1	1950	1249	1	3	1
3775	8	Aluminum / Vinyl	2	1914	2304	2	6	2
3776	8	Stucco	1	1917	1695	1	4	2
3777	8	Aluminum / Vinyl	1	1906	1411	1	5	1
3778	9	Aluminum / Vinyl	2	2005	3722	1	4	3
3779	9	Aluminum / Vinyl	1	1968	1399	1	3	1
3780	9	Aluminum / Vinyl	1	1958	1089	1	3	1
3781	9	Aluminum / Vinyl	1	1971	1086	1	4	2
3782	9	Brick	1	1957	1044	1	3	1
3783	9	Aluminum / Vinyl	1	1976	906	1	3	1
3784	9	Aluminum / Vinyl	2	1920	2797	1	6	2
3785	10	Aluminum / Vinyl	1	1952	1277	1	4	1
3786	10	Aluminum / Vinyl	1	1937	883	1	3	1
3787	10	Aluminum / Vinyl	1.5	1928	2067	2	4	2
3788	10	Brick	2	1936	2316	2	4	2
3789	10	Aluminum / Vinyl	2	1930	1232	2	2	2
3790	10	Aluminum / Vinyl	2	1913	2792	2	5	2
3791	10	Stucco	2	1920	1276	2	3	2
3792	10	Brick	1	1927	1816	1	4	2
3793	10	Aluminum / Vinyl	1	1919	1728	1	4	2
3794	10	Frame	1	1928	1239	1	3	2
3795	10	Aluminum / Vinyl	1.5	1914	1913	1	3	1
3796	10	Frame	1	1907	1412	1	3	2
3797	11	Brick	2	1947	1349	1	3	1
3798	11	Brick	1	1964	1411	1	3	2
3799	11	Aluminum / Vinyl	1	1956	1108	1	3	1
3800	12	Aluminum / Vinyl	2	1907	1948	2	6	2
3801	12	Brick	2	1900	1786	1	3	1
3802	13	Aluminum / Vinyl	1.5	1946	1544	1	3	1
3803	13	Brick	1	1950	1517	1	4	1

3804	13	Brick	1	1953	1203	1	3	1
3805	13	Brick	1	1949	1081	1	3	1
3806	13	Aluminum / Vinyl	2	1948	1056	1	2	2
3807	13	Aluminum / Vinyl	1.5	1928	1920	2	4	2
3808	13	Aluminum / Vinyl	1.5	1928	1461	2	4	2
3809	13	Brick	1	1969	1590	1	3	1
3810	13	Aluminum / Vinyl	1	1959	1085	1	3	2
3811	13	Aluminum / Vinyl	1	1960	1047	1	3	1
3812	13	Aluminum / Vinyl	1	1947	732	1	2	1
3813	13	Aluminum / Vinyl	1	1908	1264	1	3	2
3814	13	Frame	1	1930	779	1	1	1
3815	14	Stone	1	1937	951	1	2	1
3816	14	Stone	2	1941	1944	1	3	1
3817	14	Aluminum / Vinyl	2	1942	1929	1	3	1
3818	14	Aluminum / Vinyl	1.5	1913	2034	2	5	2
3819	14	Brick	1	1930	2004	1	3	2
3820	14	Aluminum / Vinyl	1	1952	672	1	2	1
3821	14	Aluminum / Vinyl	1	1900	1242	1	3	1
3822	14	Stone	1.5	1936	1358	1	3	2
3823	15	Aluminum / Vinyl	2	2002	1184	1	3	1
3824	15	Aluminum / Vinyl	2	1911	2034	2	5	2
3825	15	Aluminum / Vinyl	1	1969	1098	1	3	1
3826	15	Brick	2	1921	2300	1	3	1
3827	1	Frame	1.5	1951	1576	1	3	2
3828	1	Aluminum / Vinyl	1.5	1925	1539	2	3	2
3829	1	Masonry / Frame	2	1927	2186	2	4	2
3830	1	Brick	1	1930	1289	1	3	1
3831	1	Aluminum / Vinyl	1	1957	894	1	3	1
3832	1	Aluminum / Vinyl	1	1950	768	1	2	1
3833	1	Frame	1	1929	1273	1	3	1
3834	1	Frame	1	1929	1273	1	3	1
3835	1	Brick	1.5	1931	1784	1	3	2
3836	1	Stone	1.5	1938	1586	1	3	1
3837	2	Brick	1	1950	1978	1	5	2
3838	2	Frame	1	1961	1268	1	4	1
3839	2	Frame	1	1961	1147	1	4	1
3840	2	Aluminum / Vinyl	1	1957	1130	1	3	1
3841	2	Brick	1	1959	1322	1	3	1
3842	2	Aluminum / Vinyl	1	1956	1120	1	3	1
3843	2	Brick	1	1957	1083	1	3	1
3844	2	Frame	1	1955	912	1	3	1
3845	3	Stucco	1.5	1914	2708	1	5	3
3846	3	Masonry / Frame	2	1909	2633	1	6	2

3847	3	Aluminum / Vinyl	1.5	1902	1627	1	4	2
3848	3	Frame	1	1924	898	1	2	1
3849	5	Brick	1	1949	1466	1	3	3
3850	5	Brick	1	1947	1251	1	3	1
3851	5	Aluminum / Vinyl	1	1950	1242	1	3	1
3852	5	Aluminum / Vinyl	1	1952	1231	1	4	1
3853	5	Stone	1	1947	1134	1	2	1
3854	5	Frame	1	1953	1108	1	4	1
3855	5	Fiber-Cement	2	2005	3583	1	5	4
3856	5	Masonry / Frame	2	1956	2189	2	5	2
3857	5	Brick	1	1953	1220	1	3	1
3858	5	Frame	1	1957	1060	1	3	1
3859	5	Aluminum / Vinyl	1	1959	988	1	3	2
3860	5	Frame	1	1958	919	1	3	1
3861	6	Aluminum / Vinyl	1.5	1890	1817	2	4	2
3862	7	Brick	1.5	1950	1604	1	3	1
3863	7	Aluminum / Vinyl	2	1940	1128	1	3	1
3864	7	Aluminum / Vinyl	1	1890	552	1	2	1
3865	7	Frame	2	1922	2608	2	6	2
3866	7	Aluminum / Vinyl	1.5	1920	1720	2	4	2
3867	7	Aluminum / Vinyl	1	1930	1680	1	4	2
3868	7	Aluminum / Vinyl	1	1928	1502	1	4	1
3869	8	Aluminum / Vinyl	1	1923	1141	1	3	1
3870	8	Aluminum / Vinyl	2	1898	1496	1	4	1
3872	9	Brick	1	1960	1527	1	3	1
3873	9	Brick	1	1957	1356	1	3	1
3874	9	Aluminum / Vinyl	1	1956	1232	1	3	2
3875	9	Brick	1	1957	1176	1	3	1
3876	9	Brick	1	1958	1124	1	3	1
3877	9	Frame	1	1957	1073	1	3	2
3878	9	Aluminum / Vinyl	1	1963	987	1	3	1
3879	10	Stone	1.5	1950	2125	1	5	2
3880	10	Masonry / Frame	2	1953	1657	1	3	1
3881	10	Aluminum / Vinyl	2	1927	2094	2	4	2
3882	10	Stone	1	1938	1112	1	2	1
3883	10	Aluminum / Vinyl	1	1953	1025	1	2	1
3884	10	Stucco	1	1915	1598	1	4	1
3885	10	Masonry / Frame	2	1928	1439	1	3	1
3886	10	Aluminum / Vinyl	1.5	1920	1186	1	3	1
3887	11	Aluminum / Vinyl	1	1949	1211	1	3	1
3888	11	Aluminum / Vinyl	1	1955	1155	1	4	1
3889	11	Aluminum / Vinyl	1	1954	1124	1	4	2
3890	11	Frame	1	1953	958	1	3	1

3891	11	Block	2	1945	1252	1	3	1
3892	11	Aluminum / Vinyl	1	1925	1883	1	5	2
3893	11	Masonry / Frame	1	1957	1914	1	2	2
3894	11	Brick	1	1957	1160	1	3	1
3895	11	Aluminum / Vinyl	1	1958	1138	1	3	1
3896	11	Brick	1	1953	1058	1	3	1
3897	11	Brick	1	1955	1019	1	3	2
3898	11	Aluminum / Vinyl	1	1960	981	1	3	1
3899	11	Aluminum / Vinyl	1	1953	890	1	3	1
3900	11	Frame	2	1920	1531	1	3	1
3901	12	Aluminum / Vinyl	1.5	1902	1693	1	4	2
3902	12	Aluminum / Vinyl	1	1885	1320	1	2	1
3903	13	Brick	1.5	1954	1812	1	4	2
3904	13	Aluminum / Vinyl	1.5	1939	1334	1	3	1
3905	13	Brick	1	1952	1070	1	3	1
3906	13	Brick	2	1939	1443	1	3	1
3907	13	Aluminum / Vinyl	1	1926	947	1	2	1
3908	13	Aluminum / Vinyl	1	1968	1392	1	3	1
3909	13	Brick	1	1967	1364	1	3	1
3910	13	Aluminum / Vinyl	1	1950	756	1	2	1
3911	14	Aluminum / Vinyl	2	1948	1378	1	3	1
3912	14	Aluminum / Vinyl	2	1941	1090	1	2	1
3913	14	Aluminum / Vinyl	2	1913	1696	2	4	2
3914	14	Aluminum / Vinyl	1	1925	1529	1	3	1
3915	14	Aluminum / Vinyl	1	1928	1504	1	3	2
3916	14	Aluminum / Vinyl	1	1924	1390	1	4	1
3917	14	Aluminum / Vinyl	1	1926	1237	1	3	2
3918	14	Aluminum / Vinyl	1	1900	945	1	2	1
3919	14	Aluminum / Vinyl	1	1956	1078	1	3	1
3920	14	Aluminum / Vinyl	1.5	1910	1669	1	3	1
3921	14	Aluminum / Vinyl	2	1913	1512	1	3	1
3922	14	Stone	1.5	1936	1684	1	3	2
3923	15	Stucco	1	1920	1855	1	4	2
3924	15	Brick	1.5	1927	2441	1	3	1
3925	15	Aluminum / Vinyl	1	1900	1386	1	3	1
3926	1	Brick	1	1944	1318	1	4	2
3927	1	Frame	2	1941	1843	1	3	1
3928	1	Frame	2	1949	1818	2	4	2
3929	1	Brick	1	1926	2152	1	5	2
3930	2	Aluminum / Vinyl	1.5	1963	1909	2	5	2
3931	2	Stone	1	1955	1131	1	3	1
3932	2	Aluminum / Vinyl	1	1959	942	1	3	1
3933	3	Frame	1	1925	1052	1	3	1

3934	3	Stucco	2	1909	3518	1	6	3
3935	5	Brick	1	1947	2014	1	5	2
3936	5	Brick	1	1947	1503	1	3	1
3937	5	Brick	1	1953	1322	1	4	2
3938	5	Aluminum / Vinyl	1	1950	1171	1	4	1
3939	5	Aluminum / Vinyl	1	1950	1156	1	3	2
3940	5	Aluminum / Vinyl	1	1952	1092	1	4	1
3941	5	Aluminum / Vinyl	1	1952	1074	1	3	1
3942	5	Aluminum / Vinyl	1	1952	1039	1	3	1
3943	5	Aluminum / Vinyl	1	1950	1022	1	3	1
3944	5	Aluminum / Vinyl	1	1952	1016	1	3	1
3945	5	Aluminum / Vinyl	1	1950	948	1	3	1
3946	5	Brick	2	1957	2584	2	6	2
3947	5	Aluminum / Vinyl	1	1962	1420	1	3	1
3948	5	Frame	1	1955	1064	1	3	1
3949	5	Aluminum / Vinyl	1	1950	912	1	3	1
3950	5	Aluminum / Vinyl	1	1957	1487	1	3	2
3951	6	Brick	2	1922	2714	2	6	2
3952	6	Aluminum / Vinyl	1.5	1915	1809	2	4	2
3953	6	Aluminum / Vinyl	1	1927	955	1	2	1
3954	7	Aluminum / Vinyl	1	1950	1200	1	3	1
3955	7	Aluminum / Vinyl	1.5	1941	1824	2	3	2
3956	8	Frame	2	1923	2425	2	4	2
3957	8	Aluminum / Vinyl	1.5	1922	1702	2	3	2
3958	8	Aluminum / Vinyl	1.5	1910	1722	2	3	2
3959	8	Aluminum / Vinyl	1	1925	952	1	3	1
3960	8	Aluminum / Vinyl	1	1900	1320	1	4	2
3961	9	Aluminum / Vinyl	1	1977	1219	1	3	1
3962	10	Brick	1	1945	1098	1	3	2
3963	10	Aluminum / Vinyl	2	1926	2256	2	4	2
3964	10	Brick	1.5	1927	1949	1	4	1
3965	10	Brick	1	1927	1856	1	4	2
3966	10	Aluminum / Vinyl	1	1926	1617	1	5	1
3967	10	Stucco	1	1915	1565	1	3	2
3968	10	Brick	1	1927	1272	1	2	1
3969	11	Stone	1	1947	1356	1	3	2
3970	11	Aluminum / Vinyl	1	1952	1347	1	4	1
3971	11	Stone	1	1954	1310	1	3	1
3972	11	Aluminum / Vinyl	1	1953	1106	1	3	1
3973	11	Aluminum / Vinyl	2	1955	1521	1	5	2
3974	11	Brick	2	1944	1564	2	4	2
3975	11	Frame	1	1956	1363	1	3	1
3976	11	Brick	1	1960	1286	1	3	1

3977	11	Brick	1	1952	1269	1	3	1
3978	11	Prem Wood	1	1958	1252	1	3	1
3979	11	Brick	1	1960	1138	1	3	1
3980	11	Brick	1	1954	1104	1	3	1
3981	12	Aluminum / Vinyl	1	1870	452	1	1	1
3982	13	Stucco	1	1930	1487	1	3	2
3983	13	Brick	2	1964	2736	3	5	3
3984	14	Aluminum / Vinyl	1	1953	1166	1	4	1
3985	14	Brick	1.5	1926	2250	2	5	2
3986	14	Aluminum / Vinyl	1.5	1926	1929	2	4	2
3987	14	Aluminum / Vinyl	2	1891	2424	2	6	2
3988	14	Aluminum / Vinyl	2	1921	2406	2	5	2
3989	14	Aluminum / Vinyl	2	1900	1620	2	4	2
3990	14	Frame	2	1897	1916	1	3	2
3991	15	Aluminum / Vinyl	2	1922	2801	2	4	2
3992	15	Aluminum / Vinyl	1.5	1924	1826	2	5	2
3993	1	Stone	1	1951	1198	1	3	1
3994	1	Aluminum / Vinyl	1	1925	1016	1	3	1
3995	2	Frame	1.5	1951	1673	1	4	2
3996	2	Aluminum / Vinyl	1	1956	853	1	2	1
3997	2	Masonry / Frame	2	1958	2276	2	5	2
3998	2	Masonry / Frame	1.5	1958	1900	2	5	2
3999	3	Brick	1	1953	1028	1	2	2
4000	3	Frame	2	1901	2496	1	5	2
4001	5	Brick	1	1953	1598	1	3	2
4002	5	Brick	1	1956	1517	1	4	1
4003	5	Stone	1	1947	1479	1	3	1
4004	5	Frame	1	1948	1118	1	4	1
4005	5	Brick	1	1955	1256	1	3	1
4006	5	Brick	1	1956	1207	1	3	1
4007	5	Brick	1	1963	1043	1	3	2
4008	5	Brick	1	1955	1025	1	3	1
4009	5	Brick	1	1950	1009	1	3	1
4010	5	Aluminum / Vinyl	2	1976	2451	2	6	2
4011	5	Aluminum / Vinyl	2	1978	2451	2	6	2
4012	7	Aluminum / Vinyl	1.5	1941	1483	1	4	1
4013	7	Brick	1	1926	1651	1	4	2
4014	8	Aluminum / Vinyl	2	1914	1722	1	3	1
4015	8	Aluminum / Vinyl	1	1922	1582	1	4	2
4016	9	Aluminum / Vinyl	1	1953	990	1	4	1
4017	9	Aluminum / Vinyl	2	2010	2571	1	4	2
4018	10	Aluminum / Vinyl	1.5	1923	1260	1	3	2
4019	10	Brick	1.5	1928	1995	2	3	2

4020	10	Aluminum / Vinyl	1.5	1923	1468	2	5	2
4021	11	Aluminum / Vinyl	1	1951	1266	1	3	1
4022	11	Brick	1	1947	1230	1	4	1
4023	11	Aluminum / Vinyl	1	1940	896	1	3	2
4024	11	Brick	1	1959	1335	1	3	1
4025	11	Brick	1	1962	1297	1	3	1
4026	11	Aluminum / Vinyl	1	1960	1222	1	3	1
4027	11	Aluminum / Vinyl	1	1954	1000	1	3	1
4028	11	Brick	1	1959	983	1	3	2
4029	11	Aluminum / Vinyl	1	1960	931	1	3	1
4030	12	Aluminum / Vinyl	1	1903	942	1	3	1
4031	13	Block	1.5	1940	913	1	2	2
4032	13	Aluminum / Vinyl	1	1927	1482	1	4	2
4033	13	Brick	1	1960	1218	1	3	1
4034	13	Brick	1	1959	988	1	3	2
4035	14	Aluminum / Vinyl	1	1947	1035	1	3	1
4036	14	Fiber-Cement	2	2006	2040	1	3	2
4037	14	Aluminum / Vinyl	2	1937	1144	1	2	1
4038	14	Aluminum / Vinyl	2	1945	990	1	2	1
4039	14	Aluminum / Vinyl	2	1916	2882	2	6	2
4040	14	Aluminum / Vinyl	1	1927	1689	1	4	2
4041	14	Aluminum / Vinyl	1	1928	992	1	3	1
4042	15	Aluminum / Vinyl	1	1920	1681	1	4	2
4043	15	Brick	1.5	1927	2539	1	5	2
4044	1	Frame	1	1928	1624	1	4	1
4045	1	Stone	1	1955	1482	1	2	1
4046	1	Brick	1	1958	1469	1	4	1
4047	1	Aluminum / Vinyl	1	1960	988	1	3	1
4048	1	Aluminum / Vinyl	1	1924	1170	1	3	1
4049	2	Brick	2	1957	2452	2	7	2
4050	2	Aluminum / Vinyl	2	1973	2340	2	6	2
4051	2	Aluminum / Vinyl	2	1973	2340	2	6	2
4052	2	Aluminum / Vinyl	2	1973	2340	2	6	2
4053	2	Frame	2	1954	1536	2	4	2
4054	2	Stone	1	1952	1424	1	2	1
4055	2	Aluminum / Vinyl	1	1954	972	1	2	1
4056	2	Aluminum / Vinyl	1	1955	1414	1	3	1
4057	3	Aluminum / Vinyl	2	1899	3304	1	5	3
4058	3	Aluminum / Vinyl	1	1887	1050	1	3	1
4059	3	Aluminum / Vinyl	2	1900	1520	2	4	2
4060	3	Stucco	2	1913	1545	1	3	1
4061	3	Aluminum / Vinyl	1	1893	1147	1	3	1
4062	5	Aluminum / Vinyl	1	1942	1546	1	4	1

4063	5	Brick	1	1950	1355	1	3	1
4064	5	Aluminum / Vinyl	1	1952	1296	1	3	1
4065	5	Aluminum / Vinyl	1	1949	1119	1	3	2
4066	5	Aluminum / Vinyl	1	1946	1068	1	3	1
4067	5	Masonry / Frame	2	2010	1441	1	3	2
4068	5	Aluminum / Vinyl	2	1953	1422	1	3	1
4069	5	Masonry / Frame	2	1946	1174	1	3	1
4070	5	Aluminum / Vinyl	1	1957	1252	1	3	1
4071	5	Brick	1	1957	1238	1	3	2
4072	5	Brick	1	1956	979	1	2	1
4073	5	Aluminum / Vinyl	1	1928	852	1	3	1
4074	6	Aluminum / Vinyl	2	1927	1321	1	3	1
4075	6	Aluminum / Vinyl	1.5	1891	1827	2	4	2
4076	6	Aluminum / Vinyl	1	1903	1378	1	4	1
4077	6	Stucco	1	1924	1028	1	3	1
4078	7	Aluminum / Vinyl	1.5	1940	1438	1	5	2
4079	7	Brick	2	1927	2902	2	6	2
4080	7	Masonry / Frame	2	1930	2626	2	4	2
4081	7	Frame	2	1961	1879	2	4	2
4082	7	Brick	2	1945	2119	2	4	2
4083	8	Aluminum / Vinyl	1.5	1920	1933	2	4	2
4084	8	Aluminum / Vinyl	1	1924	1598	1	4	1
4085	8	Aluminum / Vinyl	1	1903	1015	1	4	1
4086	9	Aluminum / Vinyl	2	2010	1852	1	3	2
4087	9	Aluminum / Vinyl	2	1968	1636	1	4	1
4088	9	Aluminum / Vinyl	1.5	1936	1574	2	2	2
4089	9	Aluminum / Vinyl	1	2001	1940	1	3	2
4090	9	Frame	1	1957	1009	1	3	1
4091	10	Aluminum / Vinyl	1.5	1948	1238	1	3	2
4092	10	Aluminum / Vinyl	2	1925	1615	1	3	1
4093	10	Aluminum / Vinyl	2	1922	2928	2	6	2
4094	10	Aluminum / Vinyl	2	1927	2247	2	4	2
4095	10	Stucco	2	1952	1670	2	4	2
4096	10	Aluminum / Vinyl	2	1928	2120	2	4	2
4097	10	Aluminum / Vinyl	1	1925	1302	1	4	2
4098	10	Aluminum / Vinyl	1	1929	1252	1	2	1
4099	10	Aluminum / Vinyl	1	1926	961	1	3	1
4100	11	Brick	1	1949	1420	1	3	1
4101	11	Aluminum / Vinyl	1.5	1953	1205	1	3	1
4102	11	Aluminum / Vinyl	1	1959	1124	1	3	2
4103	11	Aluminum / Vinyl	1	1942	868	1	3	1
4104	11	Brick	1.5	1962	2201	2	5	2
4105	11	Brick	1	1963	1216	1	3	1

4106	11	Stone	1	1946	1105	1	2	1
4107	11	Brick	1	1955	1013	1	3	1
4108	11	Brick	1	1954	976	1	3	1
4109	11	Frame	1	1961	935	1	3	1
4110	11	Aluminum / Vinyl	1	1950	802	1	3	1
4111	11	Aluminum / Vinyl	2	1975	2478	2	6	2
4112	12	Aluminum / Vinyl	1	1900	1425	1	4	2
4113	12	Frame	1	1909	1098	1	3	1
4115	13	Brick	1	1946	1133	1	2	1
4116	13	Aluminum / Vinyl	2	1973	2515	1	3	2
4117	13	Aluminum / Vinyl	2	1930	1353	1	3	1
4118	13	Aluminum / Vinyl	2	1969	1228	1	3	1
4119	13	Frame	1	1962	1000	1	3	2
4120	13	Aluminum / Vinyl	1	1959	919	1	3	2
4121	13	Masonry / Frame	2	1972	2390	2	6	2
4122	13	Brick	1.5	1932	1846	1	3	1
4123	13	Brick	2	1932	1594	1	3	1
4124	14	Stone	1	1940	1262	1	2	2
4125	14	Aluminum / Vinyl	2	2010	1354	1	3	2
4126	14	Brick	1.5	1925	2112	2	3	2
4127	14	Aluminum / Vinyl	2	1898	1848	2	5	2
4128	14	Frame	1.5	1916	1758	1	3	2
4129	14	Aluminum / Vinyl	1	1930	1542	1	4	1
4130	14	Aluminum / Vinyl	1	1945	979	1	3	1
4131	14	Aluminum / Vinyl	1	1906	1641	1	3	2
4132	14	Aluminum / Vinyl	1.5	1916	1632	1	3	2
4133	14	Aluminum / Vinyl	1	1920	1095	1	2	1
4134	14	Aluminum / Vinyl	2	1910	1080	1	2	1
4135	14	Aluminum / Vinyl	1	1924	1017	1	3	1
4136	15	Brick	2	1921	2180	1	4	1
4137	15	Frame	2	1895	1642	2	4	2
4138	1	Masonry / Frame	2	1950	1775	2	4	2
4139	1	Aluminum / Vinyl	1	1960	1095	1	3	1
4140	2	Stucco	1.5	1941	1484	1	4	1
4141	2	Aluminum / Vinyl	1	1955	1084	1	4	1
4142	2	Brick	1	1958	1372	1	3	1
4143	2	Aluminum / Vinyl	1	1955	1080	1	3	2
4144	2	Aluminum / Vinyl	1	1958	1040	1	3	1
4145	2	Aluminum / Vinyl	1	1955	864	1	3	1
4146	3	Brick	2	1921	3764	1	3	2
4147	3	Brick	2	1919	3263	1	5	3
4148	3	Brick	1.5	1928	1497	1	3	2
4149	3	Aluminum / Vinyl	2	1924	1307	1	3	1

4150	3	Aluminum / Vinyl	1	1889	1205	1	3	1
4151	5	Frame	1	1952	1664	1	4	1
4152	5	Aluminum / Vinyl	1	1952	1549	1	3	2
4153	5	Aluminum / Vinyl	1	1952	1394	1	3	1
4154	5	Aluminum / Vinyl	1	1955	1110	1	4	1
4155	5	Brick	1	1946	1089	1	3	1
4156	5	Aluminum / Vinyl	1	1951	1076	1	3	2
4157	5	Aluminum / Vinyl	1	1953	1026	1	4	1
4158	5	Aluminum / Vinyl	2	1940	1240	1	3	1
4159	5	Brick	1	1979	1469	1	3	2
4160	5	Brick	1	1957	1198	1	3	1
4161	5	Brick	1	1956	1156	1	3	1
4162	5	Frame	1	1956	1064	1	3	2
4163	5	Aluminum / Vinyl	1	1956	999	1	3	1
4164	5	Aluminum / Vinyl	1	1952	802	1	2	1
4165	6	Frame	2	1892	2260	2	6	2
4166	6	Aluminum / Vinyl	2	1911	2044	2	6	2
4167	7	Brick	1	1948	1047	1	3	1
4168	7	Aluminum / Vinyl	1	1922	2074	1	5	1
4169	7	Brick	1	1925	1891	1	4	2
4170	8	Aluminum / Vinyl	1	1949	1135	1	3	1
4171	8	Aluminum / Vinyl	1	1895	1390	1	3	2
4172	8	Frame	1	1893	1336	1	3	2
4173	8	Aluminum / Vinyl	1	1903	910	1	3	1
4174	9	Brick	1	1942	1207	1	3	2
4175	9	Aluminum / Vinyl	1	1964	1232	1	3	1
4176	10	Aluminum / Vinyl	2	1945	1607	1	4	2
4177	10	Brick	1	1949	1512	1	3	2
4178	10	Block	1	1946	1075	1	3	1
4179	10	Aluminum / Vinyl	1	1948	1070	1	3	1
4180	10	Brick	1	1947	899	1	3	1
4181	10	Aluminum / Vinyl	1.5	1927	1452	2	4	2
4182	10	Brick	1.5	1940	1502	2	3	2
4183	10	Stucco	1	1918	1722	1	5	1
4184	10	Aluminum / Vinyl	1	1954	1100	1	2	1
4185	10	Aluminum / Vinyl	1	1954	982	1	3	1
4186	10	Brick	1.5	1924	1997	1	3	1
4187	10	Aluminum / Vinyl	1.5	1900	1685	1	5	1
4188	11	Brick	1	1960	1321	1	4	2
4189	11	Frame	1.5	1939	1186	1	2	1
4190	11	Aluminum / Vinyl	1	1953	2385	2	5	2
4191	11	Aluminum / Vinyl	1.5	1959	2314	2	6	2
4192	11	Aluminum / Vinyl	2	1954	1728	2	6	2

4193	11	Brick	1	1967	1414	1	3	1
4194	11	Brick	1	1963	1376	1	3	1
4195	11	Brick	1	1956	1189	1	3	1
4196	11	Aluminum / Vinyl	1	1964	1114	1	3	1
4197	11	Aluminum / Vinyl	1	1956	1095	1	3	1
4198	11	Aluminum / Vinyl	1	1957	1063	1	3	1
4199	11	Brick	1	1955	1062	1	2	1
4200	11	Brick	1	1953	1019	1	3	1
4201	11	Aluminum / Vinyl	1	1956	995	1	3	1
4202	11	Brick	1	1954	988	1	3	1
4203	11	Aluminum / Vinyl	1	1954	955	1	3	1
4204	11	Frame	1	1960	932	1	3	1
4205	11	Brick	1	1950	871	1	2	1
4206	12	Aluminum / Vinyl	1	1903	1404	1	3	1
4207	13	Aluminum / Vinyl	1	1947	1348	1	4	2
4208	13	Aluminum / Vinyl	1	1950	1257	1	3	2
4209	13	Aluminum / Vinyl	1	1947	1149	1	3	1
4210	13	Aluminum / Vinyl	2	1951	1564	1	5	2
4211	13	Aluminum / Vinyl	1.5	1927	2199	1	3	2
4212	13	Brick	1	1967	1808	1	3	2
4213	13	Brick	1	1973	1720	1	3	1
4214	13	Brick	1	1969	1490	1	3	1
4215	13	Aluminum / Vinyl	1	1978	1354	1	3	1
4216	13	Aluminum / Vinyl	1	1962	896	1	3	1
4217	13	Aluminum / Vinyl	1	1944	746	1	2	1
4218	13	Aluminum / Vinyl	1	1954	720	1	2	1
4219	14	Aluminum / Vinyl	1	1953	1164	1	3	1
4220	14	Brick	1	1948	992	1	3	1
4221	14	Aluminum / Vinyl	1	1949	910	1	3	1
4222	14	Aluminum / Vinyl	1	1899	814	1	2	1
4223	14	Brick	2	1929	2159	2	4	2
4224	14	Frame	1.5	1920	1614	2	4	2
4225	14	Frame	2	1911	2083	2	3	2
4226	14	Aluminum / Vinyl	1	1927	1669	1	3	2
4227	14	Frame	1	1925	1339	1	3	1
4228	14	Aluminum / Vinyl	1	1926	1160	1	3	1
4229	14	Aluminum / Vinyl	1	1953	922	1	3	1
4230	14	Aluminum / Vinyl	2	1919	1888	1	3	2
4231	14	Block	1.5	1920	1308	1	2	1
4232	14	Aluminum / Vinyl	1	1918	1252	1	3	1
4233	14	Prem Wood	1	1923	1224	1	3	2
4234	14	Frame	1	1921	1030	1	3	1
4235	15	Aluminum / Vinyl	2	2008	1914	1	4	2

4236	15	Aluminum / Vinyl	2	1921	3084	2	6	2
4237	1	Brick	1	1952	1288	1	4	1
4238	1	Aluminum / Vinyl	1	1929	1076	1	2	1
4239	1	Aluminum / Vinyl	1.5	1952	1831	2	4	2
4240	1	Frame	1	1959	1132	1	3	1
4241	1	Aluminum / Vinyl	1	1956	1130	1	3	2
4242	1	Stone	1	1946	967	1	2	1
4243	1	Aluminum / Vinyl	1	1950	672	1	2	1
4244	1	Aluminum / Vinyl	1.5	1925	1324	1	2	1
4245	1	Aluminum / Vinyl	1	1932	1076	1	4	1
4246	2	Aluminum / Vinyl	1	1995	1458	1	3	2
4247	2	Aluminum / Vinyl	1	1953	1300	1	3	2
4248	2	Brick	1.5	1953	1965	2	4	2
4249	2	Frame	1	1950	1332	1	3	1
4250	2	Aluminum / Vinyl	1	1963	1163	1	3	1
4251	2	Brick	1	1955	1131	1	3	1
4252	2	Aluminum / Vinyl	1	1958	1108	1	3	2
4253	2	Aluminum / Vinyl	1	1963	1017	1	3	1
4254	2	Aluminum / Vinyl	1	1955	864	1	3	1
4255	3	Aluminum / Vinyl	2	1890	1632	1	2	2
4256	3	Frame	1	1891	1152	1	3	2
4257	3	Aluminum / Vinyl	1	1885	680	1	2	1
4258	3	Brick	1.5	1929	2211	2	4	2
4259	3	Aluminum / Vinyl	2	1912	2291	2	4	2
4260	3	Frame	2	1907	2132	2	4	2
4261	3	Stucco	1	1916	2280	1	5	2
4262	3	Brick	2	1913	4332	1	6	3
4263	3	Aluminum / Vinyl	2	1904	1986	1	4	1
4264	3	Frame	2	1889	1667	1	3	1
4266	5	Frame	1	1984	1843	1	3	2
4267	5	Brick	1	1950	1449	1	3	2
4268	5	Brick	1	1945	1172	1	3	2
4269	5	Aluminum / Vinyl	1	1952	1120	1	3	1
4270	5	Aluminum / Vinyl	1	1949	1036	1	3	1
4271	5	Aluminum / Vinyl	1	1953	935	1	3	1
4272	5	Masonry / Frame	2	1955	2310	1	4	1
4273	5	Frame	1	1952	1320	1	3	1
4274	5	Brick	1	1950	1206	1	3	2
4275	5	Brick	1	1953	1200	1	3	1
4276	5	Brick	1	1952	1176	1	3	1
4277	5	Brick	1	1953	1125	1	2	1
4278	5	Aluminum / Vinyl	1	1957	1123	1	3	1
4279	5	Aluminum / Vinyl	1	1955	1067	1	3	1

4280	5	Brick	1	1954	1036	1	3	1
4281	6	Aluminum / Vinyl	1.5	1916	1985	1	4	2
4282	6	Frame	1.5	1898	1711	1	0	1
4283	6	Aluminum / Vinyl	1	1900	1152	1	3	1
4284	7	Aluminum / Vinyl	2	1923	2207	2	4	2
4285	7	Aluminum / Vinyl	2	1915	2161	2	4	2
4286	7	Aluminum / Vinyl	1	1893	1150	1	4	1
4287	7	Stone	1	1933	1971	1	4	2
4288	8	Aluminum / Vinyl	1	1922	1326	1	4	1
4289	8	Aluminum / Vinyl	1	1900	1278	1	3	1
4290	8	Aluminum / Vinyl	1	1895	944	1	2	1
4291	9	Frame	1	1940	1173	1	4	1
4292	10	Stone	1	1948	1785	1	4	1
4293	10	Aluminum / Vinyl	1.5	1948	1418	1	4	2
4294	10	Stone	1	1946	1149	1	3	1
4295	10	Stucco	1	1952	1095	1	4	2
4296	10	Aluminum / Vinyl	1	1952	992	1	3	1
4297	10	Aluminum / Vinyl	1	1953	984	1	3	1
4298	10	Brick	2	1924	4137	1	5	3
4299	10	Stucco	2	1915	2883	2	4	2
4300	10	Aluminum / Vinyl	1	1927	1512	1	4	2
4301	10	Aluminum / Vinyl	1	1925	1506	1	2	2
4302	10	Aluminum / Vinyl	1	1953	1068	1	3	1
4303	10	Aluminum / Vinyl	1	1954	1008	1	3	1
4304	10	Aluminum / Vinyl	1	1951	768	1	2	1
4305	10	Aluminum / Vinyl	1	1953	720	1	2	1
4306	10	Stone	1.5	1940	1890	1	3	2
4307	10	Stone	1	1936	1819	1	3	2
4308	11	Aluminum / Vinyl	1	1946	795	1	2	1
4309	11	Aluminum / Vinyl	1.5	1925	2174	2	4	3
4310	11	Brick	2	1958	2528	2	6	2
4311	11	Masonry / Frame	2	1971	2496	2	5	2
4312	11	Aluminum / Vinyl	1.5	1890	2058	2	5	3
4313	11	Brick	1	1929	1865	1	3	1
4314	11	Aluminum / Vinyl	1	1928	1000	1	3	1
4315	11	Brick	1	1955	1600	1	4	1
4316	11	Brick	1	1958	1399	1	3	2
4317	11	Brick	1	1959	1230	1	3	1
4318	11	Aluminum / Vinyl	1	1967	1216	1	3	1
4319	11	Brick	1	1958	1114	1	3	1
4320	11	Aluminum / Vinyl	1	1958	1078	1	3	1
4321	11	Aluminum / Vinyl	1	1958	1072	1	3	1
4322	11	Brick	1	1956	1064	1	3	1

4323	11	Aluminum / Vinyl	1	1957	1026	1	3	1
4324	11	Aluminum / Vinyl	1	1954	989	1	3	1
4325	11	Aluminum / Vinyl	1	1955	984	1	3	1
4326	11	Brick	1	1958	910	1	2	1
4327	11	Aluminum / Vinyl	1	1953	879	1	3	1
4328	11	Aluminum / Vinyl	1	1953	879	1	3	1
4329	11	Aluminum / Vinyl	1.5	1934	1138	1	3	1
4330	12	Frame	1	1900	1088	1	2	1
4331	13	Aluminum / Vinyl	1	1940	1283	1	3	1
4332	13	Brick	1	1962	1325	1	3	1
4333	13	Brick	1	1955	1198	1	3	1
4334	13	Stone	1	1964	1100	1	2	1
4335	13	Aluminum / Vinyl	1	1960	1084	1	3	1
4336	13	Aluminum / Vinyl	1	1946	681	1	2	1
4337	13	Frame	1.5	1929	1184	1	3	1
4338	14	Aluminum / Vinyl	1	1944	1140	1	3	1
4339	14	Frame	2	1920	2096	2	4	2
4340	14	Aluminum / Vinyl	2	1892	2741	2	3	2
4341	14	Aluminum / Vinyl	2	1919	1616	2	4	2
4342	14	Aluminum / Vinyl	1.5	1942	1215	2	3	2
4343	14	Stone	1	1926	1539	1	3	1
4344	14	Aluminum / Vinyl	1.5	1927	1661	1	3	2
4345	14	Frame	1	1915	1430	1	3	1
4346	14	Aluminum / Vinyl	1.5	1926	1133	1	2	1
4347	15	Aluminum / Vinyl	2	1911	1973	2	4	2
4348	1	Brick	1.5	1937	1247	1	3	1
4349	1	Frame	1	1941	1065	1	3	1
4350	1	Brick	2	1954	2026	2	4	2
4351	1	Brick	2	1954	1810	2	6	2
4352	1	Brick	1	1926	1209	1	3	1
4353	2	Stone	1	1952	1378	1	4	1
4354	2	Stone	1	1942	1360	1	3	1
4355	2	Aluminum / Vinyl	1	1956	1124	1	3	2
4356	2	Frame	1	1962	1330	1	3	1
4357	2	Aluminum / Vinyl	1	1956	988	1	3	1
4358	2	Brick	1	1957	964	1	2	2
4359	3	Masonry / Frame	2	1926	2088	1	3	1
4360	3	Aluminum / Vinyl	1.5	1927	1737	2	4	3
4361	3	Aluminum / Vinyl	2	1880	2686	2	6	2
4362	3	Brick	2	1916	4123	1	4	3
4363	3	Brick	2	1910	5028	1	6	4
4364	3	Brick	2	1915	3618	1	3	2
4365	3	Frame	2	1904	2899	1	4	2

4366	3	Frame	2	1922	1344	1	3	1
4367	3	Aluminum / Vinyl	2	1890	3036	3	5	3
4368	5	Stone	1	1946	1902	1	3	2
4369	5	Aluminum / Vinyl	1	1951	1496	1	3	1
4370	5	Stone	1	1952	1456	1	3	2
4371	5	Aluminum / Vinyl	1.5	1949	1367	1	4	1
4372	5	Brick	1	1948	1299	1	3	2
4373	5	Brick	1	1955	1285	1	4	1
4374	5	Aluminum / Vinyl	1	1949	1190	1	4	1
4375	5	Aluminum / Vinyl	1	1937	1143	1	2	1
4376	5	Aluminum / Vinyl	1	1946	1068	1	3	1
4377	5	Aluminum / Vinyl	2	1993	2153	1	3	2
4378	5	Brick	2	1960	2524	2	6	2
4379	5	Masonry / Frame	1.5	1956	1733	2	3	2
4380	5	Aluminum / Vinyl	1.5	1910	2161	2	4	2
4381	5	Brick	1	1965	1862	1	3	1
4382	5	Masonry / Frame	1	1962	1751	1	3	2
4383	5	Brick	1	1962	1345	1	4	1
4384	5	Brick	1	1959	1300	1	3	1
4385	5	Brick	1	1961	1221	1	3	1
4386	5	Aluminum / Vinyl	1	1956	1216	1	3	1
4387	5	Aluminum / Vinyl	1	1976	1211	1	3	1
4388	5	Aluminum / Vinyl	1	1959	1211	1	3	1
4389	5	Aluminum / Vinyl	1	1971	1204	1	4	1
4390	5	Masonry / Frame	1	1956	1114	1	3	2
4391	5	Aluminum / Vinyl	1	1956	1067	1	3	1
4392	5	Aluminum / Vinyl	1	1957	958	1	3	1
4393	5	Aluminum / Vinyl	1	1952	885	1	3	1
4394	6	Frame	2	1924	2206	2	4	2
4395	6	Aluminum / Vinyl	2	1908	2314	2	6	2
4396	6	Aluminum / Vinyl	1	1908	1384	1	3	1
4397	6	Aluminum / Vinyl	1	1904	1294	1	3	1
4398	6	Aluminum / Vinyl	1	1905	1283	1	5	1
4399	7	Stone	1.5	1936	2305	1	4	2
4400	7	Brick	1	1951	1261	1	2	1
4401	7	Brick	1.5	1932	2051	1	4	1
4402	7	Aluminum / Vinyl	1	1896	1488	1	5	2
4403	7	Masonry / Frame	2	1936	1600	1	3	1
4404	8	Aluminum / Vinyl	2	1912	2516	2	5	2
4405	8	Aluminum / Vinyl	1	1922	1270	1	4	2
4406	8	Frame	1	1900	1034	1	3	1
4407	9	Masonry / Frame	2	1957	1679	1	4	1
4408	9	Aluminum / Vinyl	1	1986	1409	1	3	1

4409	9	Aluminum / Vinyl	1	1980	1209	1	3	2
4410	9	Aluminum / Vinyl	1	1971	1200	1	4	1
4411	9	Masonry / Frame	1	1957	1058	1	3	1
4412	9	Brick	1	1957	1051	1	3	1
4413	9	Aluminum / Vinyl	1	1968	1018	1	3	1
4414	9	Frame	1	1964	1006	1	3	1
4415	9	Brick	1	1956	994	1	3	1
4416	9	Brick	1	1958	1858	1	3	1
4417	9	Aluminum / Vinyl	2	1966	2451	2	6	2
4418	10	Brick	1.5	1953	1791	1	4	2
4419	10	Brick	1	1952	1657	1	3	2
4420	10	Brick	1	1952	1134	1	3	2
4421	10	Aluminum / Vinyl	1	1948	1075	1	3	1
4422	10	Aluminum / Vinyl	1	1917	686	1	2	1
4423	10	Aluminum / Vinyl	1.5	1928	2261	2	4	2
4424	10	Frame	1.5	1928	1901	2	4	2
4425	10	Aluminum / Vinyl	1.5	1888	1740	2	4	2
4426	10	Aluminum / Vinyl	1	1922	1686	1	4	2
4427	10	Aluminum / Vinyl	1	1925	1191	1	3	1
4428	10	Brick	1	1954	1125	1	3	2
4429	10	Stone	1	1938	1115	1	2	1
4430	10	Aluminum / Vinyl	1	1922	1066	1	3	1
4431	10	Aluminum / Vinyl	1	1925	1023	1	4	1
4432	10	Brick	1	1926	1884	1	4	2
4433	11	Aluminum / Vinyl	1.5	1954	1779	1	3	2
4434	11	Aluminum / Vinyl	1	1953	1498	1	4	2
4435	11	Aluminum / Vinyl	1	1941	1121	1	3	1
4436	11	Brick	1	1953	986	1	2	1
4437	11	Aluminum / Vinyl	1	1952	953	1	3	1
4438	11	Brick	1	1949	941	1	2	1
4439	11	Aluminum / Vinyl	2	1928	2778	1	5	1
4440	11	Masonry / Frame	2	1940	2174	1	3	1
4441	11	Aluminum / Vinyl	2	1957	2206	2	6	2
4442	11	Masonry / Frame	1.5	1964	2198	2	5	2
4443	11	Brick	1.5	1931	2252	1	3	1
4444	11	Aluminum / Vinyl	1	1955	1481	1	3	2
4445	11	Aluminum / Vinyl	1	1938	1245	1	3	1
4446	11	Aluminum / Vinyl	1	1957	1202	1	3	1
4447	11	Brick	1	1956	1153	1	3	2
4448	11	Brick	1	1957	1152	1	2	1
4449	11	Aluminum / Vinyl	1	1971	1136	1	4	1
4450	11	Frame	1	1964	1128	1	3	1
4451	11	Brick	1	1955	1112	1	3	1

4452	11	Aluminum / Vinyl	1	1953	1076	1	3	1
4453	11	Stone	1	1940	1054	1	2	2
4454	11	Aluminum / Vinyl	1	1957	1040	1	3	1
4455	11	Aluminum / Vinyl	1	1956	1019	1	3	1
4456	11	Frame	1	1956	995	1	3	1
4457	11	Brick	1	1955	924	1	3	1
4458	11	Aluminum / Vinyl	1	1955	876	1	3	1
4459	11	Aluminum / Vinyl	1	1953	770	1	2	2
4460	13	Stone	1	1942	1561	1	4	2
4461	13	Aluminum / Vinyl	1	1950	1253	1	3	2
4462	13	Stone	1.5	1953	2103	2	4	2
4463	13	Aluminum / Vinyl	1	1923	1239	1	3	2
4464	13	Brick	1	1964	1724	1	3	1
4465	13	Aluminum / Vinyl	1	1968	1503	1	4	1
4466	13	Brick	1	1969	1329	1	3	2
4467	13	Brick	1	1956	1150	1	3	2
4468	13	Brick	1	1955	1119	1	3	1
4469	13	Aluminum / Vinyl	1	1961	1059	1	3	1
4470	13	Aluminum / Vinyl	1	1960	956	1	3	1
4471	13	Aluminum / Vinyl	1	1959	919	1	3	1
4472	13	Aluminum / Vinyl	1	1947	771	1	2	1
4473	13	Aluminum / Vinyl	1	1944	759	1	2	1
4474	13	Aluminum / Vinyl	1	1944	730	1	2	1
4475	13	Aluminum / Vinyl	1	1944	730	1	2	1
4476	13	Aluminum / Vinyl	1	1951	696	1	2	1
4477	13	Aluminum / Vinyl	1	1929	782	1	2	1
4478	14	Aluminum / Vinyl	2	2010	1320	1	3	2
4479	14	Aluminum / Vinyl	1	1896	640	1	1	1
4480	14	Aluminum / Vinyl	1	1910	532	1	2	1
4481	14	Masonry / Frame	2	1954	2090	2	4	2
4482	14	Aluminum / Vinyl	1.5	1918	1264	1	3	1
4483	14	Brick	1	1929	1252	1	3	1
4484	14	Brick	1	1955	840	1	2	1
4485	14	Frame	1.5	1918	1560	1	3	2
4486	14	Frame	1	1900	1288	1	3	1
4487	14	Aluminum / Vinyl	1	1916	952	1	3	1
4488	14	Stone	1.5	1936	2141	1	4	1
4489	15	Aluminum / Vinyl	2	1910	2050	2	5	2
4490	1	Aluminum / Vinyl	1	1953	1120	1	4	1
4491	1	Brick	1.5	1936	1099	1	2	1
4492	1	Masonry / Frame	2	1951	1306	1	3	1
4493	1	Aluminum / Vinyl	2	1936	1144	1	3	1
4494	1	Brick	2	1957	2576	2	4	2

4495	1	Brick	1	1954	1185	1	3	1
4496	1	Aluminum / Vinyl	1	1953	1082	1	3	1
4497	2	Aluminum / Vinyl	1	1955	1119	1	3	1
4498	2	Aluminum / Vinyl	1	1952	1019	1	4	2
4499	2	Aluminum / Vinyl	1	1936	945	1	2	1
4500	2	Aluminum / Vinyl	2	1958	1872	2	4	2
4501	2	Aluminum / Vinyl	1	1961	1274	1	3	1
4502	2	Brick	1	1959	1169	1	3	1
4503	2	Aluminum / Vinyl	1	1958	942	1	3	1
4504	2	Aluminum / Vinyl	1	1954	879	1	3	1
4505	3	Brick	2	1916	3543	1	4	2
4506	3	Aluminum / Vinyl	1	1920	1918	1	4	2
4507	3	Aluminum / Vinyl	1	1919	1635	1	3	2
4508	3	Brick	2	1928	3768	1	5	3
4509	3	Brick	2	1915	3666	1	5	3
4510	3	Stucco	2	1915	3507	1	4	2
4511	3	Frame	2	1902	2781	1	4	2
4512	3	Frame	1	1908	1512	1	3	1
4513	3	Frame	1	1908	1483	1	3	1
4514	5	Aluminum / Vinyl	1.5	1998	1819	1	3	2
4515	5	Aluminum / Vinyl	1	1952	1244	1	3	1
4516	5	Aluminum / Vinyl	1	1956	1154	1	4	1
4517	5	Frame	1.5	1939	1139	1	2	1
4518	5	Aluminum / Vinyl	1	1952	1021	1	4	1
4519	5	Aluminum / Vinyl	1.5	1949	999	1	3	1
4520	5	Masonry / Frame	2	1952	1528	1	3	1
4521	5	Aluminum / Vinyl	1.5	1922	1817	2	5	2
4522	5	Stone	1	1952	1604	1	3	1
4523	5	Brick	1	1954	1232	1	2	1
4524	5	Brick	1	1949	1216	1	2	1
4525	5	Frame	1	1971	1212	1	4	1
4526	5	Aluminum / Vinyl	1	1956	1132	1	3	2
4527	5	Brick	1	1956	1082	1	3	1
4528	5	Frame	1	1960	1079	1	3	1
4529	5	Aluminum / Vinyl	1	1956	1074	1	3	1
4530	5	Aluminum / Vinyl	1	1952	1064	1	2	1
4531	5	Aluminum / Vinyl	1	1950	715	1	2	1
4532	6	Aluminum / Vinyl	2	1905	3485	2	7	3
4533	6	Aluminum / Vinyl	1.5	1908	1538	2	3	2
4534	6	Aluminum / Vinyl	1	1920	826	1	2	1
4535	7	Aluminum / Vinyl	1	1953	1227	1	4	1
4536	7	Brick	1.5	1931	2008	2	3	2
4537	7	Aluminum / Vinyl	1	1921	1644	1	4	1

4538	7	Aluminum / Vinyl	1	1891	920	1	2	1
4539	8	Aluminum / Vinyl	1	1885	1208	1	3	1
4540	8	Aluminum / Vinyl	1.5	1929	2156	2	4	2
4541	8	Aluminum / Vinyl	1.5	1921	2049	2	4	2
4542	8	Aluminum / Vinyl	2	1913	2558	2	6	2
4543	8	Aluminum / Vinyl	1.5	1900	1497	1	4	1
4544	8	Aluminum / Vinyl	1.5	1907	1470	1	4	1
4545	8	Aluminum / Vinyl	1.5	1900	1255	1	3	1
4546	8	Aluminum / Vinyl	1	1903	1254	1	4	2
4547	9	Brick	1	1939	944	1	3	2
4548	9	Aluminum / Vinyl	1	1957	1332	1	3	1
4549	9	Aluminum / Vinyl	1	1958	1039	1	3	1
4550	9	Aluminum / Vinyl	1	1965	1033	1	3	1
4551	10	Brick	1	1937	1875	1	4	1
4552	10	Stone	1	1935	1490	1	2	1
4553	10	Brick	1	1949	1204	1	3	1
4554	10	Brick	1	1948	1129	1	3	1
4555	10	Aluminum / Vinyl	1	1952	1100	1	3	1
4556	10	Aluminum / Vinyl	1	1948	1082	1	3	1
4557	10	Aluminum / Vinyl	1	1947	1063	1	2	1
4558	10	Stucco	2	1923	2304	1	4	1
4559	10	Aluminum / Vinyl	2	2004	2227	1	4	2
4560	10	Aluminum / Vinyl	2	1946	1342	1	3	1
4561	10	Aluminum / Vinyl	2	1917	2678	2	6	2
4562	10	Brick	1.5	1941	1991	2	3	2
4563	10	Aluminum / Vinyl	1.5	1917	2011	1	3	1
4564	10	Brick	1	1933	1345	1	3	1
4565	10	Aluminum / Vinyl	1	1917	964	1	3	1
4566	10	Aluminum / Vinyl	1	1917	964	1	3	1
4567	10	Brick	1	1954	1380	1	3	2
4568	10	Aluminum / Vinyl	1	1954	864	1	3	1
4569	10	Aluminum / Vinyl	1	1918	1340	1	3	1
4570	10	Aluminum / Vinyl	1	1922	1208	1	3	2
4571	10	Brick	1.5	1928	2297	1	3	1
4572	11	Stone	1	1942	2928	1	3	2
4573	11	Brick	1	1952	1705	1	4	1
4574	11	Stone	1	1950	1279	1	3	1
4575	11	Aluminum / Vinyl	1	1946	958	1	3	1
4576	11	Brick	1.5	1959	2184	2	4	2
4577	11	Aluminum / Vinyl	1.5	1941	1841	2	3	2
4578	11	Stone	1.5	1942	1808	2	3	2
4579	11	Aluminum / Vinyl	1	1959	1235	1	3	1
4580	11	Aluminum / Vinyl	1	1970	1172	1	4	1

4581	11	Aluminum / Vinyl	1	1962	1161	1	3	1
4582	11	Brick	1	1959	1053	1	3	1
4583	11	Aluminum / Vinyl	1	1955	1002	1	3	2
4584	11	Aluminum / Vinyl	1	1954	984	1	3	1
4585	11	Aluminum / Vinyl	1	1954	984	1	3	1
4586	11	Aluminum / Vinyl	1	1960	937	1	3	1
4587	11	Brick	1	1954	918	1	3	1
4588	11	Aluminum / Vinyl	1	1953	899	1	3	1
4589	12	Aluminum / Vinyl	2	2009	1672	1	3	2
4590	12	Aluminum / Vinyl	1	1885	1295	1	4	1
4591	12	Aluminum / Vinyl	1	1880	882	1	2	1
4592	12	Frame	1	1903	1530	2	4	2
4593	13	Stone	1.5	1937	1433	1	2	1
4594	13	Brick	1	1951	1348	1	3	1
4595	13	Brick	1	1947	1163	1	3	1
4596	13	Aluminum / Vinyl	1	1954	1137	1	3	1
4597	13	Brick	1.5	1951	2463	2	5	2
4598	13	Brick	1	1964	1167	1	3	1
4599	13	Brick	1	1956	961	1	2	1
4600	14	Aluminum / Vinyl	1	1952	784	1	2	1
4601	14	Frame	2	1913	1536	1	4	2
4602	14	Aluminum / Vinyl	2	1945	1360	1	2	1
4603	14	Masonry / Frame	2	1984	2660	2	6	2
4604	14	Masonry / Frame	2	1940	2378	2	5	2
4605	14	Aluminum / Vinyl	1	1929	1450	1	3	1
4606	14	Aluminum / Vinyl	1	1928	992	1	2	1
4607	14	Aluminum / Vinyl	1	1955	864	1	3	1
4608	14	Aluminum / Vinyl	2	1908	2520	1	4	2
4609	14	Aluminum / Vinyl	1.5	1883	2024	1	4	2
4610	14	Aluminum / Vinyl	1.5	1910	1745	1	4	2
4611	14	Frame	2	1910	1520	1	4	1
4612	14	Aluminum / Vinyl	1	1910	1512	1	3	1
4613	14	Frame	2	1925	1320	1	3	1
4614	15	Aluminum / Vinyl	2	2009	1604	1	3	2
4615	1	Brick	1	1951	1308	1	3	1
4616	1	Brick	1	1951	1143	1	3	1
4617	1	Aluminum / Vinyl	1	1981	1007	1	3	1
4618	1	Aluminum / Vinyl	2	1928	1359	1	3	2
4619	2	Brick	1.5	1952	1715	1	4	1
4620	2	Brick	1	1951	1235	1	3	2
4621	2	Brick	1	1960	1044	1	2	1
4622	2	Aluminum / Vinyl	1	1956	1025	1	3	1
4623	2	Aluminum / Vinyl	1	1955	1008	1	3	1

4624	2	Aluminum / Vinyl	1	1955	756	1	2	1
4625	3	Frame	2	1902	4321	1	6	2
4626	3	Brick	1.5	1928	2201	2	4	2
4627	3	Frame	1.5	1905	1929	2	5	2
4628	3	Aluminum / Vinyl	1	1920	1490	1	3	1
4629	3	Brick	2	1904	4760	1	3	2
4630	3	Masonry / Frame	2	1910	3607	1	5	3
4631	3	Brick	2	1926	3434	1	4	3
4632	3	Stucco	1	1912	2823	1	3	3
4633	3	Aluminum / Vinyl	2	1907	2568	1	6	1
4634	3	Masonry / Frame	2	1917	2063	1	1	1
4635	3	Frame	2	1904	1812	1	3	2
4637	3	Frame	1.5	1910	1564	1	3	1
4638	3	Brick	2	1907	4417	1	6	3
4639	4	Frame	2	1909	2280	2	4	2
4640	5	Brick	1	1952	1659	1	3	2
4641	5	Brick	1	1952	1476	1	3	1
4642	5	Brick	1.5	1937	1392	1	3	1
4643	5	Brick	1	1947	1332	1	4	2
4644	5	Brick	1	1949	1298	1	3	1
4645	5	Brick	1	1948	1233	1	3	1
4646	5	Aluminum / Vinyl	1	1953	1064	1	3	2
4647	5	Aluminum / Vinyl	1.5	1950	1060	1	3	1
4648	5	Aluminum / Vinyl	1	1952	1018	1	4	1
4649	5	Aluminum / Vinyl	2	1969	1852	1	4	2
4650	5	Masonry / Frame	2	1952	2384	2	4	2
4651	5	Prem Wood	1	1993	2908	1	4	3
4652	5	Aluminum / Vinyl	1	1969	1573	1	3	1
4653	5	Aluminum / Vinyl	1	1972	1424	1	3	2
4654	5	Stone	1	1955	1294	1	3	1
4655	5	Aluminum / Vinyl	1	1955	1028	1	3	1
4656	5	Brick	1	1955	1016	1	3	1
4657	5	Aluminum / Vinyl	1	1950	995	1	2	1
4658	6	Aluminum / Vinyl	2	1893	2206	2	4	2
4659	6	Frame	1	1869	1512	2	4	3
4660	6	Frame	1.5	1900	1651	1	1	0
4661	6	Aluminum / Vinyl	2	1922	1306	1	3	1
4662	6	Aluminum / Vinyl	1	1893	1012	1	3	2
4663	7	Stone	1	1935	1962	1	4	2
4664	7	Brick	1.5	1953	1358	1	4	1
4665	7	Frame	2	1924	2586	2	6	2
4666	7	Brick	1	1927	1743	1	3	1
4667	7	Aluminum / Vinyl	1	1926	1422	1	3	1

4668	7	Brick	1	1950	1163	1	3	2
4669	8	Aluminum / Vinyl	1	1892	886	1	3	1
4670	8	Aluminum / Vinyl	1.5	1922	1702	2	3	2
4671	8	Brick	1	1927	1485	1	4	1
4672	8	Frame	1	1925	980	1	3	1
4673	8	Aluminum / Vinyl	1.5	1920	1255	1	3	1
4674	8	Brick	1	1929	1390	1	3	1
4675	9	Frame	1	1966	2092	1	3	1
4676	9	Aluminum / Vinyl	1	1952	1080	1	4	1
4677	9	Brick	2	1956	1949	2	4	2
4678	9	Aluminum / Vinyl	1	1966	1333	1	4	1
4679	9	Aluminum / Vinyl	1	1968	1175	1	3	1
4680	9	Aluminum / Vinyl	1	1958	1122	1	3	1
4681	9	Brick	1	1957	1103	1	3	1
4682	9	Brick	1	1957	973	1	3	1
4683	10	Aluminum / Vinyl	2	1914	2247	1	4	2
4684	10	Brick	2	1927	2004	1	3	1
4685	10	Brick	2	1937	1734	1	3	1
4686	10	Aluminum / Vinyl	2	1941	1484	1	3	1
4687	10	Brick	2	1921	2720	2	6	2
4688	10	Aluminum / Vinyl	2	1924	2491	2	6	2
4689	10	Aluminum / Vinyl	2	1924	2212	2	4	2
4690	10	Stucco	1	1918	1637	1	4	2
4691	10	Aluminum / Vinyl	1	1929	1607	1	3	2
4692	10	Aluminum / Vinyl	1	1928	1586	1	4	2
4693	10	Brick	1	1956	1284	1	3	1
4694	10	Brick	1	1956	1269	1	3	2
4695	10	Aluminum / Vinyl	1	1955	1120	1	2	2
4696	10	Aluminum / Vinyl	1	1953	768	1	2	1
4697	11	Stone	1.5	1946	1855	1	4	1
4698	11	Stucco	1	1946	1371	1	3	1
4699	11	Aluminum / Vinyl	1	1953	1329	1	4	1
4700	11	Aluminum / Vinyl	1	1952	1296	1	3	1
4701	11	Stone	1	1941	1198	1	3	1
4702	11	Brick	1	1941	1183	1	2	1
4703	11	Aluminum / Vinyl	1	1952	1096	1	3	2
4704	11	Aluminum / Vinyl	1	1954	789	1	2	1
4705	11	Aluminum / Vinyl	2	1974	1669	1	3	2
4706	11	Masonry / Frame	2	1952	2310	2	4	2
4707	11	Brick	1	1958	1418	1	3	2
4708	11	Aluminum / Vinyl	1	1968	1413	1	4	1
4709	11	Aluminum / Vinyl	1	1979	1247	1	3	1
4710	11	Masonry / Frame	1	1952	1216	1	3	1

4711	11	Aluminum / Vinyl	1	1972	1152	1	3	1
4712	11	Brick	1	1956	1142	1	3	1
4713	11	Brick	1	1957	1114	1	3	1
4714	11	Aluminum / Vinyl	1	1955	1055	1	3	1
4715	11	Aluminum / Vinyl	1	1957	1035	1	3	1
4716	11	Brick	1	1950	975	1	3	1
4717	11	Brick	1	1960	965	1	3	1
4718	11	Brick	1	1954	959	1	3	2
4719	11	Aluminum / Vinyl	1	1950	826	1	2	1
4720	11	Aluminum / Vinyl	1	1949	726	1	2	1
4721	13	Brick	1	1949	1353	1	3	2
4722	13	Aluminum / Vinyl	1	1971	1134	1	4	1
4723	13	Aluminum / Vinyl	2	1924	2440	2	5	2
4724	13	Aluminum / Vinyl	2	1954	1536	2	4	2
4725	13	Aluminum / Vinyl	1	1979	1878	1	3	2
4726	13	Aluminum / Vinyl	1	1977	1744	1	3	3
4727	13	Aluminum / Vinyl	1	1972	1407	1	3	1
4728	13	Brick	1	1973	1325	1	3	1
4729	13	Brick	1	1964	1164	1	3	1
4730	13	Frame	1	1963	1012	1	3	1
4731	13	Aluminum / Vinyl	1	1950	832	1	2	1
4732	14	Aluminum / Vinyl	1	1953	1233	1	3	1
4733	14	Aluminum / Vinyl	1	1953	1054	1	3	1
4734	14	Aluminum / Vinyl	2	1947	1378	1	3	1
4735	14	Aluminum / Vinyl	1	1900	686	1	2	2
4736	14	Brick	1.5	1953	2073	2	4	2
4737	14	Aluminum / Vinyl	2	1943	1620	2	4	2
4738	14	Aluminum / Vinyl	2	1954	1490	2	4	2
4739	14	Aluminum / Vinyl	2	1918	1752	2	4	2
4740	14	Aluminum / Vinyl	1.5	1915	1735	1	4	2
4741	14	Aluminum / Vinyl	1	1925	1530	1	4	2
4742	14	Aluminum / Vinyl	1	1924	906	1	3	1
4743	14	Brick	1	1952	1099	1	2	1
4744	14	Aluminum / Vinyl	1	1954	891	1	3	1
4745	14	Aluminum / Vinyl	1	1953	704	1	2	1
4746	14	Brick	1	1923	1648	1	4	1
4747	14	Aluminum / Vinyl	1.5	1897	1591	1	3	2
4748	14	Frame	1.5	1916	1424	1	4	1
4749	14	Aluminum / Vinyl	1	1905	1197	1	4	3
4750	14	Aluminum / Vinyl	1	1924	1102	1	4	1
4751	15	Aluminum / Vinyl	1	1987	1362	1	3	2
4752	15	Stucco	1.5	1919	2262	1	4	1
4753	1	Stone	1	1946	1416	1	4	1

4754	1	Prem Wood	2	1931	2113	1	3	2
4755	1	Frame	1	1958	1236	1	3	1
4756	1	Brick	1	1955	1197	1	3	1
4757	1	Aluminum / Vinyl	1	1967	1186	1	3	1
4758	1	Aluminum / Vinyl	1	1954	672	1	2	1
4759	2	Frame	1	1972	1434	1	4	1
4760	2	Aluminum / Vinyl	1	1955	1302	1	4	1
4761	2	Masonry / Frame	2	1942	1310	1	3	1
4762	2	Aluminum / Vinyl	1	1971	1164	1	4	1
4763	3	Frame	2	1900	3323	1	4	3
4764	3	Stucco	2	1916	3307	1	4	3
4765	3	Frame	2	1894	1274	1	3	1
4766	3	Brick	2	1931	2552	2	4	2
4767	3	Aluminum / Vinyl	2	1915	2402	2	6	2
4768	3	Brick	2	1921	3082	1	4	2
4769	3	Fiber-Cement	2	1890	2350	1	4	2
4770	3	Frame	2	1891	2280	1	4	2
4771	3	Brick	1	1925	2253	1	3	2
4772	3	Aluminum / Vinyl	2	1912	1608	1	3	1
4773	4	Aluminum / Vinyl	2	2001	2236	1	3	3
4774	4	Aluminum / Vinyl	1	1888	574	1	2	1
4775	4	Frame	2	1903	2396	1	5	1
4776	4	Aluminum / Vinyl	2	1885	2208	1	4	2
4777	5	Brick	1	1952	1502	1	3	2
4778	5	Brick	1.5	1957	1421	1	3	2
4779	5	Stone	1.5	1953	1349	1	3	1
4780	5	Brick	1	1948	1341	1	3	1
4781	5	Aluminum / Vinyl	1	1946	1234	1	3	1
4782	5	Brick	1	1952	1159	1	3	1
4783	5	Aluminum / Vinyl	1	1950	1113	1	3	2
4784	5	Aluminum / Vinyl	1	1950	1087	1	3	2
4785	5	Brick	1	1952	950	1	2	1
4786	5	Stucco	1	1953	784	1	2	1
4787	5	Aluminum / Vinyl	2	2011	3166	1	5	3
4788	5	Brick	2	1956	2426	2	4	2
4789	5	Masonry / Frame	2	1947	2072	2	4	2
4790	5	Aluminum / Vinyl	1	1934	1476	1	4	2
4791	5	Brick	1	1961	1300	1	3	2
4792	5	Stone	1	1950	1254	1	2	1
4793	5	Brick	1	1959	1197	1	3	1
4794	5	Aluminum / Vinyl	1	1964	1164	1	3	1
4795	5	Aluminum / Vinyl	1	1958	970	1	3	1
4796	5	Aluminum / Vinyl	1	1954	962	1	3	1

4797	5	Aluminum / Vinyl	1	1955	948	1	3	1
4798	5	Aluminum / Vinyl	1	1950	720	1	2	1
4799	6	Aluminum / Vinyl	2	1905	2148	2	4	2
4800	6	Aluminum / Vinyl	1	1926	1275	1	3	2
4801	6	Brick	2	1890	3026	1	5	2
4802	7	Brick	1	1946	1073	1	3	1
4803	7	Aluminum / Vinyl	1.5	1923	2042	2	4	2
4804	7	Aluminum / Vinyl	1.5	1892	1661	2	5	2
4805	7	Brick	1.5	1925	2063	1	3	1
4806	8	Aluminum / Vinyl	2	2009	1870	1	3	2
4807	8	Frame	1	1890	1035	1	3	1
4808	8	Aluminum / Vinyl	2	1950	2260	2	4	2
4809	8	Aluminum / Vinyl	1.5	1908	1760	1	5	2
4810	8	Aluminum / Vinyl	1	1910	1639	1	3	1
4811	8	Aluminum / Vinyl	1	1903	1545	1	4	1
4813	9	Aluminum / Vinyl	1	1974	1430	1	4	2
4814	9	Aluminum / Vinyl	1	1967	1303	1	3	1
4815	9	Aluminum / Vinyl	1	1966	1227	1	3	1
4816	10	Stone	1	1945	1391	1	4	2
4817	10	Aluminum / Vinyl	1	1947	1107	1	3	2
4818	10	Brick	1.5	1920	2535	2	4	2
4819	10	Brick	1.5	1942	2240	2	3	2
4820	10	Aluminum / Vinyl	1	1929	1390	2	3	2
4821	10	Brick	2	1958	5394	2	6	4
4822	10	Masonry / Frame	2	1952	1900	2	4	2
4823	10	Aluminum / Vinyl	1.5	1971	1889	2	5	2
4824	10	Aluminum / Vinyl	1.5	1924	1307	2	2	2
4825	10	Aluminum / Vinyl	1	1924	982	1	2	1
4826	10	Brick	1	1952	960	1	2	1
4827	10	Aluminum / Vinyl	1	1951	874	1	2	1
4828	10	Aluminum / Vinyl	1	1952	828	1	2	1
4829	10	Aluminum / Vinyl	1	1926	1078	1	3	2
4830	10	Aluminum / Vinyl	1	1924	1019	1	2	1
4831	11	Brick	1	1953	1333	1	3	1
4832	11	Aluminum / Vinyl	1	1958	1054	1	3	1
4833	11	Aluminum / Vinyl	2	1959	2162	2	5	2
4834	11	Aluminum / Vinyl	2	1959	1946	2	6	2
4835	11	Aluminum / Vinyl	2	1954	1824	2	6	2
4836	11	Brick	1.5	1923	2365	1	4	2
4837	11	Frame	1	1964	1553	1	3	1
4838	11	Brick	1	1960	1438	1	3	2
4839	11	Aluminum / Vinyl	1	1964	1240	1	3	1
4840	11	Frame	1	1959	1203	1	3	1

4841	11	Aluminum / Vinyl	1	1957	1078	1	3	1
4842	11	Aluminum / Vinyl	1	1960	1063	1	3	1
4843	11	Brick	1	1955	946	1	3	1
4844	11	Aluminum / Vinyl	1	1952	846	1	2	1
4845	11	Aluminum / Vinyl	1	1941	800	1	2	1
4846	11	Aluminum / Vinyl	1	1954	768	1	2	1
4847	11	Aluminum / Vinyl	2	1971	3102	2	6	2
4848	12	Aluminum / Vinyl	1	1903	1808	1	5	2
4849	13	Aluminum / Vinyl	1.5	1940	1360	1	4	2
4850	13	Stone	1	1940	1159	1	3	2
4851	13	Brick	1	1945	1075	1	3	1
4852	13	Aluminum / Vinyl	2	1948	1378	1	3	1
4853	13	Aluminum / Vinyl	1	1971	1196	1	2	1
4854	13	Aluminum / Vinyl	1	1945	826	1	2	1
4855	13	Aluminum / Vinyl	1.5	1921	1314	1	2	1
4856	13	Brick	2	1956	2435	3	5	3
4857	14	Aluminum / Vinyl	1	1940	1300	1	3	1
4858	14	Aluminum / Vinyl	1	1948	1044	1	3	1
4859	14	Brick	2	1915	2716	1	4	1
4860	14	Aluminum / Vinyl	2	2009	2052	1	3	1
4861	14	Aluminum / Vinyl	2	2010	1354	1	3	2
4862	14	Brick	2	1929	2850	2	6	2
4863	14	Brick	2	1929	2850	2	6	2
4864	14	Brick	1.5	1926	2664	2	5	3
4865	14	Brick	1.5	1929	2284	2	5	2
4866	14	Stucco	1.5	1923	1842	2	3	2
4867	14	Aluminum / Vinyl	1.5	1885	2410	2	6	2
4868	14	Brick	1	1926	2134	1	3	2
4869	14	Aluminum / Vinyl	1	1925	1145	1	3	1
4870	14	Aluminum / Vinyl	1	1952	877	1	2	1
4871	14	Aluminum / Vinyl	2	1903	1728	1	4	1
4872	14	Fiber-Cement	2	1890	1604	1	3	2
4873	14	Frame	2	1900	1520	1	3	2
4874	14	Aluminum / Vinyl	1	1916	1458	1	3	1
4875	14	Frame	1	1895	1130	1	3	1
4876	14	Aluminum / Vinyl	1	1900	1088	1	2	1
4877	15	Brick	1	1924	2175	1	3	2
4878	15	Aluminum / Vinyl	1	2001	1298	1	3	2
4879	15	Brick	1.5	1925	5383	1	4	>4
4880	1	Stone	1	1950	1627	1	3	1
4881	1	Brick	2	1958	2156	1	5	3
4882	1	Aluminum / Vinyl	2	1966	1292	1	3	1
4883	1	Aluminum / Vinyl	1	1954	1314	1	3	1

4884	1	Aluminum / Vinyl	1	1955	1082	1	3	1
4885	1	Frame	1	1925	970	1	3	1
4886	2	Brick	1	1957	1698	2	3	2
4887	2	Aluminum / Vinyl	1	1961	1398	1	3	1
4888	2	Brick	1	1956	1237	1	3	1
4889	2	Aluminum / Vinyl	1	1959	1108	1	3	1
4890	2	Brick	1	1962	1022	1	3	1
4891	2	Frame	1	1955	963	1	3	1
4892	2	Frame	1	1955	875	1	3	1
4893	2	Aluminum / Vinyl	1	1947	858	1	2	1
4894	3	Aluminum / Vinyl	2	1870	2016	1	3	2
4895	3	Brick	2	1905	7014	2	6	4
4896	3	Frame	2	1901	2960	1	5	2
4897	3	Brick	1.5	1927	2669	1	4	2
4898	3	Masonry / Frame	2	1916	2426	1	4	2
4899	3	Aluminum / Vinyl	1	1890	1113	1	3	1
4900	4	Brick	2	1916	3607	2	6	2
4901	4	Aluminum / Vinyl	2	1899	2329	1	3	1
4902	5	Aluminum / Vinyl	1	1956	1696	1	4	2
4903	5	Stone	1	1947	1410	1	3	1
4904	5	Brick	1	1954	1359	1	4	1
4905	5	Stone	1	1948	1335	1	3	1
4906	5	Aluminum / Vinyl	1	1955	1293	1	4	2
4907	5	Brick	1	1947	1279	1	2	1
4908	5	Aluminum / Vinyl	1	1952	1176	1	3	1
4909	5	Aluminum / Vinyl	1	1954	1001	1	4	1
4910	5	Aluminum / Vinyl	1	1953	993	1	4	1
4911	5	Aluminum / Vinyl	2	1940	1248	1	3	1
4912	5	Aluminum / Vinyl	2	1942	1084	1	2	1
4913	5	Brick	2	1952	2846	2	6	2
4914	5	Aluminum / Vinyl	1	2010	2222	1	3	2
4915	5	Brick	1	1980	1573	1	2	2
4916	5	Stone	1	1963	1385	1	3	1
4917	5	Brick	1	1961	1309	1	3	1
4918	5	Aluminum / Vinyl	1	1956	1132	1	3	1
4919	5	Brick	1	1955	1101	1	2	2
4920	5	Aluminum / Vinyl	1	1956	1064	1	3	1
4921	5	Aluminum / Vinyl	1	1950	977	1	3	1
4922	5	Aluminum / Vinyl	1	1954	861	1	3	1
4923	5	Aluminum / Vinyl	1	1949	828	1	2	1
4924	5	Aluminum / Vinyl	1	1948	704	1	2	1
4925	6	Fiber-Cement	2	2004	1824	1	3	2
4926	6	Aluminum / Vinyl	2	1922	1898	2	4	2

4927	6	Aluminum / Vinyl	1.5	1908	1302	1	3	1
4928	7	Aluminum / Vinyl	1	1941	1480	1	4	1
4929	7	Stone	1	1938	1454	1	4	1
4930	7	Brick	1	1955	1180	1	3	1
4931	7	Aluminum / Vinyl	1	1929	924	1	3	1
4932	8	Aluminum / Vinyl	1	1925	960	1	2	1
4933	8	Frame	1	1890	1707	1	3	1
4934	8	Aluminum / Vinyl	2	1915	1702	1	4	1
4935	8	Aluminum / Vinyl	1	1913	1403	1	4	2
4936	8	Aluminum / Vinyl	1	1916	1096	1	3	1
4937	9	Frame	1	1960	936	1	3	1
4938	9	Masonry / Frame	2	1971	3500	3	>8	3
4939	10	Aluminum / Vinyl	1.5	1948	1380	1	3	1
4940	10	Frame	1	1952	1291	1	3	1
4941	10	Aluminum / Vinyl	1	1949	1056	1	3	1
4942	10	Stucco	2	1913	2942	2	6	2
4943	10	Brick	1	1926	1707	1	5	2
4944	10	Brick	1	1927	1368	1	3	1
4945	10	Aluminum / Vinyl	1	1953	1098	1	3	2
4946	10	Aluminum / Vinyl	1	1922	912	1	3	1
4947	11	Stucco	1	1941	1271	1	3	1
4948	11	Brick	1	1950	1150	1	3	1
4949	11	Stone	1	1952	1056	1	2	1
4950	11	Aluminum / Vinyl	1	1942	848	1	3	1
4951	11	Aluminum / Vinyl	2	1949	1400	1	3	1
4952	11	Aluminum / Vinyl	2	1954	1554	2	4	2
4953	11	Aluminum / Vinyl	1	1963	1342	1	3	1
4954	11	Brick	1	1956	1288	1	3	1
4955	11	Brick	1	1966	1170	1	3	2
4956	11	Brick	1	1958	1121	1	3	1
4957	11	Frame	1	1956	1090	1	3	1
4958	11	Aluminum / Vinyl	1	1957	1011	1	3	1
4959	11	Aluminum / Vinyl	1	1956	958	1	3	1
4960	11	Brick	1	1955	935	1	3	1
4961	12	Aluminum / Vinyl	1.5	1923	2055	2	4	2
4962	13	Stone	1	1946	1280	1	3	1
4963	13	Masonry / Frame	2	1981	1785	1	3	1
4964	13	Brick	1.5	1930	2102	2	3	2
4965	13	Aluminum / Vinyl	1	1966	1371	1	3	1
4966	13	Brick	1	1963	1333	1	3	2
4967	13	Brick	1	1959	1312	1	3	1
4968	13	Aluminum / Vinyl	1	1969	1302	1	3	2
4969	13	Brick	1	1956	1150	1	3	1

4970	13	Aluminum / Vinyl	1	1958	982	1	3	1
4971	13	Aluminum / Vinyl	1	1960	950	1	3	1
4972	13	Aluminum / Vinyl	1	1940	702	1	2	1
4973	14	Aluminum / Vinyl	1.5	1926	2073	2	4	2
4974	14	Brick	1.5	1958	2059	2	5	2
4975	14	Aluminum / Vinyl	2	1974	2012	2	6	2
4976	14	Frame	2	1913	2544	2	4	2
4977	14	Frame	1	1912	1973	2	5	2
4978	14	Aluminum / Vinyl	1	1923	1757	1	3	2
4979	14	Aluminum / Vinyl	1	1929	1175	1	4	1
4980	14	Aluminum / Vinyl	1	1992	914	1	3	1
4981	14	Aluminum / Vinyl	2	1909	2141	1	3	2
4982	14	Aluminum / Vinyl	2	1903	1440	1	4	1
4983	14	Aluminum / Vinyl	1.5	1913	1424	1	4	1
4984	14	Frame	1.5	1900	1339	1	3	1
4985	14	Aluminum / Vinyl	1	1917	1176	1	4	1
4986	14	Frame	1	1899	1000	1	2	1
4987	15	Aluminum / Vinyl	2	1912	2856	2	6	2
4988	15	Aluminum / Vinyl	2	1911	2145	2	4	2
4989	15	Aluminum / Vinyl	2	1911	2144	2	4	2
4990	15	Aluminum / Vinyl	1.5	1883	1510	2	4	2
4991	1	Aluminum / Vinyl	2	1969	2502	2	6	2
4992	1	Aluminum / Vinyl	1	1962	1068	1	3	2
4993	2	Aluminum / Vinyl	1	1953	1518	1	4	2
4994	2	Aluminum / Vinyl	1	1953	986	1	3	1
4995	2	Aluminum / Vinyl	2	2003	3079	1	5	3
4996	2	Brick	1	1952	1290	1	2	1
4997	2	Brick	1	1951	1008	1	3	1
4998	2	Brick	1	1956	936	1	3	1
4999	2	Brick	1	1956	936	1	3	1
5000	3	Aluminum / Vinyl	1	1895	1778	2	4	1
5001	3	Masonry / Frame	2	1910	3135	1	4	2
5002	3	Aluminum / Vinyl	2	1899	1556	1	4	1
5003	3	Aluminum / Vinyl	2	1895	1516	1	2	2
5004	3	Frame	1.5	1900	1317	1	3	1
5005	5	Aluminum / Vinyl	1	1955	1316	1	4	1
5006	5	Brick	1	1952	1259	1	2	1
5007	5	Aluminum / Vinyl	1	1949	1231	1	4	1
5008	5	Aluminum / Vinyl	1	1940	1150	1	3	1
5009	5	Masonry / Frame	2	1952	3312	1	5	2
5010	5	Aluminum / Vinyl	2	1953	1411	1	3	1
5011	5	Aluminum / Vinyl	2	1954	1976	2	4	2
5012	5	Brick	1	1955	2262	1	2	1

5013	5	Brick	1	1958	1492	1	3	1
5014	5	Brick	1	1949	1453	1	3	2
5015	5	Brick	1	1955	1205	1	3	1
5016	5	Aluminum / Vinyl	1	1971	1204	1	4	1
5017	5	Aluminum / Vinyl	1	1958	1129	1	3	1
5018	6	Aluminum / Vinyl	2	1910	2274	2	5	2
5019	6	Aluminum / Vinyl	2	1895	1424	1	4	1
5021	7	Stone	1.5	1941	1643	1	3	2
5022	7	Aluminum / Vinyl	1	1947	1328	1	4	1
5023	7	Block	1	1948	1120	1	3	2
5024	7	Aluminum / Vinyl	1.5	1940	2124	2	3	2
5025	7	Aluminum / Vinyl	2	1942	1798	2	4	2
5026	8	Frame	2	1910	2208	2	4	2
5027	8	Aluminum / Vinyl	1.5	1913	2100	2	4	2
5028	8	Aluminum / Vinyl	1	1925	1357	1	4	1
5029	8	Stucco	2	1916	2000	1	3	2
5031	9	Aluminum / Vinyl	1	1951	1040	1	4	1
5032	9	Aluminum / Vinyl	2	2011	2096	1	4	2
5033	10	Brick	1.5	1936	1471	1	3	1
5034	10	Brick	1.5	1946	1402	1	4	2
5035	10	Aluminum / Vinyl	1	1890	1296	1	3	3
5036	10	Masonry / Frame	2	1926	2804	2	6	2
5037	10	Brick	1.5	1931	2505	2	4	3
5038	10	Masonry / Frame	2	1925	2495	2	6	2
5039	10	Frame	2	1920	2314	2	4	2
5040	10	Brick	1	1928	2315	1	4	1
5041	10	Aluminum / Vinyl	1	1927	1461	1	3	1
5042	10	Brick	1	1952	1292	1	3	1
5043	10	Aluminum / Vinyl	1	1919	1659	1	3	1
5044	10	Aluminum / Vinyl	1.5	1911	1525	1	3	1
5045	10	Brick	1	1931	1474	1	3	1
5046	11	Brick	1.5	1953	2008	1	3	2
5047	11	Aluminum / Vinyl	1	1954	1417	1	4	1
5048	11	Stone	1	1940	1221	1	3	1
5049	11	Brick	1	1953	1184	1	3	1
5050	11	Aluminum / Vinyl	1.5	1929	1760	2	4	2
5051	11	Masonry / Frame	2	1964	2229	2	6	2
5052	11	Aluminum / Vinyl	1.5	1978	2061	2	5	2
5053	11	Brick	1	1952	1489	1	3	2
5054	11	Brick	1	1960	1423	1	3	1
5055	11	Aluminum / Vinyl	1	1954	1271	1	3	1
5056	11	Brick	1	1954	1257	1	3	2
5057	11	Aluminum / Vinyl	1	1963	1168	1	3	1

5058	11	Brick	1	1959	1134	1	3	1
5059	11	Brick	1	1954	1116	1	3	1
5060	11	Fiber-Cement	1	1957	1099	1	3	2
5061	11	Brick	1	1959	1052	1	2	1
5062	11	Brick	1	1957	1040	1	3	1
5063	11	Brick	1	1953	1013	1	3	1
5064	11	Aluminum / Vinyl	1	1953	864	1	3	1
5065	11	Aluminum / Vinyl	1	1952	811	1	2	1
5066	11	Aluminum / Vinyl	1	1954	804	1	2	1
5067	12	Brick	2	1908	2566	2	4	2
5068	12	Aluminum / Vinyl	1	1890	1547	2	4	2
5069	12	Aluminum / Vinyl	1	1884	1370	1	4	1
5070	13	Brick	1	1966	2073	1	4	2
5071	13	Aluminum / Vinyl	1.5	1949	1373	1	3	2
5072	13	Aluminum / Vinyl	1	1952	1247	1	3	2
5073	13	Brick	1	1949	1200	1	3	1
5074	13	Aluminum / Vinyl	1	1942	1099	1	3	1
5075	13	Brick	1	1954	971	1	2	1
5076	13	Aluminum / Vinyl	1	1947	860	1	2	1
5077	13	Aluminum / Vinyl	1	1941	732	1	2	1
5078	13	Aluminum / Vinyl	2	1966	1957	1	3	1
5079	13	Aluminum / Vinyl	1	1925	1349	1	3	2
5080	13	Aluminum / Vinyl	1	1969	1824	1	3	1
5081	13	Brick	1	1973	1501	1	3	1
5082	13	Aluminum / Vinyl	1	1969	1345	1	3	1
5083	13	Aluminum / Vinyl	1	1960	1139	1	3	1
5084	13	Aluminum / Vinyl	1	1942	763	1	2	1
5085	14	Aluminum / Vinyl	1	1937	948	1	2	1
5086	14	Aluminum / Vinyl	1.5	1922	1384	2	3	3
5087	14	Aluminum / Vinyl	2	1900	1496	2	3	2
5088	14	Brick	1	1931	1472	1	4	2
5089	14	Brick	1	1956	1002	1	3	1
5090	14	Aluminum / Vinyl	1	1952	976	1	2	1
5091	14	Aluminum / Vinyl	1.5	1915	1557	1	4	2
5092	15	Aluminum / Vinyl	2	1915	2616	2	6	2
5095	1	Aluminum / Vinyl	1	1955	1188	1	4	1
5096	1	Aluminum / Vinyl	1	1940	1077	1	4	1
5097	2	Brick	1	1952	1321	1	4	1
5098	2	Frame	1.5	1955	1208	1	4	1
5099	2	Aluminum / Vinyl	1	1953	984	1	3	1
5100	2	Stone	1	1946	2152	2	3	2
5101	2	Frame	1	1929	1489	1	5	2
5102	2	Brick	1	1961	1261	1	3	1

5103	2	Aluminum / Vinyl	1	1965	1144	1	3	1
5104	2	Aluminum / Vinyl	1	1961	1101	1	3	1
5105	2	Aluminum / Vinyl	1	1957	882	1	3	2
5106	2	Aluminum / Vinyl	2	1984	2504	2	6	2
5107	3	Frame	2	1929	2259	2	4	2
5108	3	Aluminum / Vinyl	1.5	1922	1659	2	4	2
5109	3	Stone	2	1929	7664	1	7	4
5110	3	Brick	2	1909	5862	1	5	4
5111	3	Brick	2	1926	4843	1	6	>4
5112	3	Frame	2	1896	3342	1	6	2
5113	3	Aluminum / Vinyl	2	1898	2887	1	4	2
5114	3	Masonry / Frame	2	1906	2490	1	3	2
5115	3	Frame	2	1900	2206	1	4	1
5116	3	Frame	1	1914	1692	1	4	1
5117	3	Aluminum / Vinyl	1.5	1889	1688	1	3	1
5118	3	Brick	2	1912	4792	1	6	3
5119	4	Aluminum / Vinyl	1	1890	1302	1	3	1
5120	5	Brick	1.5	1952	1624	1	3	2
5121	5	Stone	1	1948	1331	1	4	2
5122	5	Aluminum / Vinyl	1	1940	1127	1	2	1
5123	5	Aluminum / Vinyl	1	1952	1114	1	4	2
5124	5	Aluminum / Vinyl	2	2011	3302	1	4	3
5125	5	Brick	2	1955	2101	1	3	1
5126	5	Masonry / Frame	2	1955	2816	2	6	2
5127	5	Masonry / Frame	2	1957	2222	2	5	2
5128	5	Brick	1	1969	1642	1	3	1
5129	5	Brick	1	1951	1388	1	3	1
5130	5	Brick	1	1956	1234	1	3	1
5131	5	Brick	1	1954	1185	1	3	1
5132	5	Brick	1	1959	1124	1	3	2
5133	5	Brick	1	1951	1098	1	3	2
5134	5	Masonry / Frame	1	1954	1065	1	3	2
5135	5	Aluminum / Vinyl	1	1959	1063	1	3	1
5136	5	Aluminum / Vinyl	2	1981	2423	2	6	2
5137	6	Aluminum / Vinyl	2	1910	2952	2	6	2
5138	6	Aluminum / Vinyl	2	1908	2296	2	4	2
5139	6	Aluminum / Vinyl	2	1908	2294	2	4	2
5140	6	Aluminum / Vinyl	1	1925	1384	1	2	1
5141	6	Aluminum / Vinyl	2	1906	2092	1	4	2
5142	6	Aluminum / Vinyl	2	1910	2860	3	5	3
5143	7	Aluminum / Vinyl	1	1930	1362	1	3	1
5144	7	Aluminum / Vinyl	1.5	1924	1330	1	3	2
5145	7	Aluminum / Vinyl	1.5	1925	1229	1	3	2

5146	7	Brick	1.5	1956	2597	2	6	2
5147	8	Aluminum / Vinyl	2	1912	1408	2	4	2
5148	8	Aluminum / Vinyl	1.5	1928	1634	1	3	2
5149	9	Aluminum / Vinyl	2	2011	3441	1	4	2
5150	9	Aluminum / Vinyl	1	2011	1566	1	3	2
5151	9	Aluminum / Vinyl	1	1971	1108	1	4	1
5152	9	Aluminum / Vinyl	1	1960	936	1	3	1
5153	10	Brick	1	1949	1338	1	4	3
5154	10	Aluminum / Vinyl	1.5	1948	1265	1	3	2
5155	10	Brick	1.5	1926	2213	2	4	2
5156	10	Aluminum / Vinyl	1.5	1924	1942	2	4	2
5157	10	Aluminum / Vinyl	1.5	1924	1797	2	4	2
5158	10	Aluminum / Vinyl	2	1911	2250	2	6	2
5159	10	Aluminum / Vinyl	1	1919	1641	1	4	2
5160	10	Stucco	1	1924	1293	1	3	1
5161	10	Stucco	1	1924	1089	1	2	1
5162	10	Aluminum / Vinyl	1	1948	880	1	1	1
5163	10	Aluminum / Vinyl	2	1886	1476	1	3	1
5164	11	Aluminum / Vinyl	1	1953	1186	1	4	1
5165	11	Aluminum / Vinyl	1	1953	1177	1	4	1
5166	11	Aluminum / Vinyl	1	1941	907	1	4	1
5167	11	Brick	1	1955	1697	1	3	2
5168	11	Brick	1	1959	1307	1	3	1
5169	11	Brick	1	1959	1126	1	3	1
5170	11	Brick	1	1956	1057	1	3	1
5171	11	Aluminum / Vinyl	1	1953	1035	1	3	1
5172	11	Aluminum / Vinyl	1	1954	991	1	3	1
5173	11	Aluminum / Vinyl	1	1956	979	1	3	1
5174	11	Brick	1	1960	965	1	3	1
5175	11	Aluminum / Vinyl	1	1954	933	1	3	1
5176	11	Brick	1	1954	838	1	2	1
5177	11	Aluminum / Vinyl	1	1951	811	1	2	1
5178	11	Aluminum / Vinyl	1	1952	749	1	2	1
5179	12	Aluminum / Vinyl	1	1884	832	1	2	1
5180	13	Brick	1	1942	1491	1	3	2
5181	13	Aluminum / Vinyl	1	1951	1091	1	3	1
5182	13	Brick	2	1961	2156	2	4	2
5183	13	Aluminum / Vinyl	1.5	1929	1837	2	3	2
5184	13	Aluminum / Vinyl	1	1930	1348	1	4	1
5185	13	Brick	1	1971	1674	1	4	1
5186	13	Aluminum / Vinyl	1	1964	1342	1	3	1
5187	13	Brick	1	1953	1044	1	3	1
5188	14	Aluminum / Vinyl	1	1947	1022	1	3	1

5189	14	Aluminum / Vinyl	1	1927	1248	1	3	1
5190	14	Aluminum / Vinyl	1	1926	1004	1	3	1
5191	14	Frame	2	1904	1813	1	4	1
5192	14	Aluminum / Vinyl	2	1908	1276	1	2	1
5193	15	Aluminum / Vinyl	1.5	1925	1950	2	4	2
5194	15	Aluminum / Vinyl	2	1908	2976	2	6	2
5195	1	Aluminum / Vinyl	1	1940	1206	1	3	1
5196	1	Aluminum / Vinyl	1	1952	908	1	2	1
5197	2	Brick	1	1953	1512	1	3	2
5198	2	Aluminum / Vinyl	1	1956	1326	1	4	1
5199	2	Aluminum / Vinyl	2	2003	1991	1	4	2
5200	2	Aluminum / Vinyl	1	1955	1247	1	4	1
5201	2	Aluminum / Vinyl	1	1963	1151	1	3	2
5202	2	Brick	1	1957	1148	1	3	2
5203	2	Brick	1	1956	1144	1	2	2
5204	2	Masonry / Frame	1	1948	1032	1	3	1
5205	2	Brick	1	1950	794	1	2	1
5206	2	Aluminum / Vinyl	2	1981	2402	2	6	2
5207	3	Brick	2	1920	3212	1	5	4
5208	3	Brick	1.5	1925	2234	2	4	2
5209	3	Aluminum / Vinyl	2	1940	1872	2	4	2
5210	3	Brick	2	1921	3346	1	4	2
5211	3	Frame	2	1898	2492	1	5	2
5212	5	Stone	1	1947	1442	1	3	1
5213	5	Brick	1	1949	1338	1	3	2
5214	5	Brick	1	1955	1221	1	4	2
5215	5	Brick	1	1950	1146	1	3	1
5216	5	Aluminum / Vinyl	1	1953	693	1	2	1
5217	5	Aluminum / Vinyl	1	1957	1120	1	3	1
5218	5	Brick	1	1956	1031	1	3	2
5219	6	Frame	1	1890	1722	1	2	1
5220	6	Aluminum / Vinyl	2	1890	2288	2	5	2
5221	6	Aluminum / Vinyl	1	1890	1476	1	4	1
5222	7	Stone	1.5	1936	1775	1	4	3
5223	7	Brick	1	1947	1561	1	3	2
5224	7	Masonry / Frame	2	1926	1532	1	3	1
5225	7	Aluminum / Vinyl	1	1925	1596	1	3	2
5226	8	Aluminum / Vinyl	1	1890	1268	1	4	1
5227	8	Aluminum / Vinyl	1	1922	1138	1	3	1
5228	8	Aluminum / Vinyl	1	1913	1549	1	4	1
5229	8	Aluminum / Vinyl	1.5	1907	1398	1	3	2
5230	9	Aluminum / Vinyl	1	1955	984	1	3	1
5231	9	Aluminum / Vinyl	2	1978	1347	1	4	1

5232	10	Aluminum / Vinyl	1	1948	1170	1	4	1
5233	10	Aluminum / Vinyl	1.5	1924	1682	2	4	2
5234	10	Aluminum / Vinyl	2	1930	1232	2	2	2
5235	10	Aluminum / Vinyl	1.5	1927	1929	1	4	3
5236	10	Aluminum / Vinyl	1	1927	1513	1	4	2
5237	10	Aluminum / Vinyl	1	1926	1224	1	3	1
5238	10	Aluminum / Vinyl	1.5	1913	2135	1	3	2
5239	10	Aluminum / Vinyl	1	1924	840	1	2	1
5240	11	Frame	1	1971	2008	1	4	2
5241	11	Brick	1	1954	1442	1	4	1
5242	11	Aluminum / Vinyl	1	1953	1406	1	4	2
5243	11	Aluminum / Vinyl	1	1953	1113	1	3	1
5244	11	Masonry / Frame	2	1954	1671	2	3	2
5245	11	Brick	1	1929	2067	1	3	1
5246	11	Frame	1	1964	1126	1	3	2
5247	11	Brick	1	1950	893	1	2	1
5248	11	Stone	1	1948	819	1	2	1
5249	11	Aluminum / Vinyl	1	1942	762	1	2	1
5250	12	Aluminum / Vinyl	1	1925	1872	2	5	3
5251	13	Frame	1.5	1947	1276	1	3	2
5252	13	Brick	1	1949	1183	1	3	1
5253	13	Aluminum / Vinyl	1	1951	836	1	2	1
5254	13	Masonry / Frame	2	1968	2073	1	5	1
5255	13	Brick	1.5	1955	1518	2	4	2
5256	13	Aluminum / Vinyl	2	1920	2230	2	6	2
5257	13	Aluminum / Vinyl	1	1973	1106	1	4	1
5258	13	Aluminum / Vinyl	1	1959	978	1	3	1
5259	14	Stucco	1	1943	1131	1	2	1
5260	14	Aluminum / Vinyl	1.5	1923	1951	2	4	2
5261	14	Brick	1	1929	1677	1	3	2
5262	14	Aluminum / Vinyl	1	1924	1532	1	3	2
5263	14	Stucco	1.5	1911	2206	1	4	3
5264	14	Aluminum / Vinyl	2	1908	1726	1	3	1
5265	14	Aluminum / Vinyl	1	1906	1404	1	4	2
5266	14	Aluminum / Vinyl	1	1900	1374	1	3	2
5267	14	Frame	1	1910	1298	1	3	1
5268	14	Aluminum / Vinyl	1	1905	1136	1	4	1
5269	15	Brick	1	1919	2270	1	4	2
5270	1	Masonry / Frame	2	1970	1237	1	4	1
5271	1	Aluminum / Vinyl	1	1951	1302	1	4	1
5272	2	Aluminum / Vinyl	1	1955	1229	1	3	1
5273	2	Stone	1.5	1925	2033	2	4	2
5274	2	Brick	2	1957	2522	3	5	3

5275	3	Stucco	2	1920	2292	1	5	1
5276	3	Aluminum / Vinyl	>2	2003	2112	1	3	3
5277	3	Aluminum / Vinyl	2	1910	3944	2	6	2
5278	3	Aluminum / Vinyl	2	1912	2090	2	5	3
5279	3	Brick	2	1912	5044	1	5	3
5280	3	Brick	1.5	1930	2439	1	3	2
5281	4	Masonry / Frame	2	1907	3006	3	6	2
5282	5	Aluminum / Vinyl	1	1946	1108	1	3	2
5283	5	Aluminum / Vinyl	1	1947	1598	1	3	2
5284	5	Frame	1	1951	1256	1	3	2
5285	5	Aluminum / Vinyl	1	1956	1219	1	3	1
5286	5	Brick	1	1949	1216	1	2	1
5287	5	Aluminum / Vinyl	1	1954	1163	1	3	1
5288	5	Frame	1	1956	1093	1	3	1
5290	5	Brick	1	1956	1040	1	3	1
5291	5	Aluminum / Vinyl	1	1956	1019	1	3	1
5292	7	Stone	1	1950	1908	1	3	2
5293	8	Aluminum / Vinyl	2	1912	2516	2	5	2
5294	8	Aluminum / Vinyl	1.5	1921	1741	2	4	2
5295	8	Frame	1	1893	1385	1	4	1
5296	9	Frame	1	1970	1600	1	3	1
5297	9	Brick	1	1958	1396	1	3	1
5298	9	Aluminum / Vinyl	1	1961	967	1	3	1
5299	10	Frame	1	1949	1263	1	3	1
5300	10	Stucco	1	1919	2054	1	4	2
5301	10	Brick	1	1928	2008	1	3	1
5302	10	Brick	1	1928	1673	1	5	2
5303	10	Aluminum / Vinyl	1.5	1920	1098	1	3	1
5304	11	Aluminum / Vinyl	1	1953	1393	1	3	1
5305	11	Aluminum / Vinyl	1	1952	1016	1	3	1
5306	11	Aluminum / Vinyl	2	1964	1662	1	3	1
5307	11	Brick	2	1932	2225	2	4	2
5308	11	Frame	1	1928	1913	1	4	2
5309	11	Block	1	1951	1088	1	2	1
5310	11	Brick	2	1928	2733	1	3	1
5311	11	Aluminum / Vinyl	2	1964	2176	2	6	2
5312	13	Brick	1	1938	1443	1	3	2
5313	13	Aluminum / Vinyl	2	1994	1850	1	3	2
5314	13	Masonry / Frame	1.5	1959	1991	2	5	2
5315	13	Aluminum / Vinyl	1	1960	891	1	3	1
5316	14	Brick	1.5	1948	1401	1	4	1
5317	14	Aluminum / Vinyl	1	1957	1168	1	3	1
5318	14	Frame	2	1918	2034	2	6	2

5319	14	Frame	1	1912	1973	2	5	2
5320	14	Aluminum / Vinyl	1	1922	1462	1	4	1
5321	14	Aluminum / Vinyl	1.5	1913	1163	1	3	1
5322	14	Aluminum / Vinyl	1	1910	897	1	3	2
5323	14	Aluminum / Vinyl	1	1910	840	1	1	2
5324	15	Brick	1	1920	1974	1	3	1
5325	1	Aluminum / Vinyl	1	1942	1339	1	3	2
5326	1	Block	1.5	1947	1664	2	4	2
5327	1	Stucco	1	1924	1495	1	4	1
5328	1	Aluminum / Vinyl	1	1927	1087	1	3	1
5329	2	Brick	1	1951	1168	1	3	2
5330	2	Brick	1	1922	1174	1	2	2
5331	3	Frame	2	1898	2526	1	4	2
5332	3	Aluminum / Vinyl	2	1960	1534	1	4	1
5333	3	Frame	2	1902	2949	1	4	1
5334	3	Brick	2	1909	2068	1	4	1
5335	5	Brick	1	1952	1573	1	3	1
5336	5	Brick	1	1948	1512	1	3	1
5337	5	Frame	1	1952	1326	1	3	1
5338	5	Aluminum / Vinyl	1	1948	1256	1	4	1
5339	5	Brick	1	1947	1251	1	4	1
5340	5	Brick	1	1951	1205	1	3	1
5341	5	Brick	1	1949	1059	1	2	2
5342	5	Aluminum / Vinyl	1	1950	1040	1	3	1
5343	5	Aluminum / Vinyl	2	2010	2506	1	4	2
5344	5	Aluminum / Vinyl	2	2011	2480	1	4	2
5345	5	Brick	2	1952	2330	2	4	2
5346	5	Stone	1	1951	1703	1	2	2
5347	5	Aluminum / Vinyl	1	1969	1629	1	3	2
5348	5	Frame	1	1952	1407	1	3	1
5349	5	Frame	1	1951	1256	1	3	2
5350	5	Aluminum / Vinyl	1	1962	1238	1	3	2
5351	5	Brick	1	1959	1128	1	3	1
5352	6	Aluminum / Vinyl	2	1891	1980	2	6	3
5353	6	Prem Wood	2	1890	1878	1	4	2
5354	6	Aluminum / Vinyl	1	1892	1116	1	2	1
5355	7	Aluminum / Vinyl	1.5	1940	1504	1	3	1
5356	7	Stone	2	1945	1817	1	3	2
5357	7	Brick	2	1935	1555	1	3	1
5358	7	Brick	2	1944	1824	2	4	2
5359	8	Aluminum / Vinyl	1.5	1924	1896	2	4	2
5360	8	Aluminum / Vinyl	2	1903	1916	2	4	2
5361	8	Frame	2	1906	1875	2	4	2

5363	9	Fiber-Cement	2	2006	2028	1	3	2
5364	9	Brick	1	1957	997	1	3	1
5365	9	Aluminum / Vinyl	1	1975	906	1	3	1
5366	10	Aluminum / Vinyl	1	1984	1414	1	3	2
5367	10	Brick	1.5	1955	2230	1	4	2
5368	10	Aluminum / Vinyl	1	1953	1363	1	3	2
5369	10	Brick	1	1953	1350	1	3	1
5370	10	Masonry / Frame	2	1937	1671	1	3	1
5371	10	Aluminum / Vinyl	2	1914	2768	2	5	2
5372	10	Aluminum / Vinyl	1.5	1918	2178	1	5	2
5373	10	Frame	1	1919	1391	1	3	1
5374	10	Aluminum / Vinyl	1	1923	1387	1	4	1
5375	10	Frame	1	1926	1354	1	3	1
5376	10	Aluminum / Vinyl	1	1917	964	1	3	1
5377	10	Aluminum / Vinyl	2	1926	1356	1	4	1
5378	10	Aluminum / Vinyl	1	1928	906	1	2	1
5379	11	Aluminum / Vinyl	1	1942	1728	1	3	1
5380	11	Aluminum / Vinyl	1	1953	1525	1	4	2
5381	11	Brick	1	1948	1477	1	4	2
5382	11	Aluminum / Vinyl	1	1948	958	1	3	1
5383	11	Stone	2	1937	1676	1	3	1
5384	11	Masonry / Frame	2	1954	2168	2	4	2
5385	11	Aluminum / Vinyl	1.5	1955	1697	2	4	2
5386	11	Frame	1	1928	1121	1	3	1
5387	11	Aluminum / Vinyl	1	1953	1727	1	3	2
5388	11	Brick	1	1951	1455	1	3	1
5389	11	Brick	1	1967	1232	1	3	1
5390	11	Brick	1	1955	1166	1	3	2
5391	11	Aluminum / Vinyl	1	1970	1107	1	3	2
5392	11	Aluminum / Vinyl	1	1956	1073	1	3	1
5393	11	Brick	1	1953	1013	1	3	1
5394	11	Aluminum / Vinyl	1	1958	936	1	3	1
5395	11	Aluminum / Vinyl	1	1939	858	1	2	1
5396	11	Frame	1	1956	672	1	2	1
5397	12	Aluminum / Vinyl	2	2009	1848	1	4	2
5398	12	Aluminum / Vinyl	2	1909	2117	2	5	2
5399	12	Aluminum / Vinyl	1	1930	1588	1	4	1
5400	13	Brick	1	1947	1292	1	4	1
5401	13	Block	1	1940	1227	1	4	2
5402	13	Brick	1	1950	1190	1	3	1
5403	13	Aluminum / Vinyl	1	1950	1141	1	3	2
5404	13	Aluminum / Vinyl	1	1949	979	1	3	1
5405	13	Aluminum / Vinyl	2	1994	1636	1	3	1

5406	13	Brick	2	1949	1404	1	3	1
5407	13	Aluminum / Vinyl	1.5	1930	1397	2	4	2
5408	13	Brick	1	1968	1463	1	2	1
5409	13	Brick	1	1963	1242	1	3	1
5410	13	Brick	1	1963	1104	1	3	1
5411	13	Aluminum / Vinyl	1	1960	891	1	3	1
5412	13	Fiber-Cement	1	1929	1056	1	3	1
5413	13	Aluminum / Vinyl	2	1976	2538	2	6	2
5414	14	Aluminum / Vinyl	1	1947	1132	1	3	1
5415	14	Aluminum / Vinyl	1	1948	1054	1	3	1
5416	14	Prem Wood	2	1981	1574	1	2	1
5417	14	Aluminum / Vinyl	2	1941	1196	1	2	1
5418	14	Aluminum / Vinyl	1.5	1925	1982	2	5	2
5419	14	Aluminum / Vinyl	2	1885	2660	2	5	2
5420	14	Brick	1	1952	961	1	3	1
5421	14	Brick	2	1922	2761	1	5	1
5422	14	Frame	2	1888	1627	1	3	2
5423	14	Aluminum / Vinyl	1.5	1885	1232	1	3	2
5424	15	Aluminum / Vinyl	2	2000	1449	1	3	1
5425	15	Aluminum / Vinyl	2	1924	2022	2	4	2
5426	15	Aluminum / Vinyl	1	1920	1655	1	4	2
5427	15	Frame	1	1922	1553	1	4	1
5428	1	Stone	1	1936	1560	1	4	1
5429	1	Aluminum / Vinyl	1	1941	1062	1	3	1
5430	1	Brick	1	1945	1033	1	3	1
5431	1	Brick	1	1948	1302	1	4	2
5432	2	Brick	1	1952	1699	1	3	1
5433	2	Frame	1	1970	1120	1	4	1
5434	2	Aluminum / Vinyl	1	1959	1103	1	3	1
5435	2	Aluminum / Vinyl	1	1950	936	1	3	1
5436	3	Frame	2	1906	5160	2	5	4
5437	3	Aluminum / Vinyl	2	1927	1725	1	3	1
5438	3	Masonry / Frame	2	1906	3056	2	6	2
5439	3	Aluminum / Vinyl	1.5	1898	1594	2	3	2
5440	3	Frame	1	1920	1420	1	3	1
5441	3	Brick	2	1922	4044	1	5	3
5442	3	Aluminum / Vinyl	1.5	1900	1786	1	4	1
5443	3	Brick	1.5	1939	1313	1	2	1
5444	4	Aluminum / Vinyl	2	1898	1882	1	4	1
5445	4	Aluminum / Vinyl	1.5	1888	1568	1	3	1
5446	5	Brick	1	1950	1530	1	4	2
5447	5	Brick	1	1947	1294	1	3	1
5448	5	Aluminum / Vinyl	1.5	1956	1271	1	4	1

5449	5	Brick	1	1949	1186	1	3	2
5450	5	Aluminum / Vinyl	1	1947	1072	1	3	1
5451	5	Stone	1	1950	989	1	2	1
5452	5	Aluminum / Vinyl	1	1951	971	1	3	1
5453	5	Aluminum / Vinyl	1	1953	946	1	3	1
5454	5	Aluminum / Vinyl	2	1950	1378	1	3	1
5455	5	Brick	1.5	1980	2366	2	5	2
5456	5	Masonry / Frame	1	1950	1733	1	3	2
5457	5	Aluminum / Vinyl	1	1968	1616	1	3	2
5458	5	Brick	1	1953	1308	1	3	2
5459	5	Brick	1	1959	1290	1	3	2
5460	5	Aluminum / Vinyl	1	1949	1286	1	3	1
5461	5	Brick	1	1958	1276	1	3	1
5462	5	Aluminum / Vinyl	1	1954	1168	1	3	1
5463	5	Masonry / Frame	1	1954	1137	1	3	1
5464	5	Frame	1	1960	1115	1	3	1
5465	5	Aluminum / Vinyl	1	1952	1040	1	2	1
5466	5	Aluminum / Vinyl	1	1956	1019	1	3	1
5467	6	Aluminum / Vinyl	2	1890	2028	2	4	2
5468	7	Brick	1	1945	1460	1	4	2
5469	7	Stone	1	1938	1396	1	3	1
5470	7	Aluminum / Vinyl	1	1942	1357	1	3	1
5471	7	Aluminum / Vinyl	2	1940	1364	1	3	1
5472	7	Frame	2	1924	2618	2	4	2
5473	7	Masonry / Frame	2	1924	2930	1	5	2
5474	7	Brick	1	1924	2429	1	4	2
5476	7	Brick	1	1956	1498	1	3	2
5477	8	Aluminum / Vinyl	1.5	1913	1783	2	4	2
5478	8	Aluminum / Vinyl	1	1925	1261	1	4	2
5479	8	Frame	1	1923	945	1	3	1
5480	9	Aluminum / Vinyl	2	1999	1503	1	3	1
5481	9	Aluminum / Vinyl	1	1957	1560	1	4	1
5482	9	Aluminum / Vinyl	1	1971	1210	1	3	1
5483	10	Stone	1	1939	1688	1	3	1
5484	10	Brick	1.5	1937	1496	1	4	1
5485	10	Frame	1	1953	1484	1	3	2
5486	10	Brick	1	1952	1376	1	4	1
5487	10	Brick	1	1923	1342	1	3	2
5488	10	Aluminum / Vinyl	1	1949	1204	1	3	2
5489	10	Aluminum / Vinyl	1	1952	1144	1	3	1
5490	10	Frame	1	1890	920	1	3	1
5491	10	Frame	2	1923	2530	2	4	2
5492	10	Stone	2	1937	2417	2	4	2

5493	10	Aluminum / Vinyl	2	1923	2212	2	4	2
5494	10	Masonry / Frame	1.5	1929	2053	2	4	2
5495	10	Aluminum / Vinyl	1.5	1928	1973	2	4	2
5496	10	Frame	1	1918	1835	1	3	2
5497	10	Frame	1	1900	1684	1	3	1
5498	10	Aluminum / Vinyl	1	1930	1454	1	4	1
5499	10	Aluminum / Vinyl	1	1924	1235	1	3	1
5500	10	Aluminum / Vinyl	1	1929	1169	1	3	1
5501	10	Aluminum / Vinyl	1	1926	1120	1	4	1
5502	10	Aluminum / Vinyl	1	1926	1010	1	3	1
5503	10	Stone	1.5	1940	1890	1	3	2
5504	11	Brick	1	1950	1588	1	4	1
5505	11	Aluminum / Vinyl	1	1954	1321	1	3	1
5507	11	Aluminum / Vinyl	1	1949	1106	1	3	1
5508	11	Aluminum / Vinyl	1	1950	1032	1	3	1
5509	11	Masonry / Frame	2	1989	2481	1	4	2
5510	11	Masonry / Frame	2	1968	1577	1	4	1
5511	11	Masonry / Frame	2	1955	2096	2	4	2
5512	11	Brick	1.5	1956	1846	2	4	2
5513	11	Aluminum / Vinyl	2	1954	1728	2	6	2
5514	11	Brick	1	1959	1533	1	3	1
5515	11	Brick	1	1968	1506	1	3	1
5516	11	Brick	1	1956	1334	1	3	2
5517	11	Brick	1	1958	1304	1	2	2
5518	11	Brick	1	1963	1172	1	3	1
5519	11	Frame	1	1964	1128	1	3	1
5520	11	Aluminum / Vinyl	1	1954	1065	1	3	1
5521	11	Brick	1	1958	1050	1	3	1
5522	11	Brick	1	1953	1013	1	3	1
5523	11	Brick	1	1930	1822	1	4	2
5524	12	Aluminum / Vinyl	2	2010	1892	1	4	2
5525	12	Aluminum / Vinyl	2	2011	1848	1	4	2
5526	12	Frame	2	1909	2052	2	4	2
5527	12	Frame	2	1923	2032	2	6	2
5528	12	Aluminum / Vinyl	1	1902	1118	1	3	1
5529	13	Aluminum / Vinyl	1	1955	2475	1	4	1
5530	13	Brick	1	1949	1327	1	3	1
5531	13	Stone	1	1941	1291	1	3	1
5532	13	Aluminum / Vinyl	1	1946	1137	1	3	1
5533	13	Brick	1	1949	1120	1	3	1
5534	13	Aluminum / Vinyl	2	1940	1468	1	3	1
5535	13	Aluminum / Vinyl	2	1948	1300	1	3	1
5536	13	Aluminum / Vinyl	1	1976	1820	1	3	2

5537	13	Brick	1	1965	1116	1	3	1
5538	13	Brick	1	1952	1092	1	3	1
5539	13	Aluminum / Vinyl	1	1944	698	1	2	1
5540	13	Frame	1.5	1890	1602	1	3	2
5541	13	Aluminum / Vinyl	1.5	1924	1458	1	2	1
5542	13	Aluminum / Vinyl	1.5	1900	1141	1	2	1
5543	13	Frame	1	1900	1007	1	2	1
5544	14	Aluminum / Vinyl	1	1947	1021	1	3	1
5545	14	Aluminum / Vinyl	1	1949	998	1	4	1
5546	14	Frame	>2	1959	1996	1	3	2
5547	14	Aluminum / Vinyl	2	1940	1248	1	3	1
5548	14	Aluminum / Vinyl	1.5	1950	1456	2	3	2
5549	14	Aluminum / Vinyl	1	1924	1772	1	5	1
5550	14	Aluminum / Vinyl	1	1925	1474	1	3	2
5551	14	Aluminum / Vinyl	1	1926	1165	1	3	1
5552	14	Aluminum / Vinyl	1	1950	768	1	2	1
5553	14	Stucco	1.5	1889	1800	1	4	2
5554	14	Aluminum / Vinyl	1.5	1890	1766	1	3	2
5555	14	Aluminum / Vinyl	1	1910	1333	1	3	2
5556	14	Aluminum / Vinyl	1.5	1919	1247	1	3	2
5557	14	Aluminum / Vinyl	1	1898	1187	1	3	1
5558	14	Aluminum / Vinyl	1.5	1926	934	1	2	1
5560	15	Aluminum / Vinyl	1	1917	1863	1	3	2
5561	15	Fiber-Cement	1	2010	2008	1	4	2
5562	1	Prem Wood	1.5	1915	1386	1	4	2
5563	1	Aluminum / Vinyl	2	1923	1357	1	3	1
5564	1	Aluminum / Vinyl	1	1989	1548	1	3	1
5565	2	Brick	1	1953	1230	1	3	1
5566	2	Aluminum / Vinyl	1	1953	1207	1	3	1
5567	2	Brick	2	1961	2768	2	6	2
5568	2	Aluminum / Vinyl	1	1954	1256	1	3	1
5569	2	Aluminum / Vinyl	1	1957	909	1	3	1
5570	2	Brick	2	1953	2880	3	5	3
5571	3	Aluminum / Vinyl	1	1890	1000	1	2	1
5572	3	Aluminum / Vinyl	2	1897	2468	2	2	2
5573	3	Stone	2	1930	4517	1	5	3
5574	3	Masonry / Frame	2	1912	3605	1	7	2
5575	3	Masonry / Frame	2	1913	1795	1	4	1
5576	3	Aluminum / Vinyl	1	1875	1484	1	2	1
5577	3	Aluminum / Vinyl	1	1925	1152	1	2	1
5578	3	Aluminum / Vinyl	1	1904	1144	1	3	1
5579	3	Prem Wood	1.5	1885	1042	1	2	1
5580	3	Brick	2	1922	3858	1	5	3

5581	3	Masonry / Frame	2	1910	3795	1	5	3
5582	4	Frame	2	1904	1786	1	4	2
5583	5	Aluminum / Vinyl	1.5	1950	1402	1	3	1
5584	5	Brick	1	1952	1354	1	3	1
5585	5	Aluminum / Vinyl	1	1952	1351	1	4	2
5586	5	Aluminum / Vinyl	1	1951	1275	1	3	1
5587	5	Block	1	1948	1267	1	3	1
5588	5	Brick	1	1950	1259	1	3	2
5589	5	Brick	1	1949	1246	1	3	1
5590	5	Aluminum / Vinyl	1	1952	1235	1	4	2
5591	5	Aluminum / Vinyl	1	1951	1209	1	4	1
5592	5	Aluminum / Vinyl	1	1951	1172	1	3	1
5593	5	Aluminum / Vinyl	1	1952	1089	1	3	1
5594	5	Brick	1	1946	1087	1	3	1
5595	5	Aluminum / Vinyl	1	1950	988	1	3	1
5596	5	Masonry / Frame	2	1952	1680	1	3	1
5597	5	Aluminum / Vinyl	2	1941	1486	1	3	1
5598	5	Brick	2	1969	2668	2	6	2
5599	5	Brick	2	1957	2234	2	6	2
5600	5	Brick	1	1956	1269	1	4	1
5601	5	Aluminum / Vinyl	1	1952	1138	1	3	2
5602	5	Brick	1	1960	1117	1	2	1
5603	5	Brick	1	1956	1112	1	3	1
5604	5	Aluminum / Vinyl	1	1955	1028	1	3	1
5605	5	Aluminum / Vinyl	1	1954	984	1	3	2
5606	5	Aluminum / Vinyl	1	1949	912	1	3	1
5607	5	Aluminum / Vinyl	1	1949	842	1	2	1
5608	6	Aluminum / Vinyl	2	1895	2833	1	6	2
5609	6	Aluminum / Vinyl	1	1927	1200	1	3	2
5610	6	Aluminum / Vinyl	1.5	1915	1779	1	5	2
5611	6	Aluminum / Vinyl	2	1930	1500	1	3	1
5612	7	Aluminum / Vinyl	1	1940	1219	1	3	1
5613	7	Aluminum / Vinyl	2	1924	3056	2	6	2
5614	7	Stone	2	1936	2302	2	4	2
5615	7	Aluminum / Vinyl	1	1927	1576	1	3	1
5616	7	Brick	1	1927	1538	1	3	2
5617	7	Brick	2	1933	1886	1	2	1
5618	8	Frame	1	1890	1320	1	4	1
5619	8	Aluminum / Vinyl	2	1916	2260	2	4	2
5620	8	Aluminum / Vinyl	2	1908	1502	1	4	1
5621	8	Aluminum / Vinyl	1	1910	1110	1	3	1
5622	9	Aluminum / Vinyl	2	2008	2190	1	3	2
5623	9	Frame	2	1966	2108	1	4	2

5624	9	Frame	2	1968	1636	1	4	1
5625	9	Aluminum / Vinyl	1	1964	1227	1	3	1
5626	9	Aluminum / Vinyl	1	1958	1119	1	3	1
5627	9	Aluminum / Vinyl	1	1957	985	1	3	1
5628	9	Aluminum / Vinyl	1	1952	864	1	3	1
5629	10	Aluminum / Vinyl	1.5	1942	1715	1	4	2
5630	10	Brick	1	1938	1479	1	4	2
5631	10	Brick	1	1949	1296	1	3	1
5632	10	Aluminum / Vinyl	1	1954	1223	1	3	1
5633	10	Aluminum / Vinyl	1	1955	1024	1	3	1
5634	10	Aluminum / Vinyl	1	1953	965	1	3	1
5635	10	Stone	2	1946	1982	1	3	1
5636	10	Masonry / Frame	2	1956	1873	1	3	1
5637	10	Aluminum / Vinyl	1.5	1916	1658	1	3	1
5638	10	Aluminum / Vinyl	2	1945	1260	1	3	1
5639	10	Frame	2	1912	3012	2	6	2
5640	10	Brick	1	1930	2099	1	4	4
5641	10	Stucco	1.5	1915	2091	1	4	2
5642	10	Brick	1.5	1927	1980	1	4	2
5643	10	Brick	1	1928	1944	1	3	2
5644	10	Brick	1	1925	1715	1	4	1
5645	10	Brick	1	1927	1633	1	3	1
5646	10	Aluminum / Vinyl	1	1926	1462	1	3	1
5647	10	Aluminum / Vinyl	1	1953	918	1	3	1
5648	10	Aluminum / Vinyl	2	1920	1464	1	3	2
5649	10	Aluminum / Vinyl	2	1926	1444	1	3	2
5650	11	Stucco	1.5	1955	2580	1	4	2
5651	11	Brick	1.5	1937	1751	1	3	1
5652	11	Brick	1.5	1952	1395	1	4	1
5653	11	Aluminum / Vinyl	1	1938	1157	1	4	1
5654	11	Brick	1	1950	1148	1	3	1
5655	11	Aluminum / Vinyl	1	1951	1096	1	3	1
5656	11	Brick	1	1949	1044	1	2	1
5657	11	Aluminum / Vinyl	1	1922	999	1	3	1
5658	11	Aluminum / Vinyl	2	1959	2174	1	4	2
5659	11	Aluminum / Vinyl	2	1940	1256	1	3	1
5660	11	Stone	1.5	1952	1977	2	4	2
5661	11	Brick	1.5	1942	1636	2	4	2
5662	11	Aluminum / Vinyl	1	1928	1452	1	4	1
5663	11	Aluminum / Vinyl	1	1953	1296	1	3	1
5664	11	Frame	1	1964	1292	1	4	1
5665	11	Brick	1	1959	1181	1	3	1
5666	11	Aluminum / Vinyl	1	1954	1158	1	3	1

5667	11	Brick	1	1954	1070	1	3	1
5668	11	Aluminum / Vinyl	1	1954	1034	1	3	1
5669	11	Brick	1	1955	1013	1	2	1
5670	11	Aluminum / Vinyl	1	1954	882	1	3	1
5671	11	Brick	1	1958	1845	1	4	2
5672	12	Frame	1.5	1885	1445	2	4	2
5673	13	Aluminum / Vinyl	1	1941	1436	1	3	1
5674	13	Aluminum / Vinyl	1	1947	1294	1	3	1
5675	13	Aluminum / Vinyl	1	1948	1154	1	3	2
5676	13	Aluminum / Vinyl	1	1943	1008	1	3	1
5677	13	Aluminum / Vinyl	2	1990	1856	1	2	3
5678	13	Aluminum / Vinyl	2	1957	2237	2	6	2
5679	13	Brick	2	1957	2224	2	5	2
5680	13	Brick	1	1961	1660	1	3	1
5681	13	Frame	1	1956	1650	1	3	2
5682	13	Aluminum / Vinyl	1	1973	1465	1	3	1
5683	13	Brick	1	1953	1188	1	3	2
5684	13	Aluminum / Vinyl	1	1959	1181	1	3	1
5685	13	Brick	1	1956	1150	1	3	2
5686	13	Stone	1	1951	967	1	3	1
5687	13	Brick	1	1932	1515	1	3	2
5688	13	Aluminum / Vinyl	1	1918	1410	1	4	1
5689	14	Stone	1.5	1948	1717	1	3	3
5690	14	Frame	1	1954	1273	1	3	2
5691	14	Aluminum / Vinyl	1	1953	1128	1	3	1
5692	14	Aluminum / Vinyl	2	1915	2745	2	5	3
5693	14	Aluminum / Vinyl	1.5	1928	1980	2	5	2
5694	14	Aluminum / Vinyl	1.5	1927	1900	2	4	2
5695	14	Aluminum / Vinyl	1.5	1925	1817	2	4	2
5696	14	Aluminum / Vinyl	1.5	1915	1892	1	4	2
5697	14	Stucco	1	1911	1294	1	3	2
5698	14	Aluminum / Vinyl	1	1925	1223	1	3	2
5699	14	Aluminum / Vinyl	1	1929	1198	1	3	1
5700	14	Aluminum / Vinyl	1	1923	1178	1	3	1
5701	14	Aluminum / Vinyl	1	1922	1060	1	3	1
5702	14	Aluminum / Vinyl	1	1954	1353	1	3	1
5703	14	Aluminum / Vinyl	1	1951	884	1	2	1
5704	14	Brick	1	1955	878	1	2	1
5705	14	Brick	1	1956	874	1	2	1
5706	14	Aluminum / Vinyl	1	1951	867	1	2	1
5707	14	Aluminum / Vinyl	1	1953	704	1	2	1
5708	14	Aluminum / Vinyl	2	2011	1785	1	2	3
5709	14	Aluminum / Vinyl	1.5	1889	1656	1	3	2

5710	14	Aluminum / Vinyl	1.5	1919	1414	1	4	2
5711	14	Aluminum / Vinyl	1	1920	1392	1	4	1
5712	14	Aluminum / Vinyl	1.5	1916	1305	1	3	1
5713	14	Aluminum / Vinyl	1	1887	1158	1	3	1
5714	14	Aluminum / Vinyl	1	1905	951	1	2	1
5715	15	Aluminum / Vinyl	1	1997	1588	1	4	2
5716	15	Aluminum / Vinyl	2	1913	2464	2	6	2
5717	15	Aluminum / Vinyl	1.5	1901	1741	1	3	1
5718	15	Frame	1	1894	1401	1	5	1
5719	15	Aluminum / Vinyl	1	1890	1100	1	3	1
5720	1	Brick	1	1947	1538	1	3	1
5721	1	Brick	1	1956	1174	1	3	1
5722	2	Brick	1	1952	1199	1	3	2
5723	2	Brick	1	1958	1017	1	2	1
5724	2	Aluminum / Vinyl	1	1961	1406	1	3	1
5725	2	Aluminum / Vinyl	1	1948	1091	1	3	1
5726	2	Brick	1	1958	1043	1	3	1
5727	3	Stucco	2	1913	2918	2	6	2
5728	3	Aluminum / Vinyl	2	1913	2362	2	4	3
5729	3	Aluminum / Vinyl	2	1894	1936	2	6	2
5730	3	Aluminum / Vinyl	1	1927	1035	1	3	1
5731	3	Stucco	2	1911	3553	1	6	3
5732	3	Stucco	2	1916	3097	1	6	3
5733	3	Frame	2	1889	2552	1	4	1
5734	3	Masonry / Frame	2	1911	1950	1	4	1
5735	3	Frame	2	1894	1924	1	3	1
5736	3	Frame	2	1902	1672	1	4	1
5737	3	Aluminum / Vinyl	1	1893	1610	1	4	1
5738	3	Frame	1	1890	1443	1	3	2
5739	3	Aluminum / Vinyl	1	1900	1382	1	3	2
5740	3	Frame	1	1894	1246	1	3	2
5741	3	Brick	2	1922	4733	1	6	>4
5742	5	Aluminum / Vinyl	1	1971	1566	1	4	1
5743	5	Aluminum / Vinyl	1	1950	1447	1	4	1
5744	5	Brick	1	1951	1386	1	4	1
5745	5	Aluminum / Vinyl	1	1948	1338	1	3	1
5746	5	Brick	1	1952	1336	1	3	1
5747	5	Aluminum / Vinyl	1.5	1952	1332	1	3	1
5748	5	Brick	1	1950	1329	1	3	1
5749	5	Aluminum / Vinyl	1	1950	1308	1	3	1
5750	5	Masonry / Frame	1	1959	1304	1	3	1
5751	5	Brick	1	1951	1273	1	3	1
5752	5	Aluminum / Vinyl	1	1949	1029	1	4	1

5753	5	Aluminum / Vinyl	2	2011	3110	1	5	3
5754	5	Brick	2	1948	2376	2	4	2
5755	5	Masonry / Frame	1.5	1957	2045	2	4	2
5756	5	Stone	1	1954	1415	1	3	2
5757	5	Stone	1	1956	1405	1	3	1
5758	5	Brick	1	1957	1288	1	3	1
5759	5	Brick	1	1962	1283	1	3	1
5760	5	Brick	1	1952	1268	1	2	1
5761	5	Brick	1	1959	1197	1	3	1
5762	5	Aluminum / Vinyl	1	1954	1168	1	3	1
5763	5	Aluminum / Vinyl	1	1958	1163	1	2	1
5764	5	Aluminum / Vinyl	1	1957	1163	1	3	1
5765	5	Brick	1	1954	1051	1	3	1
5766	5	Aluminum / Vinyl	1	1955	1008	1	3	1
5767	5	Frame	1	1952	873	1	2	1
5768	5	Aluminum / Vinyl	1	1950	828	1	2	1
5769	6	Brick	1.5	1929	2133	2	4	2
5770	6	Aluminum / Vinyl	2	1890	2013	2	4	2
5771	6	Aluminum / Vinyl	2	1890	1717	1	4	2
5773	7	Fiber-Cement	2	2003	1908	1	4	2
5774	7	Frame	1	1926	1453	1	5	1
5775	8	Frame	1	1925	850	1	2	1
5776	8	Aluminum / Vinyl	1	1896	1550	1	4	2
5777	8	Frame	1.5	1912	1383	1	3	1
5778	9	Aluminum / Vinyl	1	1954	1111	1	3	1
5779	9	Aluminum / Vinyl	2	2004	2016	1	3	2
5780	9	Aluminum / Vinyl	2	1969	1743	1	4	1
5781	9	Frame	2	1980	1445	1	3	1
5782	9	Aluminum / Vinyl	1	1958	1322	1	3	1
5783	9	Aluminum / Vinyl	1	1972	1134	1	3	1
5784	9	Aluminum / Vinyl	1	1964	1054	1	3	1
5785	9	Brick	1	1960	1014	1	3	1
5786	9	Frame	1	1963	1006	1	3	1
5787	9	Frame	1	1965	1006	1	3	1
5788	9	Aluminum / Vinyl	1	1974	906	1	3	1
5789	9	Aluminum / Vinyl	1	1985	1668	1	3	1
5790	10	Frame	1	1951	1410	1	3	2
5791	10	Stone	1	1951	1391	1	4	2
5792	10	Aluminum / Vinyl	1	1949	1114	1	3	1
5793	10	Brick	2	1938	1608	1	3	1
5794	10	Frame	2	1925	1536	1	3	2
5795	10	Frame	1	1925	720	1	1	1
5796	10	Aluminum / Vinyl	2	1930	2642	2	4	2

5797	10	Aluminum / Vinyl	1.5	1915	2220	1	4	1
5798	10	Aluminum / Vinyl	1	1922	1800	1	3	1
5799	10	Stucco	1	1918	1589	1	4	1
5800	10	Aluminum / Vinyl	1	1928	1584	1	4	1
5801	10	Aluminum / Vinyl	1	1925	1332	1	3	1
5802	10	Aluminum / Vinyl	1	1915	1296	1	4	2
5803	10	Frame	1	1925	1158	1	4	2
5804	10	Aluminum / Vinyl	1	1922	1125	1	3	1
5805	10	Aluminum / Vinyl	1.5	1923	1050	1	2	1
5806	10	Aluminum / Vinyl	1	1925	932	1	2	2
5807	10	Brick	1	1926	1449	1	2	2
5808	10	Aluminum / Vinyl	1	1969	1132	1	2	1
5809	10	Brick	1	1953	937	1	3	1
5810	10	Brick	1	1931	1548	1	3	1
5811	11	Stone	1.5	1942	1670	1	3	1
5812	11	Brick	1	1950	1450	1	4	2
5813	11	Aluminum / Vinyl	1	1938	1146	1	2	1
5814	11	Aluminum / Vinyl	1	1949	1072	1	2	2
5815	11	Aluminum / Vinyl	1	1942	1001	1	3	1
5816	11	Aluminum / Vinyl	1	1940	975	1	3	1
5817	11	Stucco	1	1938	928	1	3	1
5818	11	Aluminum / Vinyl	2	1958	2228	1	3	1
5819	11	Aluminum / Vinyl	2	2004	2174	1	3	2
5820	11	Aluminum / Vinyl	1.5	1923	1979	2	5	2
5821	11	Brick	1.5	1966	2274	2	5	2
5822	11	Brick	1.5	1961	1960	2	5	2
5823	11	Aluminum / Vinyl	1.5	1950	1669	2	3	2
5824	11	Brick	2	1932	2225	2	4	2
5825	11	Aluminum / Vinyl	1	1929	1653	1	4	2
5826	11	Brick	1	1962	1952	1	3	2
5827	11	Brick	1	1958	1524	1	3	1
5828	11	Brick	1	1962	1342	1	3	1
5829	11	Aluminum / Vinyl	1	1957	1137	1	3	1
5830	11	Brick	1	1954	1120	1	3	1
5831	11	Aluminum / Vinyl	1	1956	1073	1	3	1
5832	11	Brick	1	1956	1070	1	3	1
5833	11	Aluminum / Vinyl	1	1958	1055	1	2	2
5834	11	Aluminum / Vinyl	1	1956	1033	1	3	1
5835	11	Aluminum / Vinyl	1	1955	1020	1	3	1
5836	11	Aluminum / Vinyl	1	1955	972	1	3	2
5837	11	Aluminum / Vinyl	1	1955	867	1	3	1
5838	11	Aluminum / Vinyl	2	1990	3355	2	6	4
5840	12	Aluminum / Vinyl	1	1893	1870	2	5	2

5841	13	Brick	1	1951	1638	1	3	2
5842	13	Brick	1	1955	1619	1	4	2
5843	13	Aluminum / Vinyl	1	1946	1421	1	3	1
5844	13	Brick	1	1950	1347	1	2	1
5845	13	Brick	1	1959	1227	1	3	1
5846	13	Aluminum / Vinyl	1	1943	1008	1	3	1
5847	13	Frame	2	1984	1895	1	3	1
5848	13	Masonry / Frame	2	1977	1769	1	3	1
5849	13	Aluminum / Vinyl	2	1940	1144	1	2	1
5850	13	Masonry / Frame	2	1958	2352	2	4	2
5851	13	Masonry / Frame	2	1944	2111	2	4	2
5852	13	Aluminum / Vinyl	2	1952	2090	2	4	2
5853	13	Aluminum / Vinyl	1.5	1959	1744	2	4	2
5854	13	Stone	1.5	1952	1616	2	3	2
5855	13	Frame	1.5	1887	1919	2	5	1
5856	13	Aluminum / Vinyl	1	1929	1462	1	3	1
5857	13	Brick	1	1966	1726	1	4	2
5858	13	Brick	1	1963	1427	1	3	1
5859	13	Aluminum / Vinyl	1	1961	1411	1	4	1
5860	13	Aluminum / Vinyl	1	1976	1338	1	3	2
5861	13	Brick	1	1959	1174	1	3	1
5862	13	Aluminum / Vinyl	1	1960	1021	1	3	1
5863	13	Aluminum / Vinyl	1	1960	909	1	3	1
5864	13	Aluminum / Vinyl	1	1961	902	1	3	1
5866	13	Aluminum / Vinyl	1.5	1932	1600	1	5	2
5867	13	Aluminum / Vinyl	1.5	1925	1366	1	4	1
5868	14	Frame	1	1950	1361	1	4	1
5869	14	Aluminum / Vinyl	1	1949	1167	1	3	1
5870	14	Aluminum / Vinyl	1	1941	1150	1	2	1
5871	14	Aluminum / Vinyl	1	1951	1098	1	3	1
5872	14	Aluminum / Vinyl	1	1948	1048	1	3	1
5873	14	Aluminum / Vinyl	1	1890	1040	1	3	1
5874	14	Aluminum / Vinyl	1.5	1926	2100	2	4	2
5875	14	Aluminum / Vinyl	1	1895	1951	2	6	2
5876	14	Aluminum / Vinyl	1	1926	1726	1	3	2
5877	14	Aluminum / Vinyl	1	1925	1621	1	4	1
5878	14	Aluminum / Vinyl	1	1926	1425	1	4	2
5879	14	Aluminum / Vinyl	1	1919	1386	1	4	2
5881	14	Aluminum / Vinyl	1	1929	1164	1	3	1
5882	14	Aluminum / Vinyl	1	1919	848	1	3	1
5883	14	Brick	1	1961	984	1	3	1
5884	14	Aluminum / Vinyl	1	1947	784	1	2	1
5885	14	Aluminum / Vinyl	2	1915	1768	1	2	1

5886	14	Stucco	2	1913	1373	1	2	1
5887	14	Aluminum / Vinyl	1	1918	1368	1	2	1
5888	14	Aluminum / Vinyl	1	1911	997	1	2	1
5889	15	Aluminum / Vinyl	2	1918	2963	2	6	2
5890	15	Aluminum / Vinyl	2	1922	2712	2	4	2
5891	15	Brick	1	1927	2971	1	5	3
5892	15	Aluminum / Vinyl	1	1921	1407	1	4	1
5893	1	Aluminum / Vinyl	1	1940	1359	1	3	1
5894	1	Frame	1	1952	672	1	2	1
5895	2	Aluminum / Vinyl	1.5	1993	2324	1	3	2
5896	2	Aluminum / Vinyl	2	1961	2070	2	6	2
5897	2	Aluminum / Vinyl	1	1974	1959	1	4	2
5898	2	Stone	1	1954	1298	1	3	2
5899	2	Aluminum / Vinyl	1	1955	988	1	3	2
5900	2	Aluminum / Vinyl	1	1956	963	1	3	1
5901	2	Brick	1	1958	888	1	2	1
5902	2	Aluminum / Vinyl	1	1932	1231	1	3	1
5903	3	Stucco	1.5	1918	1305	1	2	1
5904	3	Masonry / Frame	2	1907	3809	1	4	2
5905	3	Aluminum / Vinyl	2	1905	2744	1	5	2
5906	3	Brick	2	1910	2694	1	5	2
5907	3	Stucco	2	1909	2629	1	5	2
5908	3	Masonry / Frame	2	1910	2414	1	3	1
5909	3	Masonry / Frame	2	1907	2358	1	4	2
5910	3	Stucco	2	1916	1937	1	4	1
5911	3	Aluminum / Vinyl	2	1896	1707	1	3	1
5912	3	Aluminum / Vinyl	1.5	1897	1345	1	4	2
5913	5	Stone	1	1949	2254	1	5	2
5914	5	Brick	1	1950	1400	1	4	2
5915	5	Brick	1	1956	1304	1	3	1
5916	5	Stone	1.5	1948	1301	1	3	1
5917	5	Frame	1	1950	1298	1	4	1
5918	5	Brick	1	1952	1167	1	3	1
5919	5	Brick	1	1950	1164	1	3	1
5920	5	Frame	1	1950	1149	1	3	1
5921	5	Aluminum / Vinyl	1	1951	1119	1	3	1
5922	5	Aluminum / Vinyl	1	1939	1049	1	2	1
5923	5	Aluminum / Vinyl	1	1950	846	1	3	1
5924	5	Aluminum / Vinyl	2	1980	2248	1	4	2
5925	5	Masonry / Frame	2	1955	2064	2	4	2
5926	5	Frame	1	1965	1865	1	3	2
5927	5	Frame	1	1965	1750	1	3	1
5928	5	Brick	1	1962	1225	1	3	1

5929	5	Brick	1	1957	1183	1	3	1
5930	5	Stone	1	1951	1147	1	3	1
5931	5	Aluminum / Vinyl	1	1950	1103	1	2	1
5932	5	Aluminum / Vinyl	1	1952	1084	1	3	1
5933	5	Aluminum / Vinyl	1	1963	1080	1	3	1
5934	5	Aluminum / Vinyl	1	1955	1008	1	3	1
5935	5	Aluminum / Vinyl	1	1953	977	1	3	2
5937	5	Brick	1	1953	952	1	2	1
5938	5	Aluminum / Vinyl	1	1949	915	1	2	1
5939	5	Aluminum / Vinyl	1	1951	772	1	2	1
5940	6	Frame	1.5	1922	2028	2	4	2
5941	6	Aluminum / Vinyl	1.5	1894	1629	2	4	2
5942	6	Aluminum / Vinyl	1	1925	1440	1	4	1
5943	6	Frame	2	1898	2572	1	4	2
5944	6	Aluminum / Vinyl	1.5	1880	1752	1	3	0
5945	7	Frame	1.5	1950	1336	1	4	2
5946	7	Brick	1	1951	1335	1	3	1
5947	7	Brick	2	1931	1734	1	4	2
5948	7	Aluminum / Vinyl	1.5	1931	1574	1	3	2
5949	7	Aluminum / Vinyl	1.5	1880	1415	1	5	2
5950	8	Aluminum / Vinyl	1.5	1907	1678	2	3	2
5951	8	Frame	1.5	1889	1645	1	4	2
5952	8	Aluminum / Vinyl	1.5	1906	1402	1	4	2
5953	8	Aluminum / Vinyl	1.5	1922	1197	1	3	1
5954	9	Aluminum / Vinyl	1	1952	990	1	4	1
5955	9	Fiber-Cement	2	1994	2325	1	3	2
5956	9	Aluminum / Vinyl	2	2008	2055	1	4	2
5957	9	Aluminum / Vinyl	1	1972	1210	1	3	1
5958	9	Aluminum / Vinyl	1	1965	1147	1	3	1
5959	9	Aluminum / Vinyl	1	1961	1041	1	3	1
5960	9	Frame	1	1959	898	1	3	1
5961	9	Frame	1	1957	1608	1	4	1
5962	9	Aluminum / Vinyl	1	1992	1547	1	3	2
5963	10	Stone	1.5	1941	1794	1	2	2
5964	10	Aluminum / Vinyl	1	1950	1482	1	3	1
5965	10	Brick	1	1949	1453	1	4	2
5966	10	Brick	1	1948	1395	1	4	1
5967	10	Aluminum / Vinyl	1	1953	1380	1	3	1
5968	10	Stone	1	1951	1290	1	3	2
5969	10	Brick	1	1940	1248	1	3	2
5970	10	Aluminum / Vinyl	1.5	1940	1153	1	3	1
5971	10	Aluminum / Vinyl	2	1948	1300	1	3	1
5972	10	Brick	1.5	1931	2553	2	4	2

5973	10	Aluminum / Vinyl	2	1923	2486	2	6	2
5974	10	Aluminum / Vinyl	2	1928	2364	2	5	2
5975	10	Stone	1.5	1935	2129	2	4	2
5976	10	Aluminum / Vinyl	1.5	1929	1386	2	3	2
5977	10	Aluminum / Vinyl	1.5	1894	1654	2	4	2
5978	10	Brick	1	1926	1863	1	3	2
5979	10	Aluminum / Vinyl	1.5	1926	1811	1	3	2
5980	10	Frame	1	1925	1734	1	4	2
5981	10	Brick	1	1927	1727	1	3	2
5982	10	Aluminum / Vinyl	1	1921	1719	1	3	1
5983	10	Aluminum / Vinyl	1	1925	1556	1	3	2
5984	10	Aluminum / Vinyl	1	1928	1445	1	4	2
5985	10	Aluminum / Vinyl	1	1925	1398	1	3	2
5986	10	Aluminum / Vinyl	1	1928	1373	1	4	1
5987	10	Brick	1	1965	1636	1	3	1
5988	10	Frame	1	1952	1056	1	3	2
5989	10	Brick	1	1948	788	1	2	1
5990	10	Brick	1.5	1930	1860	1	3	1
5991	10	Aluminum / Vinyl	1	1926	1318	1	3	0
5992	10	Aluminum / Vinyl	1	1927	691	1	2	1
5993	11	Stone	1	1936	1702	1	3	2
5994	11	Aluminum / Vinyl	1	1961	1098	1	3	1
5995	11	Aluminum / Vinyl	1	1942	1038	1	3	1
5996	11	Masonry / Frame	2	1953	1896	1	6	2
5997	11	Aluminum / Vinyl	2	1953	1554	1	3	1
5998	11	Aluminum / Vinyl	2	1938	1032	1	2	1
5999	11	Aluminum / Vinyl	2	1957	1946	2	4	2
6000	11	Brick	2	1954	1925	2	4	2
6001	11	Brick	2	1865	2274	2	4	2
6002	11	Frame	1	1987	1581	1	3	2
6003	11	Brick	1	1952	1305	1	2	1
6004	11	Brick	1	1959	1235	1	3	1
6005	11	Brick	1	1960	1201	1	3	1
6006	11	Aluminum / Vinyl	1	1965	1110	1	3	1
6007	11	Brick	1	1965	1057	1	3	1
6008	11	Aluminum / Vinyl	1	1958	1018	1	3	1
6009	11	Brick	1	1955	1013	1	3	1
6010	11	Brick	1	1959	1004	1	2	1
6011	11	Aluminum / Vinyl	1	1957	915	1	3	1
6012	11	Aluminum / Vinyl	1	1953	879	1	3	1
6013	11	Aluminum / Vinyl	1	1953	879	1	3	1
6014	11	Aluminum / Vinyl	1	1953	864	1	3	1
6015	12	Frame	2	1918	1796	2	4	2

6016	12	Aluminum / Vinyl	1.5	1890	1828	2	5	2
6017	12	Stone	1.5	1940	2315	1	5	2
6018	12	Brick	2	1895	3627	>3	>8	3
6019	13	Brick	1.5	1951	1822	1	4	2
6020	13	Brick	1	1976	1601	1	3	2
6021	13	Stucco	1	1951	1453	1	4	2
6022	13	Aluminum / Vinyl	1.5	1939	1338	1	3	1
6023	13	Brick	1	1949	1327	1	3	1
6024	13	Frame	1	1951	1300	1	3	1
6025	13	Brick	1	1964	1036	1	2	1
6026	13	Brick	1	1953	826	1	2	1
6027	13	Brick	1	1951	825	1	2	1
6028	13	Aluminum / Vinyl	1	1948	805	1	2	1
6029	13	Aluminum / Vinyl	2	1957	2098	2	6	2
6030	13	Aluminum / Vinyl	1.5	1959	1738	2	4	3
6031	13	Brick	1	1948	1497	2	3	2
6032	13	Aluminum / Vinyl	1.5	1923	1660	1	4	2
6033	13	Frame	1	1923	1653	1	4	2
6034	13	Brick	1	1966	1678	1	3	1
6035	13	Brick	1	1964	1116	1	3	1
6036	13	Aluminum / Vinyl	1	1959	936	1	3	1
6037	13	Aluminum / Vinyl	1	1900	1199	1	2	1
6038	14	Aluminum / Vinyl	1	1947	1468	1	3	2
6040	14	Fiber-Cement	>2	2005	3504	1	4	3
6041	14	Frame	1.5	1895	1190	1	3	1
6042	14	Aluminum / Vinyl	1	1890	592	1	1	1
6043	14	Aluminum / Vinyl	2	1923	2217	2	5	2
6044	14	Aluminum / Vinyl	2	1913	1968	2	5	2
6045	14	Aluminum / Vinyl	1.5	1900	1840	2	4	2
6046	14	Aluminum / Vinyl	1.5	1917	1582	1	3	2
6047	14	Aluminum / Vinyl	1	1925	1573	1	3	2
6048	14	Stucco	1	1922	1380	1	3	2
6049	14	Brick	1	1956	1505	1	3	1
6050	14	Aluminum / Vinyl	1	1950	923	1	3	1
6051	14	Aluminum / Vinyl	1	1947	826	1	2	1
6052	14	Brick	1.5	1918	1521	1	3	1
6053	14	Aluminum / Vinyl	1	1919	1290	1	3	1
6054	14	Aluminum / Vinyl	1	1895	1276	1	4	2
6055	14	Aluminum / Vinyl	2	1931	3265	3	7	3
6056	15	Aluminum / Vinyl	2	1883	2378	2	4	2
6057	15	Aluminum / Vinyl	1.5	1890	1746	2	5	2
6058	15	Brick	2	1905	2455	1	3	1
6059	1	Masonry / Frame	2	1931	2770	2	6	2

6060	1	Brick	1	1928	1706	1	4	1
6061	1	Aluminum / Vinyl	1	1955	1082	1	3	1
6062	2	Aluminum / Vinyl	1	1997	1360	1	3	2
6063	2	Aluminum / Vinyl	1	1955	1384	1	3	1
6064	2	Aluminum / Vinyl	1	1957	1084	1	3	2
6065	2	Aluminum / Vinyl	1	1959	1226	1	3	1
6066	2	Stone	1	1950	1147	1	3	1
6067	2	Brick	1	1962	1121	1	3	1
6068	2	Brick	1	1961	1071	1	3	1
6069	2	Brick	1	1958	1019	1	3	1
6070	2	Aluminum / Vinyl	1	1947	844	1	2	2
6071	2	Aluminum / Vinyl	1	1956	784	1	2	1
6072	2	Aluminum / Vinyl	1	1955	756	1	2	1
6073	2	Aluminum / Vinyl	1	1955	1486	1	3	1
6074	3	Frame	1	1956	1488	1	3	1
6075	3	Aluminum / Vinyl	1	1892	1662	2	3	2
6076	3	Aluminum / Vinyl	1	1901	900	1	1	1
6077	3	Frame	2	1924	2940	2	6	2
6078	3	Frame	2	1916	2420	2	4	2
6079	3	Stucco	2	1922	2047	2	4	2
6080	3	Aluminum / Vinyl	2	1895	2310	2	4	2
6081	3	Brick	2	1916	4562	1	6	3
6082	3	Aluminum / Vinyl	2	1898	3185	2	4	2
6083	3	Brick	2	1923	3137	1	4	2
6084	3	Brick	2	1906	2403	1	4	1
6085	3	Frame	2	1905	2328	1	4	2
6086	3	Aluminum / Vinyl	2	1912	1749	1	4	1
6087	4	Aluminum / Vinyl	2	1970	1471	1	4	1
6088	4	Frame	2	1902	3685	2	5	3
6089	4	Aluminum / Vinyl	2	1910	1661	1	3	1
6090	5	Brick	1	1953	1609	1	4	2
6091	5	Aluminum / Vinyl	1	1955	1477	1	3	1
6092	5	Brick	1	1949	1467	1	3	1
6094	5	Aluminum / Vinyl	1.5	1950	1251	1	3	2
6095	5	Aluminum / Vinyl	1	1953	1181	1	3	1
6096	5	Brick	1	1947	1158	1	2	1
6097	5	Aluminum / Vinyl	1	1950	1105	1	3	1
6098	5	Aluminum / Vinyl	1	1950	1092	1	3	3
6099	5	Aluminum / Vinyl	1	1953	1027	1	3	1
6100	5	Masonry / Frame	2	1952	1795	1	3	1
6101	5	Aluminum / Vinyl	2	1955	1419	1	3	1
6102	5	Aluminum / Vinyl	2	1939	1248	1	2	1
6103	5	Aluminum / Vinyl	2	1967	2100	2	6	2

6104	5	Aluminum / Vinyl	1.5	1963	2018	2	5	2
6105	5	Brick	1	1952	1516	1	3	1
6106	5	Frame	1	1964	1491	1	3	1
6107	5	Stone	1	1958	1252	1	3	1
6108	5	Brick	1	1960	1238	1	3	1
6109	5	Aluminum / Vinyl	1	1964	1212	1	3	1
6110	5	Aluminum / Vinyl	1	1961	1024	1	3	1
6111	5	Aluminum / Vinyl	1	1956	1019	1	2	2
6112	5	Aluminum / Vinyl	1	1956	1014	1	3	1
6113	5	Aluminum / Vinyl	1	1957	987	1	3	1
6114	5	Brick	1	1956	965	1	3	1
6115	5	Aluminum / Vinyl	1	1952	912	1	3	1
6116	5	Aluminum / Vinyl	1	1952	910	1	3	1
6117	5	Aluminum / Vinyl	1	1952	768	1	2	1
6118	6	Aluminum / Vinyl	2	2004	2366	1	3	2
6119	6	Aluminum / Vinyl	2	1924	2192	2	4	2
6120	6	Brick	1	1927	2212	1	5	2
6121	6	Aluminum / Vinyl	1.5	1881	1871	1	3	2
6122	7	Brick	2	1932	2696	2	5	3
6123	7	Aluminum / Vinyl	2	1924	2422	2	4	2
6124	7	Aluminum / Vinyl	1	1920	1781	1	5	1
6125	7	Aluminum / Vinyl	1	1932	1233	1	3	1
6126	7	Brick	1.5	1940	1802	1	3	1
6127	8	Frame	2	1922	2557	2	6	2
6128	8	Aluminum / Vinyl	1.5	1885	2071	2	5	3
6129	9	Frame	1	1957	1181	1	3	1
6130	9	Aluminum / Vinyl	1	1962	1001	1	3	1
6131	10	Brick	1	1952	1346	1	4	1
6132	10	Aluminum / Vinyl	1	1951	1134	1	3	2
6133	10	Brick	1	1941	1031	1	3	1
6134	10	Brick	1	1950	932	1	2	1
6135	10	Masonry / Frame	2	1941	1546	1	3	1
6136	10	Masonry / Frame	2	1941	1448	1	3	1
6137	10	Aluminum / Vinyl	2	1947	1117	1	2	1
6138	10	Brick	1.5	1928	2283	2	4	2
6139	10	Aluminum / Vinyl	1.5	1919	2012	2	4	2
6140	10	Aluminum / Vinyl	1	1919	1939	1	3	1
6141	10	Brick	2	1926	1792	1	4	2
6142	10	Aluminum / Vinyl	1	1927	1582	1	4	2
6143	10	Aluminum / Vinyl	1	1925	1458	1	3	2
6144	10	Aluminum / Vinyl	1	1923	962	1	3	1
6145	10	Brick	1	1952	1049	1	2	1
6146	10	Brick	1	1952	1035	1	2	1

6147	10	Aluminum / Vinyl	1	1953	704	1	2	1
6148	10	Stucco	1.5	1921	2090	1	5	1
6149	10	Frame	1	1918	1152	1	3	2
6150	10	Frame	1	1925	748	1	2	1
6151	11	Aluminum / Vinyl	1	1941	1448	1	4	1
6152	11	Aluminum / Vinyl	1	1925	1026	1	2	1
6153	11	Brick	1	1970	1484	1	3	1
6155	11	Brick	1	1956	1382	1	2	1
6156	11	Brick	1	1958	1191	1	3	1
6157	11	Aluminum / Vinyl	1	1960	1186	1	3	1
6158	11	Aluminum / Vinyl	1	1958	1079	1	3	1
6159	11	Aluminum / Vinyl	1	1960	946	1	2	1
6160	11	Brick	1	1956	941	1	3	1
6161	11	Stone	1	1939	924	1	2	1
6162	11	Aluminum / Vinyl	1	1949	726	1	2	1
6163	12	Aluminum / Vinyl	1	1903	1575	2	5	2
6164	13	Aluminum / Vinyl	1.5	1989	2103	1	4	2
6165	13	Brick	1	1953	1379	1	3	2
6166	13	Brick	1	1949	1217	1	3	1
6167	13	Brick	1	1947	1152	1	3	1
6168	13	Brick	1	1955	1090	1	3	2
6169	13	Aluminum / Vinyl	1	1947	923	1	3	1
6170	13	Masonry / Frame	2	1938	1569	1	3	1
6171	13	Aluminum / Vinyl	2	1959	2293	2	6	2
6172	13	Brick	1.5	1956	2016	2	5	2
6173	13	Aluminum / Vinyl	1.5	1953	1650	2	4	2
6174	13	Aluminum / Vinyl	1	1922	1333	1	2	2
6175	13	Masonry / Frame	1	1961	1863	1	3	2
6176	13	Brick	1	1981	1797	1	3	2
6177	13	Aluminum / Vinyl	1	1975	1466	1	3	1
6179	13	Aluminum / Vinyl	1	1954	1194	1	3	1
6180	13	Brick	1	1963	1090	1	3	1
6181	13	Aluminum / Vinyl	1	1969	1087	1	3	1
6182	13	Brick	1	1958	1048	1	3	1
6183	13	Brick	1	1958	998	1	3	1
6184	13	Brick	1	1964	997	1	3	1
6185	13	Aluminum / Vinyl	1	1960	994	1	3	2
6186	13	Aluminum / Vinyl	1	1958	978	1	3	1
6187	13	Stucco	1	1954	936	1	3	1
6188	13	Aluminum / Vinyl	1	1961	907	1	3	1
6189	14	Aluminum / Vinyl	1	1950	1392	1	4	2
6190	14	Aluminum / Vinyl	1	1951	1150	1	4	1
6191	14	Aluminum / Vinyl	1	1952	795	1	2	1

6192	14	Aluminum / Vinyl	2	1950	1512	1	3	1
6193	14	Brick	2	1948	1391	1	3	1
6194	14	Aluminum / Vinyl	2	1948	979	1	2	1
6195	14	Aluminum / Vinyl	1	1929	747	1	2	1
6196	14	Aluminum / Vinyl	2	1908	3403	2	6	4
6197	14	Aluminum / Vinyl	2	1893	2648	2	4	2
6198	14	Aluminum / Vinyl	2	1890	1952	2	3	2
6199	14	Aluminum / Vinyl	1.5	1896	1849	2	4	2
6200	14	Frame	1	1926	1444	1	2	1
6201	14	Brick	1	1929	1340	1	3	1
6202	14	Aluminum / Vinyl	1	1929	1194	1	3	1
6204	15	Aluminum / Vinyl	2	1919	2714	2	6	2
6205	15	Aluminum / Vinyl	1.5	1910	1713	2	3	2
6206	1	Aluminum / Vinyl	1	1951	1213	1	3	1
6207	1	Aluminum / Vinyl	1	1955	1082	1	3	1
6208	2	Brick	1	1954	1512	1	3	1
6209	2	Brick	1	1954	1204	1	4	1
6210	2	Aluminum / Vinyl	2	1957	1984	2	6	2
6211	2	Frame	1	1956	1046	1	3	1
6212	3	Frame	2	1898	3159	1	5	4
6213	3	Aluminum / Vinyl	1.5	1921	1678	2	4	2
6214	3	Brick	2	1953	2080	2	4	2
6215	3	Frame	2	1890	2224	2	4	2
6216	3	Brick	1	1890	1898	2	4	2
6217	3	Stucco	2	1917	6567	1	6	3
6218	3	Brick	2	1915	6143	1	6	3
6219	3	Brick	2	1916	3872	1	4	3
6220	3	Brick	2	1922	3365	1	5	3
6221	3	Aluminum / Vinyl	2	1904	2251	1	4	1
6222	3	Aluminum / Vinyl	2	1900	2068	1	3	2
6223	3	Frame	2	1901	1717	1	4	1
6224	3	Aluminum / Vinyl	1	1903	1465	1	3	2
6225	3	Stucco	1	1922	1371	1	3	2
6226	4	Frame	1	1848	853	1	3	1
6227	4	Aluminum / Vinyl	2	1885	2528	2	4	2
6228	5	Aluminum / Vinyl	1	1948	1226	1	3	2
6229	5	Aluminum / Vinyl	1	1942	1165	1	2	1
6230	5	Brick	1	1949	892	1	2	1
6231	5	Aluminum / Vinyl	1	1952	828	1	2	2
6232	5	Fiber-Cement	2	2002	2452	1	3	2
6233	5	Aluminum / Vinyl	2	1976	1610	1	3	1
6234	5	Aluminum / Vinyl	2	1941	1270	1	3	1
6235	5	Aluminum / Vinyl	2	1947	1205	1	3	1

6236	5	Aluminum / Vinyl	2	1956	2240	2	6	2
6237	5	Aluminum / Vinyl	1	1950	1430	1	3	1
6238	5	Brick	1	1960	1237	1	3	2
6239	5	Aluminum / Vinyl	1	1971	1204	1	4	1
6240	5	Brick	1	1956	1075	1	3	1
6241	5	Aluminum / Vinyl	1	1957	936	1	3	1
6242	7	Frame	2	1924	3280	2	7	2
6243	7	Aluminum / Vinyl	1.5	1925	1822	2	4	2
6244	7	Aluminum / Vinyl	2	1892	2028	2	4	2
6245	7	Brick	1	1929	1584	1	3	1
6246	7	Stone	1.5	1936	1647	1	3	1
6247	8	Aluminum / Vinyl	1.5	1921	1186	1	4	1
6248	8	Stucco	1	1915	1531	1	2	1
6249	8	Aluminum / Vinyl	1.5	1910	1133	1	3	2
6250	9	Aluminum / Vinyl	2	2002	1900	1	3	2
6251	9	Aluminum / Vinyl	1	1980	1609	1	3	1
6252	9	Aluminum / Vinyl	1	1968	1426	1	3	1
6253	9	Aluminum / Vinyl	1	1964	1327	1	3	1
6254	9	Aluminum / Vinyl	1	1969	1232	1	3	1
6255	9	Aluminum / Vinyl	1	1973	1194	1	3	1
6256	9	Brick	1	1956	1042	1	3	1
6257	9	Frame	1	1964	1006	1	3	1
6258	9	Brick	1	1958	953	1	3	1
6259	9	Aluminum / Vinyl	1	1992	1751	1	4	2
6261	10	Aluminum / Vinyl	1	1942	1113	1	3	1
6262	10	Brick	1	1945	1107	1	3	1
6263	10	Brick	2	1924	2334	1	4	1
6264	10	Masonry / Frame	2	1937	1751	1	3	1
6265	10	Aluminum / Vinyl	2	1928	1152	1	3	1
6266	10	Masonry / Frame	1.5	1931	2391	2	5	2
6267	10	Aluminum / Vinyl	1.5	1928	2081	2	4	2
6268	10	Frame	1.5	1924	1914	2	4	2
6269	10	Aluminum / Vinyl	1	1926	1417	1	3	2
6270	10	Aluminum / Vinyl	1	1927	1339	1	3	2
6271	10	Frame	1	1930	1318	1	3	1
6272	10	Aluminum / Vinyl	1	1955	828	1	2	1
6273	10	Aluminum / Vinyl	1	1913	1663	1	4	2
6274	10	Aluminum / Vinyl	1.5	1920	1186	1	3	1
6275	10	Frame	1.5	1912	1122	1	3	1
6276	11	Brick	1.5	1951	1883	1	4	2
6277	11	Brick	1	1941	1487	1	4	1
6278	11	Stone	1	1952	1305	1	3	2
6279	11	Aluminum / Vinyl	1	1942	1095	1	3	1

6280	11	Prem Wood	2	2003	2204	1	4	2
6281	11	Aluminum / Vinyl	1	1958	1519	1	3	1
6282	11	Stone	1	1952	1501	1	3	1
6283	11	Brick	1	1955	1392	1	3	1
6284	11	Brick	1	1957	1240	1	2	2
6285	11	Stone	1	1951	1181	1	2	1
6286	11	Brick	1	1961	1172	1	3	1
6287	11	Aluminum / Vinyl	1	1953	1158	1	3	1
6288	11	Aluminum / Vinyl	1	1966	1157	1	3	2
6289	11	Brick	1	1958	1114	1	3	2
6290	11	Aluminum / Vinyl	1	1954	1073	1	3	1
6291	11	Brick	1	1959	1052	1	2	2
6292	11	Aluminum / Vinyl	1	1957	1023	1	2	2
6293	11	Aluminum / Vinyl	1	1958	1001	1	3	1
6294	11	Frame	1	1953	985	1	3	1
6295	11	Aluminum / Vinyl	1	1955	984	1	3	1
6296	11	Aluminum / Vinyl	1	1960	931	1	3	1
6297	11	Brick	1	1949	862	1	2	1
6298	11	Aluminum / Vinyl	1	1955	793	1	2	1
6299	11	Aluminum / Vinyl	1	1950	776	1	2	1
6300	11	Aluminum / Vinyl	1	1952	759	1	2	1
6301	11	Aluminum / Vinyl	1	1937	740	1	2	1
6302	11	Aluminum / Vinyl	1	1949	726	1	2	1
6303	11	Aluminum / Vinyl	1.5	1932	870	1	3	1
6304	11	Aluminum / Vinyl	2	1966	2409	2	6	2
6305	12	Frame	1	1884	740	1	3	1
6306	12	Block	2	1906	2998	2	6	2
6307	12	Frame	2	1870	1840	2	5	2
6308	12	Aluminum / Vinyl	1	1886	976	1	4	1
6309	13	Brick	1	1949	1398	1	4	2
6310	13	Aluminum / Vinyl	1	1955	1243	1	3	1
6311	13	Aluminum / Vinyl	1	1948	1184	1	4	1
6312	13	Stone	1	1951	1045	1	3	1
6313	13	Stucco	1	1942	1027	1	3	1
6314	13	Brick	1	1957	1025	1	2	1
6315	13	Aluminum / Vinyl	2	1989	2009	1	3	1
6316	13	Brick	1	1955	1304	1	3	2
6317	13	Aluminum / Vinyl	1	1961	1275	1	4	1
6318	13	Aluminum / Vinyl	1	1936	1092	1	3	1
6319	13	Frame	1	1963	1011	1	3	2
6320	13	Aluminum / Vinyl	1	1963	999	1	3	1
6321	13	Brick	1	1956	827	1	2	1
6322	13	Aluminum / Vinyl	1	1951	707	1	2	1

6323	14	Brick	1	1947	1731	1	5	2
6324	14	Brick	1	1948	1153	1	3	2
6325	14	Aluminum / Vinyl	1	1947	1026	1	3	2
6326	14	Aluminum / Vinyl	2	1940	1528	1	3	1
6327	14	Aluminum / Vinyl	1.5	1927	2295	2	4	2
6328	14	Aluminum / Vinyl	2	1880	2106	2	4	2
6329	14	Aluminum / Vinyl	2	1922	1716	2	4	2
6330	14	Brick	1	1929	1461	1	3	1
6331	14	Brick	1	1931	1174	1	3	1
6332	14	Aluminum / Vinyl	1	1955	936	1	3	1
6333	14	Aluminum / Vinyl	1.5	1917	1636	1	4	2
6334	14	Aluminum / Vinyl	2	1911	1478	1	4	1
6335	14	Frame	1.5	1907	1415	1	4	2
6336	14	Aluminum / Vinyl	1.5	1899	1339	1	3	1
6337	1	Block	2	1949	1664	1	3	2
6338	1	Frame	1.5	1925	1342	2	4	2
6339	1	Aluminum / Vinyl	1	1928	1480	1	5	1
6340	1	Masonry / Frame	1	1953	912	1	3	1
6341	1	Aluminum / Vinyl	1	1954	728	1	2	1
6342	1	Frame	1.5	1901	1824	1	3	1
6343	1	Aluminum / Vinyl	1	1929	924	1	3	1
6344	2	Aluminum / Vinyl	1	1947	1509	1	4	2
6345	2	Aluminum / Vinyl	2	1957	2142	2	6	2
6346	2	Brick	1	1954	1176	1	3	1
6347	2	Aluminum / Vinyl	1	1960	1044	1	3	1
6348	2	Aluminum / Vinyl	1	1958	1040	1	3	1
6349	2	Brick	1	1958	1019	1	3	2
6350	2	Aluminum / Vinyl	1	1940	918	1	2	1
6351	3	Stucco	1	1900	1694	1	2	1
6352	3	Aluminum / Vinyl	1.5	1924	1952	2	4	2
6353	3	Aluminum / Vinyl	1	1890	2020	2	4	2
6354	3	Aluminum / Vinyl	2	1922	1988	2	4	2
6355	3	Stone	2	1926	4818	1	6	3
6356	3	Stucco	2	1906	2350	1	5	1
6357	3	Aluminum / Vinyl	1	1918	1450	1	3	1
6358	5	Brick	1	1945	1465	1	3	1
6359	5	Aluminum / Vinyl	1	1952	1288	1	3	2
6360	5	Brick	1	1952	1224	1	3	1
6361	5	Aluminum / Vinyl	1	1947	1221	1	2	2
6362	5	Aluminum / Vinyl	1	1952	1188	1	4	1
6363	5	Aluminum / Vinyl	1	1950	1176	1	4	1
6364	5	Aluminum / Vinyl	1	1950	1166	1	3	1
6365	5	Aluminum / Vinyl	1	1952	1163	1	4	1

6366	5	Aluminum / Vinyl	1	1950	1133	1	3	1
6367	5	Aluminum / Vinyl	1	1954	1037	1	3	1
6368	5	Aluminum / Vinyl	1	1940	1026	1	3	1
6369	5	Aluminum / Vinyl	2	1953	1422	1	3	1
6370	5	Brick	2	1950	2240	2	4	2
6371	5	Aluminum / Vinyl	2	1957	2076	2	6	2
6372	5	Aluminum / Vinyl	1	1954	1904	2	4	3
6373	5	Aluminum / Vinyl	1	1952	1486	1	3	2
6374	5	Aluminum / Vinyl	1	1957	1300	1	3	1
6375	5	Brick	1	1957	1246	1	3	1
6376	5	Brick	1	1960	1208	1	3	1
6377	5	Aluminum / Vinyl	1	1969	1136	1	3	1
6378	5	Brick	1	1956	1082	1	3	2
6379	5	Aluminum / Vinyl	1	1956	1019	1	3	1
6380	5	Aluminum / Vinyl	1	1956	997	1	3	2
6381	5	Aluminum / Vinyl	1	1952	972	1	3	1
6382	5	Aluminum / Vinyl	1	1957	963	1	3	1
6383	5	Frame	1	1957	948	1	2	2
6384	6	Frame	2	1885	3122	2	5	2
6385	6	Aluminum / Vinyl	2	1880	4091	2	6	2
6386	7	Brick	1	1936	1632	1	4	1
6387	7	Brick	2	1929	3074	2	6	2
6388	7	Masonry / Frame	2	1928	2265	2	5	2
6389	7	Stone	1	1949	916	1	2	1
6390	7	Aluminum / Vinyl	1.5	1928	1348	1	3	1
6391	7	Aluminum / Vinyl	1	1923	864	1	3	1
6392	8	Aluminum / Vinyl	2	1910	2666	2	4	2
6393	8	Aluminum / Vinyl	1	1928	1556	1	4	2
6394	8	Aluminum / Vinyl	1	1926	1263	1	3	1
6395	8	Frame	2	1895	1718	1	4	1
6396	9	Brick	1	1958	1616	1	4	2
6397	10	Aluminum / Vinyl	1.5	1951	1575	1	3	2
6398	10	Aluminum / Vinyl	1.5	1939	1573	1	3	1
6399	10	Brick	1	1947	1402	1	3	1
6400	10	Aluminum / Vinyl	1	1953	1260	1	4	1
6401	10	Aluminum / Vinyl	1	1942	1242	1	3	1
6402	10	Brick	1	1941	1199	1	3	1
6403	10	Aluminum / Vinyl	1	1950	1114	1	3	1
6404	10	Brick	2	1936	1869	1	3	1
6405	10	Aluminum / Vinyl	2	1914	1754	1	3	2
6406	10	Aluminum / Vinyl	2	1941	1641	1	3	1
6407	10	Aluminum / Vinyl	2	1921	1237	1	4	1
6408	10	Aluminum / Vinyl	2	1926	2204	2	4	2

6409	10	Aluminum / Vinyl	1.5	1925	1891	1	3	2
6410	10	Aluminum / Vinyl	1	1928	1707	1	4	2
6411	10	Aluminum / Vinyl	1	1919	1617	1	4	2
6412	10	Aluminum / Vinyl	1	1925	1570	1	4	2
6413	10	Aluminum / Vinyl	1	1922	1126	1	3	1
6414	10	Aluminum / Vinyl	1	1942	828	1	2	1
6415	10	Brick	2	1931	2000	1	3	1
6416	11	Aluminum / Vinyl	1.5	1952	2276	1	4	3
6417	11	Aluminum / Vinyl	1	1956	1320	1	4	2
6418	11	Aluminum / Vinyl	1	1953	1198	1	4	1
6419	11	Brick	1	1949	1017	1	3	1
6420	11	Aluminum / Vinyl	2	1959	1378	1	3	1
6421	11	Aluminum / Vinyl	2	1968	1264	1	3	1
6422	11	Brick	1	1929	1865	1	3	1
6423	11	Frame	1	1958	1354	1	3	1
6424	11	Aluminum / Vinyl	1	1954	1210	1	3	1
6425	11	Aluminum / Vinyl	1	1959	1122	1	3	1
6426	11	Aluminum / Vinyl	1	1963	1109	1	3	1
6427	11	Brick	1	1954	1076	1	3	2
6428	11	Aluminum / Vinyl	1	1964	1064	1	3	2
6429	11	Frame	1	1957	1038	1	3	1
6430	11	Brick	1	1954	1013	1	3	1
6431	11	Aluminum / Vinyl	1	1955	988	1	3	1
6432	11	Aluminum / Vinyl	1	1953	922	1	3	1
6433	11	Aluminum / Vinyl	1	1953	906	1	3	1
6434	11	Brick	1	1953	886	1	2	1
6435	11	Aluminum / Vinyl	1	1956	876	1	3	1
6436	11	Brick	2	1931	2200	1	4	1
6437	13	Aluminum / Vinyl	1	1947	1321	1	4	3
6438	13	Block	1	1935	922	1	2	1
6439	13	Aluminum / Vinyl	2	1997	2669	1	5	2
6440	13	Aluminum / Vinyl	1.5	1952	2232	2	4	2
6441	13	Stone	1	1964	1663	1	3	2
6442	13	Brick	1	1960	1510	1	4	2
6443	13	Brick	1	1963	1428	1	3	1
6444	13	Brick	1	1966	1298	1	3	1
6445	13	Aluminum / Vinyl	1	1958	1168	1	3	1
6446	13	Aluminum / Vinyl	1	1953	1067	1	3	1
6447	13	Aluminum / Vinyl	1	1958	1016	1	3	2
6448	13	Brick	1	1955	982	1	3	1
6449	13	Brick	1	1954	784	1	2	1
6450	14	Aluminum / Vinyl	1	1953	1296	1	3	1
6451	14	Aluminum / Vinyl	1	1949	1022	1	3	1

6452	14	Brick	1.5	1926	1777	2	4	2
6453	14	Aluminum / Vinyl	2	1893	2372	2	6	2
6454	14	Aluminum / Vinyl	1.5	1928	1699	1	3	2
6455	14	Frame	1	1924	1381	1	3	1
6456	14	Aluminum / Vinyl	1	1928	1312	1	5	1
6457	14	Aluminum / Vinyl	1	1924	1074	1	3	1
6458	14	Brick	1	1955	840	1	2	1
6459	14	Aluminum / Vinyl	1	1942	792	1	2	1
6460	14	Aluminum / Vinyl	1	1950	725	1	2	1
6461	14	Frame	1.5	1913	1610	1	4	2
6462	14	Aluminum / Vinyl	1.5	1894	1548	1	4	2
6463	14	Frame	2	1913	1544	1	4	1
6464	14	Aluminum / Vinyl	1	1925	1196	1	2	1
6465	14	Aluminum / Vinyl	1	1921	1122	1	3	1
6466	14	Aluminum / Vinyl	1	1924	850	1	1	1
6467	15	Brick	1	1921	2189	1	5	1
6468	1	Aluminum / Vinyl	1.5	1950	1401	1	4	1
6469	1	Aluminum / Vinyl	1.5	1969	1779	2	4	2
6470	1	Brick	1	1927	1507	1	3	1
6471	1	Aluminum / Vinyl	1	1950	672	1	2	1
6472	2	Brick	1	1954	1008	1	2	1
6473	2	Aluminum / Vinyl	2	1965	1970	2	6	2
6474	2	Brick	1	1956	1243	1	3	2
6475	2	Aluminum / Vinyl	1	1957	925	1	3	1
6476	3	Aluminum / Vinyl	2	1908	1496	2	4	2
6477	3	Stucco	2	1920	3037	1	5	3
6478	3	Frame	2	1892	2850	1	4	2
6479	3	Aluminum / Vinyl	2	1885	1235	1	3	1
6481	3	Frame	2	1903	3572	3	8	3
6482	3	Masonry / Frame	2	1922	3282	1	4	3
6483	4	Aluminum / Vinyl	2	1904	2191	2	5	2
6484	4	Aluminum / Vinyl	1.5	1891	2064	1	4	2
6485	5	Fiber-Cement	2	2006	2694	1	4	2
6486	5	Brick	1	1950	1832	1	4	2
6487	5	Stone	1	1952	1550	1	3	1
6488	5	Brick	1	1952	1502	1	3	1
6489	5	Aluminum / Vinyl	1	1949	1338	1	3	2
6490	5	Stone	1	1950	1223	1	2	1
6491	5	Masonry / Frame	2	1952	2027	1	4	2
6492	5	Brick	2	1967	2672	2	6	2
6493	5	Aluminum / Vinyl	1	1951	1619	1	3	1
6494	5	Brick	1	1967	1300	1	3	1
6495	5	Masonry / Frame	1	1957	1216	1	3	1

6496	5	Frame	1	1956	1131	1	3	1
6497	5	Brick	1	1962	1129	1	3	1
6498	5	Aluminum / Vinyl	1	1955	1095	1	3	1
6499	5	Aluminum / Vinyl	1	1950	1062	1	2	1
6500	5	Brick	1	1945	990	1	2	2
6501	5	Aluminum / Vinyl	1	1956	977	1	3	2
6502	5	Aluminum / Vinyl	1	1958	919	1	3	1
6503	5	Aluminum / Vinyl	1	1940	912	1	2	1
6504	5	Aluminum / Vinyl	1	1955	1489	1	3	2
6505	6	Fiber-Cement	1	1900	1524	1	3	2
6506	6	Aluminum / Vinyl	2	1880	2270	2	4	2
6507	6	Frame	1	1863	1652	1	2	2
6508	7	Aluminum / Vinyl	1	1940	1279	1	3	1
6509	7	Aluminum / Vinyl	1	1924	1110	1	2	1
6510	7	Aluminum / Vinyl	1.5	1925	2078	1	5	3
6511	7	Brick	1	1948	1152	1	3	1
6512	7	Aluminum / Vinyl	1	1974	1086	1	3	1
6513	8	Brick	1	1922	1626	1	3	1
6514	8	Aluminum / Vinyl	1	1913	1461	1	3	1
6515	9	Aluminum / Vinyl	1	1974	1695	1	4	2
6516	9	Aluminum / Vinyl	2	2001	1916	1	3	1
6517	9	Aluminum / Vinyl	1	1980	1218	1	3	2
6518	9	Brick	1	1957	1176	1	3	1
6519	9	Aluminum / Vinyl	1	1964	1132	1	3	2
6520	9	Frame	1	1968	1063	1	3	1
6521	9	Aluminum / Vinyl	1	1961	762	1	2	1
6522	10	Brick	1.5	1939	1608	1	3	2
6523	10	Brick	2	1941	1920	1	4	1
6524	10	Stucco	1.5	1921	2080	2	4	2
6525	10	Frame	1.5	1919	2010	2	4	2
6526	10	Aluminum / Vinyl	1.5	1925	1972	2	4	2
6527	10	Aluminum / Vinyl	1.5	1972	2122	2	5	2
6528	10	Aluminum / Vinyl	1	1951	768	1	2	1
6529	10	Aluminum / Vinyl	1	1951	707	1	2	1
6530	11	Stone	1.5	1948	1767	1	4	2
6531	11	Brick	1	1949	1150	1	2	1
6532	11	Frame	2	1919	1724	1	4	2
6533	11	Brick	1	1960	1227	1	3	1
6534	11	Masonry / Frame	1	1952	1216	1	3	1
6535	11	Aluminum / Vinyl	1	1963	1156	1	3	1
6536	11	Brick	1	1960	1016	1	3	1
6537	11	Brick	1	1954	1013	1	3	1
6538	11	Brick	1	1959	957	1	2	1

6539	11	Aluminum / Vinyl	1	1956	919	1	2	1
6540	11	Aluminum / Vinyl	1	1954	867	1	2	1
6541	13	Aluminum / Vinyl	1	1953	1373	1	3	1
6542	13	Aluminum / Vinyl	1	1951	1117	1	3	1
6543	13	Aluminum / Vinyl	2	2003	2271	1	4	3
6544	13	Aluminum / Vinyl	1.5	1967	2158	2	5	3
6545	13	Aluminum / Vinyl	1	1926	1348	1	4	1
6546	13	Aluminum / Vinyl	1	1926	1192	1	3	1
6547	13	Brick	1	1966	1666	1	3	1
6548	13	Brick	1	1974	1644	1	3	1
6549	13	Brick	1	1954	1346	1	3	2
6550	13	Brick	1	1958	1208	1	3	1
6551	13	Brick	1	1958	1074	1	3	2
6552	13	Aluminum / Vinyl	1	1958	1053	1	3	1
6553	13	Aluminum / Vinyl	1	1959	1041	1	3	1
6554	13	Aluminum / Vinyl	1	1915	1424	1	3	2
6555	14	Aluminum / Vinyl	1	1953	1222	1	4	1
6556	14	Stucco	2	1913	1441	1	3	1
6557	14	Frame	2	1947	1378	1	3	1
6558	14	Aluminum / Vinyl	1	1905	842	1	2	1
6559	14	Aluminum / Vinyl	1	1900	711	1	2	2
6560	14	Brick	1.5	1929	2510	2	4	2
6561	14	Brick	>2	1900	4634	2	4	2
6562	14	Aluminum / Vinyl	1.5	1911	1786	2	3	3
6563	14	Aluminum / Vinyl	1	1926	1711	1	5	2
6564	14	Aluminum / Vinyl	1	1923	1617	1	3	2
6565	14	Aluminum / Vinyl	1	1924	1143	1	3	1
6566	14	Aluminum / Vinyl	2	1900	1931	1	4	2
6567	14	Masonry / Frame	2	1935	1793	1	3	1
6568	14	Frame	2	1895	1673	1	4	1
6569	14	Aluminum / Vinyl	1.5	1891	1336	1	3	1
6571	14	Frame	2	1889	2885	3	5	3
6572	1	Aluminum / Vinyl	1.5	1924	1624	2	3	3
6573	1	Aluminum / Vinyl	2	1953	1824	2	6	2
6574	1	Aluminum / Vinyl	1	1927	1426	1	4	1
6575	1	Aluminum / Vinyl	1	1953	768	1	2	1
6576	1	Aluminum / Vinyl	1	1915	1371	1	4	1
6577	2	Brick	1.5	1952	1389	1	4	2
6578	2	Aluminum / Vinyl	1	1980	1198	1	3	2
6579	3	Brick	2	1919	2466	1	4	3
6580	3	Frame	2	1895	1872	2	4	2
6581	3	Fiber-Cement	1.5	1890	1796	2	4	2
6582	3	Frame	2	1910	1642	2	4	2

6583	3	Stucco	2	1912	2624	1	2	2
6584	3	Masonry / Frame	2	1906	2316	1	3	2
6585	3	Aluminum / Vinyl	2	1904	1790	1	4	2
6586	3	Frame	1.5	1891	1669	1	3	2
6587	3	Aluminum / Vinyl	1.5	1901	1427	1	3	2
6588	3	Aluminum / Vinyl	>2	2002	3654	3	>8	3
6589	3	Brick	2	1926	2817	1	5	3
6590	5	Stone	1.5	1949	1772	1	4	2
6591	5	Stone	1	1948	1333	1	3	1
6592	5	Aluminum / Vinyl	1	1952	1243	1	3	1
6593	5	Aluminum / Vinyl	1	1956	1210	1	3	1
6594	5	Aluminum / Vinyl	1	1953	1176	1	3	1
6595	5	Aluminum / Vinyl	1	1950	1171	1	3	1
6596	5	Aluminum / Vinyl	1	1949	1036	1	4	1
6597	5	Aluminum / Vinyl	1	1952	1011	1	3	1
6598	5	Brick	1	1950	847	1	2	1
6599	5	Prem Wood	2	1964	3042	1	4	2
6600	5	Masonry / Frame	2	1948	2448	1	5	3
6601	5	Aluminum / Vinyl	1	1964	1304	1	3	1
6602	5	Aluminum / Vinyl	1	1956	1132	1	3	1
6603	5	Brick	1	1956	1110	1	3	1
6604	5	Aluminum / Vinyl	1	1955	1084	1	3	2
6605	5	Aluminum / Vinyl	1	1956	1014	1	3	2
6606	5	Brick	1	1953	986	1	2	1
6607	5	Aluminum / Vinyl	1	1957	963	1	3	1
6608	5	Aluminum / Vinyl	1	1958	952	1	3	1
6609	5	Aluminum / Vinyl	1	1961	1826	1	3	2
6610	6	Aluminum / Vinyl	1.5	1909	1845	2	4	2
6611	7	Aluminum / Vinyl	1	1940	1581	1	4	1
6612	7	Stone	1	1937	1547	1	4	1
6613	8	Block	1	1951	1522	1	4	1
6614	8	Aluminum / Vinyl	1	1941	1135	1	3	1
6615	8	Aluminum / Vinyl	1.5	1887	1519	1	5	1
6616	8	Aluminum / Vinyl	1	1907	1008	1	3	1
6617	9	Aluminum / Vinyl	2	1961	2088	2	6	2
6618	9	Aluminum / Vinyl	1	1990	1751	1	4	2
6619	10	Stone	1	1940	1416	1	3	2
6620	10	Aluminum / Vinyl	1	1941	1170	1	3	1
6621	10	Aluminum / Vinyl	2	1900	2026	2	4	2
6622	10	Stucco	1	1915	1307	1	3	1
6623	10	Aluminum / Vinyl	1.5	1920	1093	1	3	1
6624	11	Brick	1	1953	1534	1	3	2
6625	11	Brick	1	1941	982	1	2	1

6626	11	Frame	2	1942	1792	1	3	1
6627	11	Masonry / Frame	2	1950	1754	1	3	1
6628	11	Masonry / Frame	2	1940	1470	1	3	1
6629	11	Brick	1.5	1957	1947	2	3	2
6630	11	Brick	1	1979	1717	1	3	2
6631	11	Brick	1	1959	1122	1	3	1
6632	11	Brick	1	1958	1114	1	3	1
6633	11	Aluminum / Vinyl	1	1956	1050	1	3	1
6634	11	Aluminum / Vinyl	1	1959	1033	1	3	1
6635	11	Frame	1	1959	936	1	3	1
6636	12	Aluminum / Vinyl	1.5	1897	1364	1	4	1
6637	12	Frame	2	1895	1455	2	5	2
6638	12	Frame	2	1890	2141	1	3	1
6639	13	Aluminum / Vinyl	1	1947	1348	1	4	2
6640	13	Aluminum / Vinyl	1	1958	1131	1	4	2
6641	13	Stucco	2	1938	1714	1	3	1
6642	13	Aluminum / Vinyl	1.5	1898	1596	2	4	2
6643	13	Aluminum / Vinyl	1	1929	1333	1	3	2
6644	13	Aluminum / Vinyl	1	1974	1361	1	3	1
6645	13	Aluminum / Vinyl	2	1875	1611	1	4	1
6646	13	Aluminum / Vinyl	1	1925	1470	1	3	2
6647	13	Brick	2	1972	2390	2	6	2
6648	14	Aluminum / Vinyl	1	1942	1306	1	3	1
6649	14	Aluminum / Vinyl	1	1946	1204	1	2	1
6650	14	Brick	1	1950	1187	1	3	1
6651	14	Stucco	1	1944	1174	1	4	2
6652	14	Aluminum / Vinyl	2	1909	2538	2	6	2
6653	14	Aluminum / Vinyl	1.5	1921	1782	1	5	2
6654	14	Aluminum / Vinyl	1	1927	1074	1	3	1
6655	14	Brick	1	1954	1716	1	3	2
6656	14	Aluminum / Vinyl	1	1947	826	1	2	1
6657	14	Aluminum / Vinyl	1	1950	735	1	2	1
6658	14	Aluminum / Vinyl	2	1890	2193	1	4	2
6659	14	Frame	1	1918	1117	1	3	1
6660	14	Aluminum / Vinyl	1.5	1922	929	1	2	2
6661	14	Brick	2	1954	1769	2	4	2
6662	15	Aluminum / Vinyl	1.5	1924	2189	2	3	2
6663	15	Frame	2	1890	1964	2	4	2
6664	1	Brick	1	1951	1535	1	4	2
6665	1	Aluminum / Vinyl	1	1943	704	1	2	1
6666	2	Brick	1	1940	1180	1	3	1
6667	2	Aluminum / Vinyl	1	1957	1306	1	4	2
6668	2	Aluminum / Vinyl	1	1964	1254	1	3	1

6669	2	Brick	1	1957	1155	1	3	1
6670	2	Brick	1	1957	942	1	3	1
6671	2	Aluminum / Vinyl	1	1958	936	1	3	1
6672	3	Brick	2	1926	3954	1	4	3
6673	3	Aluminum / Vinyl	2	1899	1961	1	4	1
6674	3	Frame	1	1895	1705	3	3	3
6675	5	Fiber-Cement	2	2003	2703	1	4	4
6676	5	Stone	1	1947	1538	1	3	2
6677	5	Aluminum / Vinyl	1	1952	1444	1	4	1
6678	5	Aluminum / Vinyl	1.5	1956	1194	1	4	2
6679	5	Aluminum / Vinyl	1	1953	1074	1	3	1
6680	5	Aluminum / Vinyl	2	1964	1988	2	6	2
6681	5	Aluminum / Vinyl	1	1956	1324	1	3	1
6682	5	Brick	1	1955	1149	1	3	2
6683	5	Aluminum / Vinyl	1	1954	933	1	3	1
6684	5	Stone	1.5	1936	1552	1	3	1
6685	6	Frame	1	1900	1320	1	2	1
6686	6	Aluminum / Vinyl	1	1884	880	1	2	1
6687	7	Stone	1	1938	1460	1	4	2
6688	7	Stone	2	1935	2007	1	4	1
6689	8	Aluminum / Vinyl	1	1896	1189	1	3	1
6690	9	Aluminum / Vinyl	2	2012	2600	1	4	3
6691	9	Aluminum / Vinyl	1	1980	1505	1	3	1
6692	9	Frame	1	1966	1315	1	3	1
6693	9	Aluminum / Vinyl	1	1958	1217	1	3	1
6694	10	Aluminum / Vinyl	1	1951	998	1	3	1
6695	10	Frame	1	1948	928	1	3	2
6696	10	Brick	2	1914	2698	1	4	2
6697	10	Aluminum / Vinyl	2	1925	2480	2	6	2
6698	10	Aluminum / Vinyl	2	1928	2170	2	4	2
6699	10	Aluminum / Vinyl	1.5	1927	1964	2	4	2
6700	10	Brick	1	1955	2666	2	7	3
6701	10	Stucco	1	1919	2054	1	4	2
6702	10	Aluminum / Vinyl	1	1926	1398	1	3	1
6703	10	Aluminum / Vinyl	2	1908	1886	1	3	1
6704	10	Aluminum / Vinyl	1.5	1924	1445	1	3	1
6705	10	Aluminum / Vinyl	1.5	1926	1169	1	2	2
6706	10	Frame	1	1926	1018	1	3	1
6707	10	Brick	1.5	1914	3040	3	4	3
6708	11	Aluminum / Vinyl	1	1941	1617	1	4	2
6709	11	Stucco	1	1942	1330	1	3	1
6710	11	Aluminum / Vinyl	1	1948	934	1	3	1
6711	11	Aluminum / Vinyl	1.5	1960	2114	2	5	2

6712	11	Aluminum / Vinyl	1	1980	1463	1	3	1
6713	11	Frame	1	1961	1358	1	3	1
6714	11	Aluminum / Vinyl	1	1953	1088	1	3	1
6715	11	Brick	1	1950	1014	1	3	1
6716	11	Aluminum / Vinyl	1	1961	1007	1	3	1
6717	11	Brick	1	1955	924	1	3	1
6718	11	Aluminum / Vinyl	1	1953	906	1	2	1
6719	11	Brick	1	1949	863	1	2	1
6720	11	Stucco	1	1927	1147	1	3	1
6721	11	Masonry / Frame	1	1958	1537	1	3	1
6722	11	Brick	2	1933	1572	1	3	1
6723	12	Aluminum / Vinyl	1	1891	1512	1	6	2
6724	12	Frame	2	1890	2698	1	4	2
6725	12	Aluminum / Vinyl	1	1901	1058	1	2	1
6727	13	Brick	1	1960	1569	1	3	2
6728	13	Frame	1.5	1928	1654	1	4	2
6729	13	Aluminum / Vinyl	1	1950	1305	1	3	1
6730	13	Aluminum / Vinyl	1	1969	1132	1	3	2
6731	13	Aluminum / Vinyl	1	1942	808	1	2	1
6732	14	Aluminum / Vinyl	2	1939	1170	1	2	1
6733	14	Brick	1.5	1926	2154	2	5	2
6734	14	Aluminum / Vinyl	1.5	1908	1667	2	3	3
6735	14	Aluminum / Vinyl	2	1919	1616	2	4	2
6736	14	Aluminum / Vinyl	1	1929	1658	1	3	2
6737	14	Aluminum / Vinyl	1	1928	1209	1	3	1
6738	14	Brick	1	1956	1140	1	3	1
6739	14	Stucco	1	1944	784	1	2	1
6740	14	Aluminum / Vinyl	1	1942	753	1	2	1
6741	14	Aluminum / Vinyl	2	1890	1487	1	3	2
6742	14	Aluminum / Vinyl	1	1888	1320	1	2	1
6743	14	Aluminum / Vinyl	1	1898	1074	1	2	1
6744	14	Aluminum / Vinyl	1	1905	1048	1	3	2
6745	14	Aluminum / Vinyl	1	1905	748	1	2	1
6746	15	Aluminum / Vinyl	2	1902	2650	2	6	2
6747	1	Aluminum / Vinyl	1	1942	1198	1	4	2
6748	1	Aluminum / Vinyl	1	1951	1027	1	4	1
6749	1	Aluminum / Vinyl	2	1981	2143	2	5	2
6750	1	Aluminum / Vinyl	1	1953	1082	1	3	2
6751	2	Brick	1	1938	1395	1	3	2
6752	2	Aluminum / Vinyl	2	1961	1652	1	3	1
6753	2	Aluminum / Vinyl	2	1956	2265	2	6	2
6754	2	Aluminum / Vinyl	2	1940	1762	2	4	2
6755	2	Aluminum / Vinyl	1	1955	1404	1	4	2

6756	2	Aluminum / Vinyl	1	1949	820	1	2	1
6757	2	Aluminum / Vinyl	1	1955	1414	1	3	1
6758	3	Aluminum / Vinyl	2	1922	2771	2	6	2
6759	3	Aluminum / Vinyl	2	1897	2468	2	4	3
6760	3	Aluminum / Vinyl	2	1907	2130	1	5	1
6761	3	Brick	2	1909	3821	1	5	3
6762	3	Brick	2	1927	3234	1	4	3
6763	4	Aluminum / Vinyl	2	1925	2414	2	4	2
6764	4	Brick	>2	1905	8810	1	>8	>4
6765	5	Aluminum / Vinyl	1	1952	1360	1	3	2
6766	5	Aluminum / Vinyl	1	1957	1231	1	4	2
6767	5	Aluminum / Vinyl	1	1950	1152	1	4	1
6768	5	Brick	1	1946	1148	1	3	1
6769	5	Aluminum / Vinyl	1	1952	1092	1	4	2
6770	5	Aluminum / Vinyl	1	1952	1009	1	3	1
6771	5	Aluminum / Vinyl	2	1953	2063	2	6	2
6772	5	Brick	1	1960	1757	1	3	2
6773	5	Brick	1	1957	1340	1	3	1
6774	5	Brick	1	1956	1222	1	3	1
6775	5	Brick	1	1954	1150	1	3	1
6776	5	Aluminum / Vinyl	1	1951	1073	1	3	1
6777	5	Aluminum / Vinyl	1	1956	999	1	3	1
6778	5	Aluminum / Vinyl	1	1955	991	1	3	1
6779	5	Aluminum / Vinyl	1	1958	960	1	3	1
6780	5	Aluminum / Vinyl	1	1950	909	1	3	1
6781	5	Aluminum / Vinyl	1	1951	899	1	2	1
6782	5	Aluminum / Vinyl	1.5	1928	1622	1	3	2
6783	5	Masonry / Frame	2	1983	2641	2	6	2
6784	6	Aluminum / Vinyl	1.5	1909	1983	2	6	2
6785	6	Aluminum / Vinyl	1.5	1904	1549	1	4	1
6786	6	Frame	1	1922	1461	1	2	1
6788	7	Stone	1	1939	1714	1	3	1
6789	7	Brick	1	1927	1804	1	5	2
6790	8	Aluminum / Vinyl	2	1911	2408	2	6	2
6791	8	Aluminum / Vinyl	2	1927	1668	1	3	1
6792	8	Masonry / Frame	2	1909	2714	1	5	1
6793	8	Brick	1	1900	1399	1	3	1
6794	8	Aluminum / Vinyl	2	1924	1249	1	3	1
6795	8	Aluminum / Vinyl	1	1929	1152	1	3	1
6796	8	Aluminum / Vinyl	1	1900	2344	3	5	3
6797	9	Aluminum / Vinyl	2	2008	2224	1	4	2
6798	9	Frame	2	1963	2007	1	5	2
6799	10	Aluminum / Vinyl	1	1951	975	1	3	1

6800	10	Frame	2	1920	2063	1	4	1
6801	10	Aluminum / Vinyl	1	1930	1333	1	3	2
6802	10	Brick	1	1928	1298	1	3	1
6803	10	Aluminum / Vinyl	1	1953	704	1	2	1
6804	10	Aluminum / Vinyl	1	1923	1129	1	4	1
6806	10	Masonry / Frame	2	1933	2514	2	6	2
6807	11	Brick	1	1945	1243	1	3	1
6808	11	Aluminum / Vinyl	1	1953	1103	1	3	1
6809	11	Aluminum / Vinyl	1	1942	1000	1	3	1
6810	11	Aluminum / Vinyl	1	1937	917	1	2	1
6811	11	Prem Wood	2	1990	2238	1	4	2
6812	11	Aluminum / Vinyl	1	1928	984	1	3	1
6813	11	Brick	1	1967	1177	1	3	1
6814	11	Brick	1	1957	1131	1	3	1
6815	11	Aluminum / Vinyl	1	1957	1090	1	3	2
6816	11	Aluminum / Vinyl	1	1960	1077	1	3	1
6817	11	Brick	1	1955	947	1	3	1
6818	11	Aluminum / Vinyl	1	1929	1190	1	4	1
6819	12	Frame	1.5	1900	1810	1	3	2
6820	12	Aluminum / Vinyl	1	1903	1690	2	4	2
6821	13	Stone	1	1940	1251	1	4	2
6822	13	Masonry / Frame	2	1976	3014	1	4	2
6823	13	Aluminum / Vinyl	2	1953	1842	1	3	1
6824	13	Brick	2	1936	1770	1	3	1
6825	13	Brick	1	1965	1491	1	3	1
6826	13	Brick	1	1960	1082	1	3	1
6827	13	Aluminum / Vinyl	1	1936	893	1	3	1
6828	13	Frame	1.5	1900	1231	1	3	2
6829	14	Aluminum / Vinyl	1	1953	1129	1	3	1
6830	14	Aluminum / Vinyl	1.5	1924	1809	2	4	2
6831	14	Aluminum / Vinyl	1.5	1926	1800	2	3	2
6832	14	Aluminum / Vinyl	2	1920	2208	2	4	2
6834	14	Aluminum / Vinyl	1	1905	1130	1	3	1
6835	14	Brick	1	1948	684	1	2	1
6836	14	Frame	1.5	1900	1563	1	3	1
6837	14	Aluminum / Vinyl	1	1910	1003	1	3	1
6838	14	Stucco	1.5	1940	1453	1	3	1
6839	15	Aluminum / Vinyl	2	2012	1854	1	4	2
6840	15	Aluminum / Vinyl	2	1900	2264	2	6	2
6841	15	Frame	2	1892	2464	2	6	2
6842	1	Brick	1	1948	1248	1	3	1
6843	1	Brick	1	1951	1143	1	3	1
6844	1	Aluminum / Vinyl	1	1949	1041	1	3	1

6845	1	Masonry / Frame	2	1940	1684	1	3	1
6846	1	Frame	1	1907	544	1	1	1
6847	1	Aluminum / Vinyl	1	1955	1082	1	3	1
6848	1	Aluminum / Vinyl	1	1955	1082	1	3	1
6849	2	Brick	1	1955	1290	1	4	1
6850	2	Aluminum / Vinyl	1	1956	1286	1	4	1
6851	2	Aluminum / Vinyl	2	1964	1778	1	4	1
6852	2	Frame	1	1966	1237	1	3	1
6853	2	Brick	1	1956	1097	1	3	1
6854	2	Frame	1	1955	970	1	3	1
6855	2	Aluminum / Vinyl	1	1956	905	1	3	2
6856	3	Brick	1	1956	1469	1	3	1
6857	3	Aluminum / Vinyl	1.5	1900	1421	1	2	1
6858	3	Aluminum / Vinyl	2	1910	4435	2	8	3
6859	3	Frame	2	1900	2520	2	6	3
6860	3	Aluminum / Vinyl	2	1902	2112	2	5	2
6861	3	Aluminum / Vinyl	1.5	1900	1456	2	4	2
6862	3	Aluminum / Vinyl	2	1880	1952	2	4	2
6863	3	Brick	2	1911	4113	1	5	4
6864	3	Frame	2	1904	3555	1	5	2
6865	3	Frame	2	1897	3373	1	4	4
6866	3	Stucco	2	1908	2582	1	3	2
6867	3	Aluminum / Vinyl	2	1904	2345	1	4	3
6868	3	Frame	2	1899	2340	1	5	2
6869	3	Brick	1.5	1941	1368	1	2	1
6870	4	Aluminum / Vinyl	2	1977	1253	1	4	1
6871	4	Frame	2	1892	2080	2	4	2
6872	5	Frame	1	1950	1484	1	4	2
6873	5	Brick	1.5	1948	1885	1	4	2
6874	5	Aluminum / Vinyl	1	1941	1238	1	3	2
6875	5	Aluminum / Vinyl	1.5	1950	1231	1	3	1
6876	5	Aluminum / Vinyl	1	1951	1218	1	3	1
6877	5	Brick	1	1951	1133	1	2	1
6878	5	Aluminum / Vinyl	2	1979	2137	1	4	2
6879	5	Aluminum / Vinyl	2	1942	1088	1	2	1
6880	5	Brick	2	1957	2744	2	6	3
6881	5	Aluminum / Vinyl	2	2012	3027	1	3	3
6882	5	Aluminum / Vinyl	1	1969	1569	1	3	1
6883	5	Aluminum / Vinyl	1	1956	1405	1	3	1
6884	5	Brick	1	1952	1316	1	4	1
6885	5	Aluminum / Vinyl	1	1976	1287	1	3	1
6886	5	Brick	1	1953	1108	1	3	1
6887	5	Aluminum / Vinyl	1	1956	1077	1	3	1

6888	5	Brick	1	1949	1053	1	2	1
6889	5	Aluminum / Vinyl	1	1950	950	1	2	1
6890	5	Aluminum / Vinyl	1	1954	940	1	2	2
6891	5	Aluminum / Vinyl	1	1956	864	1	2	1
6892	5	Brick	2	1910	1848	1	3	1
6893	6	Aluminum / Vinyl	1.5	1917	1786	2	4	2
6894	6	Aluminum / Vinyl	2	1899	2374	2	5	2
6895	6	Aluminum / Vinyl	1	1925	1355	1	3	1
6896	6	Aluminum / Vinyl	1	1894	1070	1	3	1
6897	6	Frame	1	1926	833	1	2	1
6898	7	Aluminum / Vinyl	1.5	1940	949	1	2	1
6899	7	Brick	2	1923	3069	2	6	2
6900	7	Brick	1	1937	1737	1	4	1
6901	7	Aluminum / Vinyl	1	1926	1512	1	4	2
6902	8	Aluminum / Vinyl	1	1955	966	1	3	1
6903	8	Aluminum / Vinyl	1.5	1905	1367	1	3	1
6904	9	Aluminum / Vinyl	1	2008	1971	1	3	2
6905	9	Aluminum / Vinyl	1	1980	1505	1	3	1
6906	9	Brick	1	1958	1166	1	3	1
6907	9	Aluminum / Vinyl	1	1957	1110	1	3	2
6908	9	Aluminum / Vinyl	2	1977	2613	2	6	2
6909	9	Aluminum / Vinyl	2	1979	2290	2	6	2
6910	10	Stucco	1	1947	1389	1	4	1
6911	10	Masonry / Frame	1	1952	1335	1	4	2
6912	10	Aluminum / Vinyl	1	1952	1050	1	3	1
6913	10	Aluminum / Vinyl	1	1948	1012	1	3	1
6914	10	Aluminum / Vinyl	1	1948	982	1	3	1
6915	10	Brick	2	1918	3306	1	4	2
6916	10	Brick	2	1935	1692	1	4	2
6917	10	Aluminum / Vinyl	1	1920	680	1	1	1
6918	10	Masonry / Frame	2	1942	1766	2	4	2
6919	10	Frame	2	1957	2132	2	6	2
6920	10	Masonry / Frame	2	1957	1993	2	6	2
6921	10	Aluminum / Vinyl	1	1955	1038	1	3	1
6922	10	Stucco	2	1914	2520	1	4	1
6923	11	Brick	1	1940	1578	1	3	2
6924	11	Aluminum / Vinyl	1	1952	1389	1	3	1
6925	11	Stucco	1	1948	1382	1	3	1
6926	11	Aluminum / Vinyl	1.5	1947	1221	1	4	1
6927	11	Brick	1	1954	1183	1	3	1
6928	11	Aluminum / Vinyl	1	1939	1118	1	3	1
6929	11	Aluminum / Vinyl	1	1949	900	1	3	1
6930	11	Aluminum / Vinyl	2	1922	1410	1	3	1

6931	11	Brick	1.5	1929	1924	2	3	2
6932	11	Brick	1.5	1931	1789	1	3	2
6933	11	Fiber-Cement	1	2002	2344	1	4	3
6934	11	Brick	1	1956	1236	1	4	1
6935	11	Brick	1	1956	1188	1	3	1
6936	11	Aluminum / Vinyl	1	1963	1184	1	3	2
6937	11	Brick	1	1960	1138	1	3	1
6938	11	Brick	1	1949	1037	1	3	1
6939	11	Aluminum / Vinyl	1	1960	1036	1	3	1
6940	11	Aluminum / Vinyl	1	1955	984	1	3	1
6941	11	Aluminum / Vinyl	1	1953	908	1	3	1
6942	12	Aluminum / Vinyl	2	2012	2016	1	4	3
6943	12	Aluminum / Vinyl	1	1881	1260	1	3	1
6944	12	Aluminum / Vinyl	1	1921	1369	1	3	1
6945	12	Aluminum / Vinyl	1	1900	1518	1	3	1
6946	13	Aluminum / Vinyl	1.5	1996	3701	1	5	3
6947	13	Aluminum / Vinyl	1.5	1991	2300	1	3	2
6948	13	Aluminum / Vinyl	1	1939	1149	1	3	1
6949	13	Aluminum / Vinyl	1	1948	1019	1	3	1
6950	13	Aluminum / Vinyl	2	2001	2190	1	3	2
6951	13	Brick	2	1949	1414	1	3	1
6952	13	Aluminum / Vinyl	2	1958	2044	2	6	2
6953	13	Brick	2	1946	2000	2	4	2
6954	13	Brick	1	1959	1276	1	3	1
6955	13	Aluminum / Vinyl	1	1954	1216	1	2	2
6956	13	Brick	1	1961	1036	1	3	1
6957	13	Brick	1	1959	988	1	3	2
6958	13	Frame	1	1952	948	1	2	1
6959	13	Aluminum / Vinyl	1	1959	919	1	3	1
6960	13	Aluminum / Vinyl	1	1954	768	1	3	1
6961	13	Masonry / Frame	1	1959	1841	1	3	2
6962	14	Aluminum / Vinyl	1	1942	1307	1	3	1
6963	14	Aluminum / Vinyl	1	1956	1415	1	4	2
6964	14	Aluminum / Vinyl	1	1900	1406	1	2	2
6965	14	Frame	1.5	1900	1349	1	3	1
6966	14	Aluminum / Vinyl	1.5	1918	1045	1	1	1
6967	14	Aluminum / Vinyl	1.5	1901	1916	2	4	1
6968	14	Aluminum / Vinyl	2	1914	1672	2	4	1
6969	14	Brick	1	1928	1145	1	3	1
6970	14	Aluminum / Vinyl	1	1930	1144	1	3	2
6971	14	Fiber-Cement	1	1924	1132	1	2	1
6972	14	Frame	1	1988	1365	1	3	2
6973	14	Aluminum / Vinyl	1.5	1907	1641	1	3	2

6974	14	Aluminum / Vinyl	1.5	1895	1519	1	4	1
6975	14	Aluminum / Vinyl	1.5	1923	1175	1	3	2
6976	14	Masonry / Frame	1	1962	1921	1	2	1
6977	15	Frame	2	1914	2545	2	6	2
6978	1	Frame	1	1942	1057	1	3	2
6979	1	Aluminum / Vinyl	1	1951	971	1	4	1
6980	1	Brick	2	1953	1680	2	4	2
6981	1	Brick	2	1953	1680	2	4	2
6982	1	Frame	1	1929	1274	1	3	1
6983	1	Aluminum / Vinyl	1	1925	937	1	2	1
6984	2	Aluminum / Vinyl	1	1952	1019	1	3	1
6985	2	Brick	1	1955	1619	1	3	2
6986	2	Aluminum / Vinyl	1	1955	1356	1	3	2
6987	2	Stone	1	1954	1164	1	3	1
6988	2	Brick	1	1958	1139	1	3	1
6989	2	Stone	1	1953	1024	1	3	1
6990	2	Aluminum / Vinyl	1	1960	1002	1	3	2
6991	2	Aluminum / Vinyl	1	1955	900	1	3	2
6992	3	Brick	2	1922	2242	1	4	2
6993	3	Brick	2	1916	2188	1	3	2
6994	3	Aluminum / Vinyl	2	1901	2020	2	5	2
6995	3	Aluminum / Vinyl	2	1908	1496	2	4	2
6996	3	Aluminum / Vinyl	1	1927	1530	1	4	2
6997	3	Brick	2	1912	3973	2	4	3
6998	3	Aluminum / Vinyl	2	1902	3480	1	6	2
6999	3	Aluminum / Vinyl	2	1898	3222	1	6	2
7000	3	Frame	2	1899	2383	1	5	2
7001	3	Fiber-Cement	1.5	1889	2174	1	4	2
7002	3	Frame	2	1905	2141	1	3	2
7003	3	Aluminum / Vinyl	1.5	1928	1736	1	4	2
7004	3	Aluminum / Vinyl	1.5	1903	1510	1	3	2
7005	3	Frame	1	1890	1488	1	3	2
7006	3	Aluminum / Vinyl	2	1888	1378	1	3	1
7007	3	Frame	1.5	1900	1257	1	3	1
7008	3	Aluminum / Vinyl	2	1912	2486	3	5	3
7009	3	Brick	1.5	1914	3791	1	3	>4
7010	4	Aluminum / Vinyl	2	1894	2699	2	6	2
7011	5	Aluminum / Vinyl	1.5	1942	1737	1	4	2
7012	5	Brick	1	1950	1720	1	3	2
7013	5	Aluminum / Vinyl	1.5	1949	1511	1	3	2
7014	5	Brick	1.5	1947	1473	1	4	2
7015	5	Brick	1	1947	1368	1	4	1
7016	5	Brick	1.5	1937	1364	1	3	1

7017	5	Aluminum / Vinyl	1.5	1949	1357	1	4	2
7018	5	Stone	1	1950	1140	1	2	1
7019	5	Aluminum / Vinyl	1	1954	1130	1	4	1
7020	5	Aluminum / Vinyl	1	1952	1092	1	4	2
7021	5	Aluminum / Vinyl	1	1951	1041	1	3	1
7022	5	Aluminum / Vinyl	1	1949	1029	1	4	1
7023	5	Aluminum / Vinyl	1	1952	977	1	3	1
7024	5	Aluminum / Vinyl	2	1986	2069	1	3	2
7025	5	Masonry / Frame	2	2010	1441	1	3	2
7026	5	Masonry / Frame	2	1956	2240	2	6	3
7027	5	Aluminum / Vinyl	1	1971	2048	2	4	4
7028	5	Brick	1	1955	1320	1	3	1
7029	5	Aluminum / Vinyl	1	1962	1255	1	3	1
7030	5	Stone	1	1953	1254	1	2	1
7031	5	Masonry / Frame	1	1957	1247	1	3	1
7032	5	Aluminum / Vinyl	1	1962	1080	1	3	1
7033	5	Brick	1	1956	964	1	3	1
7034	5	Aluminum / Vinyl	1	1953	864	1	3	1
7035	5	Aluminum / Vinyl	1	1950	728	1	2	1
7036	6	Fiber-Cement	2	2007	2685	1	4	2
7037	6	Aluminum / Vinyl	2	2007	1584	1	3	3
7038	6	Aluminum / Vinyl	2	2007	1584	1	3	3
7039	6	Aluminum / Vinyl	1.5	1925	1661	2	4	2
7040	6	Aluminum / Vinyl	2	1913	1976	2	4	2
7041	6	Aluminum / Vinyl	1.5	1905	1960	2	4	2
7042	6	Frame	1.5	1917	1447	2	4	2
7043	6	Aluminum / Vinyl	2	1895	2092	2	5	3
7044	6	Brick	2	1897	3868	1	4	3
7045	6	Aluminum / Vinyl	1	1890	953	1	3	1
7046	7	Masonry / Frame	2	1946	1475	1	3	2
7047	7	Brick	1	1927	1815	1	4	3
7048	7	Stucco	1	1919	1521	1	4	1
7049	7	Aluminum / Vinyl	1	1913	1149	1	3	1
7050	7	Stone	1	1947	1363	1	4	2
7051	7	Aluminum / Vinyl	1	1955	976	1	3	1
7052	8	Frame	1.5	1900	1426	2	3	2
7053	8	Frame	1	1892	1188	2	3	2
7054	8	Stucco	1.5	1914	1267	1	3	2
7055	8	Aluminum / Vinyl	1	1925	947	1	3	1
7056	8	Aluminum / Vinyl	1	1895	1320	1	4	1
7057	9	Aluminum / Vinyl	2	1957	2052	2	6	2
7058	9	Aluminum / Vinyl	1	1968	1422	1	3	1
7059	9	Aluminum / Vinyl	1	1957	1052	1	3	2

7060	10	Brick	1.5	1937	1584	1	3	1
7061	10	Brick	1	1953	1368	1	3	1
7062	10	Aluminum / Vinyl	1	1949	1343	1	4	1
7063	10	Brick	1	1946	1150	1	3	2
7064	10	Aluminum / Vinyl	1	1949	974	1	3	1
7065	10	Frame	2	1925	1588	1	3	1
7066	10	Aluminum / Vinyl	2	1952	1473	1	3	1
7067	10	Stone	2	1940	2595	2	4	2
7068	10	Brick	1.5	1928	2108	2	5	2
7069	10	Aluminum / Vinyl	1.5	1925	1355	2	3	2
7070	10	Stucco	1	1925	1900	1	5	1
7071	10	Brick	1	1927	1807	1	3	1
7072	10	Aluminum / Vinyl	1	1924	1670	1	3	1
7073	10	Brick	1	1927	1593	1	4	1
7074	10	Aluminum / Vinyl	1	1921	1310	1	4	1
7075	10	Aluminum / Vinyl	1	1953	864	1	3	2
7076	10	Aluminum / Vinyl	1	1953	753	1	2	1
7077	10	Aluminum / Vinyl	2	1924	2071	3	2	3
7078	11	Aluminum / Vinyl	1.5	1942	1674	1	4	2
7079	11	Aluminum / Vinyl	1	1949	1257	1	3	1
7080	11	Aluminum / Vinyl	1	1942	1163	1	3	1
7081	11	Aluminum / Vinyl	1	1942	1134	1	3	1
7082	11	Brick	2	1946	1437	1	3	1
7083	11	Aluminum / Vinyl	1	1936	1374	1	2	2
7084	11	Aluminum / Vinyl	1	1956	1151	1	3	1
7085	11	Aluminum / Vinyl	1	1953	1144	1	3	2
7086	11	Brick	1	1955	1062	1	2	1
7087	11	Aluminum / Vinyl	1	1956	1054	1	2	1
7088	11	Brick	1	1958	1052	1	3	1
7089	11	Aluminum / Vinyl	1	1949	1032	1	2	1
7090	11	Brick	1	1956	994	1	3	1
7091	11	Block	1	1949	857	1	2	1
7092	11	Stone	1	1945	806	1	2	1
7093	11	Masonry / Frame	1	1965	2947	1	3	2
7094	12	Aluminum / Vinyl	2	1913	2148	2	6	2
7095	12	Aluminum / Vinyl	2	1914	1854	2	4	2
7096	12	Brick	>2	1894	3001	1	3	2
7098	13	Brick	1	1951	1419	1	3	2
7099	13	Aluminum / Vinyl	1	1948	977	1	3	1
7100	13	Brick	1.5	1931	2151	2	4	2
7101	13	Aluminum / Vinyl	1.5	1926	1950	2	5	2
7102	13	Brick	2	1952	2080	2	4	2
7103	13	Brick	1	1966	1302	1	3	2

7104	13	Brick	1	1960	1201	1	2	2
7105	13	Frame	1	1961	998	1	3	1
7106	13	Brick	1	1969	964	1	2	1
7107	13	Aluminum / Vinyl	1	1959	936	1	3	1
7108	13	Aluminum / Vinyl	1	1939	921	1	2	1
7109	13	Aluminum / Vinyl	1	1939	845	1	2	1
7110	13	Frame	1	1950	818	1	2	1
7111	13	Aluminum / Vinyl	1	1947	771	1	2	1
7112	13	Aluminum / Vinyl	1	1914	828	1	1	1
7113	13	Masonry / Frame	2	1965	2222	2	6	2
7114	14	Aluminum / Vinyl	1	1938	1140	1	3	1
7115	14	Aluminum / Vinyl	2	1939	1212	1	3	1
7116	14	Aluminum / Vinyl	2	1919	2075	2	5	2
7117	14	Brick	2	1956	2160	2	4	2
7118	14	Frame	2	1921	1935	2	4	2
7119	14	Aluminum / Vinyl	1.5	1900	1824	2	3	2
7120	14	Aluminum / Vinyl	1	1944	1624	2	4	2
7121	14	Brick	1	1929	1878	1	4	3
7122	14	Aluminum / Vinyl	1.5	1925	1845	1	3	2
7123	14	Aluminum / Vinyl	1	1925	1518	1	4	1
7124	14	Aluminum / Vinyl	1	1927	1448	1	3	1
7125	14	Aluminum / Vinyl	1	1928	1244	1	3	1
7126	14	Aluminum / Vinyl	1	1926	1172	1	2	1
7127	14	Aluminum / Vinyl	1	1943	707	1	2	1
7128	14	Aluminum / Vinyl	1.5	1891	1623	1	3	2
7129	14	Aluminum / Vinyl	1.5	1896	1441	1	4	2
7130	14	Aluminum / Vinyl	1.5	1920	1415	1	4	2
7131	14	Aluminum / Vinyl	1.5	1900	1205	1	3	1
7132	14	Aluminum / Vinyl	1	1890	1154	1	3	2
7133	14	Aluminum / Vinyl	1	1910	1044	1	3	2
7134	14	Fiber-Cement	1	1900	972	1	2	1
7135	14	Block	2	1915	3186	3	8	3
7136	15	Brick	2	1913	2333	1	5	1
7137	15	Aluminum / Vinyl	1.5	1919	1797	2	4	2
7138	15	Aluminum / Vinyl	1.5	1919	1797	2	4	2
7139	15	Fiber-Cement	1	2010	2008	1	4	2
7140	1	Aluminum / Vinyl	1	1951	1309	1	3	1
7141	1	Aluminum / Vinyl	1.5	1926	1863	2	3	2
7142	1	Fiber-Cement	1	1942	796	1	2	1
7143	2	Aluminum / Vinyl	1	1958	1136	1	3	1
7144	2	Aluminum / Vinyl	1	1955	1070	1	3	1
7145	2	Aluminum / Vinyl	1	1939	1013	1	3	1
7146	2	Stone	2	1948	1847	1	4	1

7147	2	Aluminum / Vinyl	2	1947	1124	1	4	2
7148	2	Frame	1.5	1956	2138	2	5	2
7149	2	Aluminum / Vinyl	1	1964	1306	1	4	1
7150	2	Brick	1	1960	1260	1	3	1
7151	2	Masonry / Frame	1	1961	1246	1	3	1
7152	2	Brick	1	1959	1135	1	3	1
7153	2	Aluminum / Vinyl	1	1957	945	1	3	1
7154	2	Aluminum / Vinyl	1	1959	938	1	3	1
7155	2	Aluminum / Vinyl	1	1957	909	1	3	1
7156	2	Stone	1	1954	842	1	2	2
7157	3	Brick	2	1916	4075	1	5	3
7158	3	Aluminum / Vinyl	2	1924	2432	2	4	2
7159	3	Masonry / Frame	2	1902	6733	1	>8	>4
7160	3	Brick	2	1906	5977	1	7	>4
7161	3	Frame	1.5	1921	2101	1	4	2
7162	3	Brick	1	1924	2080	1	3	2
7163	3	Aluminum / Vinyl	1	1925	1607	1	3	2
7164	3	Brick	2	1912	4304	1	4	3
7165	3	Aluminum / Vinyl	2	1902	1476	1	3	1
7166	3	Aluminum / Vinyl	1	1904	1417	1	3	2
7167	3	Aluminum / Vinyl	1.5	1900	1307	1	3	2
7168	3	Masonry / Frame	2	1926	2388	1	4	2
7169	4	Brick	2	1906	3035	2	6	2
7170	4	Frame	2	1890	3040	1	4	2
7171	5	Aluminum / Vinyl	1	1971	1746	1	4	1
7172	5	Aluminum / Vinyl	1.5	1955	1874	1	4	2
7173	5	Brick	1	1947	1613	1	4	2
7174	5	Stone	1	1952	1560	1	3	2
7175	5	Stone	1	1939	1522	1	3	2
7176	5	Brick	1	1948	1520	1	3	2
7177	5	Brick	1	1951	1511	1	3	1
7178	5	Aluminum / Vinyl	1.5	1949	1485	1	3	2
7179	5	Aluminum / Vinyl	1	1952	1445	1	3	2
7180	5	Brick	1	1951	1442	1	4	2
7181	5	Brick	1	1947	1440	1	3	2
7182	5	Frame	1	1948	1383	1	4	1
7183	5	Brick	1	1949	1372	1	3	1
7184	5	Aluminum / Vinyl	1.5	1950	1269	1	3	2
7185	5	Aluminum / Vinyl	1	1951	1192	1	3	1
7186	5	Brick	1	1951	1169	1	4	1
7187	5	Aluminum / Vinyl	1	1952	1144	1	4	1
7188	5	Aluminum / Vinyl	1	1942	1128	1	4	1
7189	5	Aluminum / Vinyl	1	1948	1051	1	3	1

7190	5	Aluminum / Vinyl	1	1949	1036	1	3	1
7191	5	Aluminum / Vinyl	1	1949	926	1	3	1
7192	5	Brick	1	1947	897	1	2	1
7193	5	Fiber-Cement	2	2005	3214	1	4	3
7194	5	Aluminum / Vinyl	2	1988	1800	1	3	1
7195	5	Brick	2	1949	1650	1	3	1
7196	5	Aluminum / Vinyl	2	1948	1493	1	3	1
7197	5	Stone	1	1952	1856	1	3	1
7198	5	Aluminum / Vinyl	1	1980	1768	1	3	2
7199	5	Aluminum / Vinyl	1	1969	1627	1	3	2
7200	5	Brick	1	1957	1339	1	3	1
7201	5	Brick	1	1956	1338	1	3	2
7202	5	Aluminum / Vinyl	1	1976	1236	1	3	1
7203	5	Brick	1	1952	1204	1	2	1
7204	5	Aluminum / Vinyl	1	1981	1176	1	3	1
7205	5	Brick	1	1961	1126	1	3	1
7206	5	Aluminum / Vinyl	1	1956	1094	1	3	1
7207	5	Aluminum / Vinyl	1	1951	1078	1	3	1
7208	5	Brick	1	1957	1076	1	3	1
7209	5	Brick	1	1959	1050	1	3	2
7210	5	Brick	1	1956	1020	1	3	1
7211	5	Aluminum / Vinyl	1	1952	720	1	2	1
7212	5	Brick	1	1931	1467	1	4	1
7213	5	Brick	1	1961	1727	1	4	2
7214	6	Fiber-Cement	2	2005	2886	1	3	2
7215	6	Aluminum / Vinyl	2	1915	2374	2	4	2
7216	6	Aluminum / Vinyl	2	1906	2256	2	5	2
7217	6	Aluminum / Vinyl	1.5	1892	1868	1	4	2
7218	6	Aluminum / Vinyl	1.5	1898	1720	1	3	1
7219	6	Aluminum / Vinyl	1	1890	1210	1	4	1
7220	6	Aluminum / Vinyl	2	1992	2762	2	6	2
7221	7	Aluminum / Vinyl	1	1949	990	1	3	1
7222	7	Brick	1	1936	1549	1	4	2
7223	7	Brick	1	1936	1549	1	4	2
7224	7	Aluminum / Vinyl	1	1916	1478	1	5	1
7225	8	Aluminum / Vinyl	2	1910	1994	1	4	1
7226	8	Frame	1	1896	1328	1	4	1
7227	8	Frame	1	1896	1328	1	4	1
7228	8	Frame	1	1896	1328	1	4	1
7229	8	Frame	1.5	1901	1186	1	3	1
7230	9	Stucco	2	1981	1354	1	3	1
7231	9	Aluminum / Vinyl	1	1965	1333	1	4	1
7232	10	Stone	1.5	1940	1470	1	3	1

7233	10	Aluminum / Vinyl	1	1950	1217	1	3	1
7234	10	Aluminum / Vinyl	1	1953	1082	1	3	1
7235	10	Aluminum / Vinyl	1	1942	1013	1	3	1
7236	10	Aluminum / Vinyl	1	1947	1010	1	3	1
7237	10	Masonry / Frame	2	1937	1513	1	4	1
7238	10	Aluminum / Vinyl	2	1923	1451	1	3	1
7239	10	Aluminum / Vinyl	2	1926	1365	1	3	1
7240	10	Aluminum / Vinyl	1.5	1924	1736	2	4	2
7241	10	Stone	2	1947	2272	2	4	2
7242	10	Frame	1	1919	1606	1	4	2
7243	10	Aluminum / Vinyl	1	1921	824	1	3	1
7244	10	Aluminum / Vinyl	1	1915	760	1	2	1
7245	11	Aluminum / Vinyl	1	1950	1748	1	4	2
7246	11	Brick	1.5	1953	1629	1	3	1
7247	11	Aluminum / Vinyl	1	1951	1353	1	3	1
7248	11	Brick	1.5	1936	1306	1	2	1
7249	11	Brick	1	1951	1209	1	3	1
7250	11	Brick	1	1947	1200	1	3	1
7251	11	Brick	1	1955	1151	1	3	2
7252	11	Aluminum / Vinyl	1.5	1941	1079	1	3	1
7253	11	Aluminum / Vinyl	1	1955	821	1	3	1
7254	11	Aluminum / Vinyl	1.5	1930	1743	2	4	3
7255	11	Aluminum / Vinyl	2	1963	2563	2	6	3
7256	11	Aluminum / Vinyl	1	1925	1338	1	3	2
7257	11	Aluminum / Vinyl	1	1965	1285	1	3	1
7258	11	Brick	1	1959	1270	1	3	1
7259	11	Aluminum / Vinyl	1	1959	1257	1	3	1
7260	11	Brick	1	1961	1127	1	3	1
7261	11	Brick	1	1964	1116	1	3	1
7262	11	Brick	1	1957	1053	1	3	1
7263	11	Stone	1	1940	1035	1	2	1
7264	11	Aluminum / Vinyl	1	1964	1025	1	3	1
7265	11	Aluminum / Vinyl	1	1955	984	1	3	1
7266	11	Aluminum / Vinyl	1	1962	912	1	3	1
7267	11	Aluminum / Vinyl	1	1953	888	1	3	1
7268	11	Aluminum / Vinyl	1	1952	884	1	3	1
7269	11	Aluminum / Vinyl	1	1953	879	1	3	1
7270	11	Masonry / Frame	1	1958	2210	1	4	2
7271	11	Brick	2	1958	2760	3	5	3
7272	11	Stone	2	1936	1674	1	3	1
7273	12	Frame	1	1895	1330	2	5	2
7274	12	Brick	1	1924	1268	1	4	1
7275	12	Aluminum / Vinyl	1	1925	1075	1	3	1

7276	13	Aluminum / Vinyl	1	1953	1176	1	3	1
7277	13	Aluminum / Vinyl	1	1948	1165	1	3	1
7278	13	Brick	1	1946	1116	1	4	1
7279	13	Aluminum / Vinyl	1	1947	1072	1	3	1
7280	13	Block	1.5	1937	1003	1	2	2
7281	13	Brick	2	1939	1859	1	3	1
7282	13	Aluminum / Vinyl	2	1940	2260	2	4	3
7283	13	Brick	1	1960	1267	1	3	1
7284	13	Brick	1	1968	1245	1	3	1
7285	13	Brick	1	1964	1126	1	3	1
7286	13	Aluminum / Vinyl	1	1960	985	1	3	1
7287	13	Aluminum / Vinyl	1	1960	966	1	3	2
7288	13	Aluminum / Vinyl	1	1951	960	1	3	1
7289	13	Aluminum / Vinyl	1	1960	941	1	3	1
7290	13	Brick	1	1954	864	1	3	1
7291	13	Aluminum / Vinyl	1	1947	732	1	2	1
7292	13	Aluminum / Vinyl	1	1948	686	1	2	1
7293	13	Aluminum / Vinyl	1	1912	972	1	3	2
7294	14	Brick	1	1953	1524	1	4	2
7295	14	Stucco	1.5	1940	1501	1	4	1
7296	14	Aluminum / Vinyl	1	1952	1096	1	4	1
7297	14	Aluminum / Vinyl	1	1950	1092	1	3	1
7298	14	Aluminum / Vinyl	2	1948	1248	1	3	1
7299	14	Brick	2	1900	2825	2	5	2
7300	14	Frame	2	1913	1616	2	4	2
7301	14	Frame	2	1913	1616	2	4	2
7302	14	Aluminum / Vinyl	1.5	1921	1404	2	3	2
7303	14	Aluminum / Vinyl	1	1923	1638	1	5	2
7304	14	Brick	1	1927	1297	1	3	1
7305	14	Aluminum / Vinyl	1	1926	924	1	3	1
7306	14	Aluminum / Vinyl	2	1897	2064	1	5	3
7307	14	Frame	2	1918	1514	1	2	1
7308	14	Frame	1	1910	1377	1	4	2
7309	14	Prem Wood	1	1923	1224	1	3	2
7310	14	Brick	2	1929	2008	1	3	1
7311	15	Aluminum / Vinyl	2	1922	2798	2	6	2
7312	15	Stucco	1	1913	1770	1	3	2
7313	15	Aluminum / Vinyl	1	1905	1276	1	5	1
7314	15	Aluminum / Vinyl	1	1890	1101	1	4	1
7315	1	Brick	1	1952	1256	1	4	1
7316	1	Aluminum / Vinyl	1	1958	985	1	3	1
7317	1	Frame	2	1940	1186	1	3	1
7318	1	Frame	1.5	1957	1618	2	4	2

7319	1	Aluminum / Vinyl	1	1954	963	1	3	2
7320	1	Aluminum / Vinyl	1	1940	898	1	3	1
7321	1	Aluminum / Vinyl	1	1952	882	1	2	1
7322	1	Aluminum / Vinyl	1	1950	876	1	2	1
7323	1	Aluminum / Vinyl	1	1951	689	1	2	1
7324	1	Brick	2	1928	2270	1	4	1
7325	2	Masonry / Frame	1.5	1960	1779	1	3	1
7326	2	Brick	1	1956	1242	1	3	1
7327	2	Brick	1	1956	1188	1	3	1
7328	2	Brick	1	1958	1152	1	3	1
7329	2	Aluminum / Vinyl	1	1957	914	1	3	2
7330	2	Aluminum / Vinyl	1	1954	870	1	3	1
7331	2	Aluminum / Vinyl	1	1951	792	1	2	1
7332	3	Brick	2	1924	1773	1	4	1
7333	3	Brick	2	1931	2552	2	4	2
7334	3	Aluminum / Vinyl	2	1923	2464	2	6	2
7335	3	Aluminum / Vinyl	1.5	1925	1838	2	4	2
7336	3	Brick	2	1901	4662	2	7	4
7337	3	Masonry / Frame	2	1905	3297	2	5	2
7338	3	Frame	2	1912	3075	2	7	2
7339	3	Aluminum / Vinyl	2	1910	2107	2	4	3
7340	3	Frame	1.5	1903	1664	2	3	2
7341	3	Aluminum / Vinyl	2	1900	1664	2	4	2
7342	3	Brick	2	1928	4588	1	6	3
7343	3	Brick	1.5	1919	4187	1	5	3
7344	3	Aluminum / Vinyl	1	1945	859	1	2	1
7345	3	Brick	2	1918	3579	1	4	3
7346	3	Masonry / Frame	2	1896	2922	1	4	1
7347	3	Brick	2	1929	2795	1	4	4
7348	3	Prem Wood	2	1885	1858	1	2	2
7349	3	Frame	1	1910	1501	1	3	1
7350	3	Brick	1.5	1921	1424	1	3	1
7351	3	Aluminum / Vinyl	1	1902	1173	1	3	1
7352	3	Aluminum / Vinyl	2	1914	2220	3	4	2
7353	4	Aluminum / Vinyl	1	1970	1104	1	3	1
7354	4	Frame	2	1906	2073	1	3	1
7355	4	Aluminum / Vinyl	1	1892	940	1	2	1
7356	5	Brick	1	1942	1452	1	3	1
7357	5	Brick	1	1954	1448	1	3	1
7358	5	Stone	1	1950	1433	1	4	2
7359	5	Brick	1	1951	1411	1	4	1
7360	5	Brick	1	1949	1328	1	3	1
7361	5	Brick	1	1949	1316	1	3	1

7362	5	Aluminum / Vinyl	1	1952	1225	1	3	2
7363	5	Aluminum / Vinyl	1	1948	1221	1	3	2
7364	5	Aluminum / Vinyl	1	1953	1217	1	4	2
7365	5	Brick	1	1952	1183	1	3	1
7366	5	Brick	1	1949	1156	1	3	1
7367	5	Brick	1	1949	1145	1	2	1
7368	5	Brick	1	1950	1134	1	2	1
7369	5	Aluminum / Vinyl	1	1949	1018	1	3	2
7370	5	Brick	1	1961	2238	2	5	3
7371	5	Brick	1.5	1952	2050	2	5	2
7372	5	Aluminum / Vinyl	1.5	1951	1510	2	4	2
7373	5	Brick	1	1956	1480	1	3	1
7374	5	Brick	1	1957	1335	1	3	1
7375	5	Aluminum / Vinyl	1	1956	1292	1	3	1
7376	5	Brick	1	1968	1290	1	3	1
7377	5	Frame	1	1971	1204	1	4	2
7378	5	Brick	1	1957	1102	1	2	1
7379	5	Masonry / Frame	1	1954	1065	1	3	1
7380	5	Aluminum / Vinyl	1	1955	1064	1	3	2
7381	5	Aluminum / Vinyl	1	1957	1060	1	3	1
7382	5	Aluminum / Vinyl	1	1956	1055	1	3	2
7383	5	Brick	1	1956	1020	1	3	1
7384	5	Aluminum / Vinyl	1	1956	1019	1	3	1
7385	5	Brick	1	1950	1009	1	3	1
7386	5	Aluminum / Vinyl	1	1954	1390	1	3	2
7387	5	Aluminum / Vinyl	1	1949	998	1	3	1
7388	5	Stone	2	1937	1568	1	3	1
7389	6	Aluminum / Vinyl	2	1922	2436	2	4	2
7390	6	Stucco	1.5	1904	1737	2	4	2
7391	6	Aluminum / Vinyl	1.5	1890	1736	2	4	2
7392	6	Aluminum / Vinyl	2	1894	2064	1	4	1
7393	6	Frame	2	1912	2960	3	6	3
7394	7	Stone	1.5	1941	2029	1	3	2
7395	7	Brick	1	1947	1501	1	3	1
7396	7	Aluminum / Vinyl	1	1945	1054	1	3	1
7397	7	Brick	2	1934	2240	1	4	1
7398	7	Stone	1	1950	1262	1	2	1
7399	7	Aluminum / Vinyl	1	1932	1359	1	3	3
7400	8	Stucco	1	1910	600	1	2	1
7401	8	Aluminum / Vinyl	1.5	1907	1969	2	4	2
7402	8	Frame	1.5	1900	1426	2	3	2
7403	8	Aluminum / Vinyl	1.5	1908	1760	1	5	2
7404	8	Brick	1	1920	1644	1	4	2

7405	8	Aluminum / Vinyl	1	1918	1544	1	4	1
7406	9	Aluminum / Vinyl	1	1974	1450	1	3	2
7407	9	Aluminum / Vinyl	2	2010	2044	1	4	2
7408	9	Aluminum / Vinyl	1	1970	1460	1	3	1
7409	9	Aluminum / Vinyl	1	1960	1117	1	3	2
7410	9	Aluminum / Vinyl	1	1968	1110	1	3	1
7411	9	Brick	1	1957	1050	1	3	1
7412	9	Aluminum / Vinyl	2	1970	2423	2	6	2
7413	10	Stone	1	1947	2310	1	3	3
7414	10	Stone	1.5	1936	1663	1	3	1
7415	10	Stone	1	1946	1549	1	3	2
7416	10	Aluminum / Vinyl	1.5	1948	1380	1	4	1
7417	10	Aluminum / Vinyl	1.5	1953	1338	1	3	2
7418	10	Brick	1	1938	1307	1	3	1
7419	10	Aluminum / Vinyl	1	1952	1277	1	4	1
7420	10	Stone	2	1940	2152	1	3	1
7421	10	Masonry / Frame	2	1946	1570	1	3	1
7422	10	Aluminum / Vinyl	2	1925	3714	2	5	2
7423	10	Aluminum / Vinyl	2	1926	2816	2	4	2
7424	10	Brick	2	1929	2771	2	5	2
7425	10	Aluminum / Vinyl	2	1918	2503	2	4	2
7426	10	Aluminum / Vinyl	2	1924	1924	2	4	2
7427	10	Stone	2	1945	2544	2	4	2
7428	10	Masonry / Frame	2	1947	2302	2	4	2
7429	10	Aluminum / Vinyl	1.5	1885	1607	2	4	2
7430	10	Brick	1	1920	2286	1	5	1
7431	10	Frame	1.5	1931	1904	1	3	2
7432	10	Brick	1	1926	1401	1	2	1
7433	10	Aluminum / Vinyl	1	1925	1384	1	3	1
7434	10	Aluminum / Vinyl	1	1954	1232	1	3	1
7435	10	Stucco	1.5	1915	2003	1	4	2
7436	10	Stone	1.5	1934	1992	1	3	1
7437	10	Aluminum / Vinyl	1	1926	1981	1	3	2
7438	10	Brick	1.5	1926	1258	1	2	1
7439	10	Aluminum / Vinyl	1	1929	1132	1	3	1
7440	10	Frame	1	1924	888	1	2	1
7441	10	Stone	1.5	1937	1796	1	3	1
7442	10	Brick	1.5	1930	1618	1	3	1
7443	11	Aluminum / Vinyl	1	1942	1208	1	4	1
7444	11	Brick	1	1940	1186	1	3	1
7445	11	Aluminum / Vinyl	1	1948	998	1	3	1
7446	11	Stone	2	1937	1652	1	3	1
7447	11	Aluminum / Vinyl	2	1974	1345	1	4	1

7448	11	Aluminum / Vinyl	2	1942	1288	1	3	1
7449	11	Brick	1	1954	2350	2	5	2
7450	11	Frame	1.5	1904	1706	2	3	2
7451	11	Stone	1	1949	1363	1	2	1
7452	11	Aluminum / Vinyl	1	1958	1172	1	3	1
7453	11	Aluminum / Vinyl	1	1959	1142	1	3	1
7454	11	Frame	1	1956	1090	1	3	1
7455	11	Aluminum / Vinyl	1	1956	1073	1	3	2
7456	11	Aluminum / Vinyl	1	1953	1064	1	3	2
7457	11	Brick	1	1958	1037	1	3	1
7458	11	Brick	1	1954	999	1	3	2
7459	11	Aluminum / Vinyl	1	1957	971	1	2	1
7460	11	Brick	1	1951	896	1	2	1
7461	11	Aluminum / Vinyl	1	1954	869	1	3	1
7462	11	Aluminum / Vinyl	1	1956	864	1	3	1
7463	11	Aluminum / Vinyl	1	1942	734	1	2	1
7464	11	Aluminum / Vinyl	1	1948	720	1	2	1
7465	11	Brick	2	1958	2768	3	6	3
7466	12	Aluminum / Vinyl	2	1914	2248	2	6	2
7467	13	Brick	1	1950	1503	1	4	2
7468	13	Aluminum / Vinyl	1	1958	1468	1	4	2
7469	13	Aluminum / Vinyl	1	1946	1165	1	3	1
7470	13	Brick	1	1949	1155	1	3	1
7471	13	Stone	2	1939	1749	1	3	1
7472	13	Aluminum / Vinyl	1	1913	1662	1	4	1
7473	13	Brick	1	1968	1748	1	3	1
7474	13	Aluminum / Vinyl	1	1974	1584	1	3	1
7475	13	Brick	1	1960	1284	1	3	1
7476	13	Brick	1	1969	1266	1	3	1
7477	13	Brick	1	1967	1248	1	3	1
7478	13	Aluminum / Vinyl	1	1964	1112	1	3	1
7479	13	Aluminum / Vinyl	1	1952	1001	1	3	1
7480	13	Aluminum / Vinyl	1	1959	1000	1	3	1
7481	13	Aluminum / Vinyl	1	1963	980	1	3	1
7482	13	Aluminum / Vinyl	1	1934	953	1	2	1
7483	13	Aluminum / Vinyl	1	1959	877	1	3	1
7484	13	Aluminum / Vinyl	1	1942	760	1	2	1
7485	13	Aluminum / Vinyl	2	1925	869	1	2	1
7486	14	Aluminum / Vinyl	1.5	1949	1422	1	4	2
7487	14	Aluminum / Vinyl	1	1943	1158	1	3	1
7488	14	Brick	1	1953	1063	1	3	1
7489	14	Aluminum / Vinyl	1	1944	1004	1	3	1
7490	14	Aluminum / Vinyl	1	1947	854	1	2	1

7491	14	Aluminum / Vinyl	2	1940	1426	1	4	2
7492	14	Aluminum / Vinyl	1	1890	1570	1	2	2
7493	14	Aluminum / Vinyl	1	1900	915	1	3	1
7494	14	Aluminum / Vinyl	2	1929	2660	2	5	2
7495	14	Brick	1.5	1946	1450	2	4	2
7496	14	Aluminum / Vinyl	2	1907	1976	2	4	2
7497	14	Frame	1	1908	1805	2	6	2
7498	14	Aluminum / Vinyl	1.5	1928	1741	1	4	1
7499	14	Aluminum / Vinyl	1	1928	1452	1	3	2
7500	14	Aluminum / Vinyl	1	1924	1320	1	4	1
7501	14	Aluminum / Vinyl	1	1928	1230	1	3	2
7502	14	Aluminum / Vinyl	1	1928	1229	1	3	1
7503	14	Aluminum / Vinyl	1	1929	1021	1	2	1
7504	14	Frame	1	1955	1134	1	3	1
7505	14	Aluminum / Vinyl	1	1952	1012	1	3	1
7506	14	Aluminum / Vinyl	1	1949	725	1	2	1
7507	14	Aluminum / Vinyl	1.5	1916	1694	1	4	1
7508	14	Aluminum / Vinyl	1.5	1927	1548	1	4	1
7509	14	Aluminum / Vinyl	1.5	1880	1444	1	3	1
7510	14	Aluminum / Vinyl	1	1913	1433	1	3	2
7511	14	Aluminum / Vinyl	1	1918	1276	1	2	1
7512	14	Aluminum / Vinyl	1	1900	1255	1	2	2
7513	14	Aluminum / Vinyl	1	1883	1218	1	3	1
7514	14	Aluminum / Vinyl	1	1905	1059	1	3	1
7515	14	Frame	1	1890	1052	1	2	1
7516	14	Aluminum / Vinyl	1	1929	1014	1	2	1
7517	14	Aluminum / Vinyl	1	1918	952	1	2	1
7518	15	Fiber-Cement	2	2011	2000	1	3	2
7519	15	Aluminum / Vinyl	2	2013	1854	1	3	2
7520	15	Aluminum / Vinyl	2	1915	2706	2	6	2
7521	15	Frame	2	1912	2172	2	4	2
7522	15	Brick	1	1919	1981	1	4	2
7523	15	Aluminum / Vinyl	1	1919	1699	1	4	2
7524	1	Brick	1	1953	1467	1	4	1
7525	1	Block	2	1945	1662	2	4	2
7526	2	Aluminum / Vinyl	1	1953	1152	1	3	1
7527	2	Aluminum / Vinyl	1	1955	1118	1	3	1
7528	2	Aluminum / Vinyl	1	1955	1114	1	4	1
7529	2	Aluminum / Vinyl	1.5	1963	2258	2	5	2
7530	2	Aluminum / Vinyl	1	1963	1253	1	4	1
7531	2	Brick	1	1957	1232	1	3	1
7532	2	Brick	1	1953	1223	1	3	1
7533	2	Brick	1	1962	1199	1	3	1

7534	2	Frame	1	1970	1120	1	4	1
7535	2	Aluminum / Vinyl	1	1957	909	1	3	1
7536	2	Aluminum / Vinyl	1	1955	1414	1	3	1
7537	3	Aluminum / Vinyl	2	1930	2748	2	6	2
7538	3	Aluminum / Vinyl	2	1925	2685	2	4	2
7539	3	Aluminum / Vinyl	1.5	1924	1890	2	5	2
7540	3	Frame	2	1961	1800	2	4	2
7541	3	Aluminum / Vinyl	2	1914	2382	2	4	2
7542	3	Aluminum / Vinyl	2	1890	2220	2	4	2
7543	3	Aluminum / Vinyl	2	1898	2188	2	4	2
7544	3	Aluminum / Vinyl	1.5	1904	2020	2	4	2
7545	3	Aluminum / Vinyl	1.5	1906	1495	2	4	2
7546	3	Brick	2	1908	3878	1	6	3
7547	3	Brick	2	1915	3666	1	5	3
7548	3	Frame	2	1901	2592	1	3	1
7549	3	Stucco	2	1920	2314	1	3	2
7550	3	Masonry / Frame	2	1908	2195	1	5	2
7551	3	Frame	1.5	1901	2121	1	3	3
7553	3	Frame	1.5	1890	1990	1	4	1
7554	3	Fiber-Cement	2	1890	1674	1	3	1
7555	5	Brick	1	1951	2000	1	3	2
7556	5	Brick	1	1951	1635	1	5	1
7557	5	Aluminum / Vinyl	1.5	1950	1453	1	4	2
7558	5	Brick	1	1950	1421	1	3	1
7559	5	Brick	1.5	1937	1413	1	2	1
7560	5	Brick	1	1948	1392	1	4	1
7561	5	Aluminum / Vinyl	1.5	1940	1351	1	2	1
7562	5	Brick	1	1949	1322	1	3	1
7563	5	Aluminum / Vinyl	1	1955	1316	1	4	1
7564	5	Brick	1	1951	1189	1	2	1
7565	5	Stone	1.5	1947	1188	1	3	1
7566	5	Brick	1	1953	1167	1	2	1
7567	5	Brick	1	1948	1149	1	3	1
7568	5	Brick	1	1952	1125	1	3	1
7569	5	Brick	1	1947	1124	1	3	1
7570	5	Aluminum / Vinyl	1	1952	1080	1	4	2
7571	5	Aluminum / Vinyl	1	1949	1036	1	3	1
7572	5	Aluminum / Vinyl	1	1949	1004	1	3	1
7573	5	Aluminum / Vinyl	1	1953	1003	1	4	2
7574	5	Aluminum / Vinyl	1	1950	1000	1	3	1
7575	5	Aluminum / Vinyl	1	1950	921	1	3	1
7576	5	Aluminum / Vinyl	2	2012	3379	1	4	3
7577	5	Aluminum / Vinyl	1	1952	1486	1	3	1

7578	5	Brick	1	1955	1246	1	3	2
7579	5	Brick	1	1957	1224	1	3	1
7580	5	Aluminum / Vinyl	1	1954	1220	1	2	1
7581	5	Aluminum / Vinyl	1	1960	1202	1	3	1
7582	5	Brick	1	1957	1164	1	3	2
7583	5	Brick	1	1958	1138	1	3	1
7584	5	Aluminum / Vinyl	1	1958	1040	1	3	1
7585	5	Aluminum / Vinyl	1	1950	1032	1	3	1
7586	5	Brick	1	1952	1014	1	3	1
7587	5	Aluminum / Vinyl	1	1955	1008	1	3	1
7588	5	Aluminum / Vinyl	1	1952	998	1	3	1
7589	5	Aluminum / Vinyl	1	1951	992	1	3	1
7590	5	Brick	1	1945	990	1	2	2
7591	5	Aluminum / Vinyl	1	1954	984	1	3	1
7592	5	Aluminum / Vinyl	1	1957	963	1	3	2
7593	5	Aluminum / Vinyl	1	1956	948	1	2	1
7594	5	Aluminum / Vinyl	1	1958	919	1	3	1
7595	5	Frame	1	1948	916	1	2	1
7596	5	Aluminum / Vinyl	1	1950	912	1	3	1
7597	5	Aluminum / Vinyl	1	1950	768	1	2	1
7598	5	Aluminum / Vinyl	1	1950	687	1	2	1
7599	5	Aluminum / Vinyl	1	1950	672	1	2	1
7600	6	Aluminum / Vinyl	1.5	1923	1982	2	4	2
7601	6	Aluminum / Vinyl	1.5	1926	1858	2	4	2
7602	6	Aluminum / Vinyl	1.5	1903	1688	2	4	2
7603	6	Aluminum / Vinyl	1	1926	1315	1	3	1
7604	6	Brick	2	1890	2850	1	3	3
7605	7	Aluminum / Vinyl	1	1940	1198	1	3	1
7606	7	Brick	1.5	1946	1075	1	2	1
7607	7	Aluminum / Vinyl	1	1938	817	1	3	1
7608	8	Brick	2	1913	2150	1	4	1
7609	8	Aluminum / Vinyl	1	1900	1151	1	3	1
7610	8	Aluminum / Vinyl	1.5	1926	1875	2	5	3
7611	8	Aluminum / Vinyl	1.5	1927	1805	2	4	2
7612	8	Aluminum / Vinyl	1	1922	1288	1	4	1
7613	8	Frame	2	1904	2168	1	4	2
7614	8	Frame	2	1913	2024	1	4	1
7615	8	Aluminum / Vinyl	1.5	1914	1481	1	3	1
7616	8	Aluminum / Vinyl	1	1924	1341	1	3	2
7617	8	Aluminum / Vinyl	1	1903	1222	1	3	1
7618	9	Aluminum / Vinyl	2	1969	1791	1	4	2
7619	9	Aluminum / Vinyl	1	2006	1888	1	4	2
7620	9	Aluminum / Vinyl	1	1966	1421	1	3	1

7621	9	Aluminum / Vinyl	1	1963	1205	1	4	2
7622	9	Frame	1	1968	1063	1	3	1
7623	9	Brick	1	1957	1050	1	3	1
7624	9	Aluminum / Vinyl	1	1975	908	1	3	1
7625	10	Stone	1.5	1937	1874	1	4	1
7626	10	Aluminum / Vinyl	1	1942	1591	1	4	1
7627	10	Brick	1.5	1935	1571	1	3	1
7628	10	Stone	1	1935	1490	1	2	1
7629	10	Brick	1	1946	1038	1	3	1
7630	10	Aluminum / Vinyl	1	1953	1015	1	4	1
7631	10	Stone	2	1937	1960	1	4	2
7632	10	Masonry / Frame	2	1937	1951	1	3	1
7633	10	Brick	2	1940	1417	1	3	1
7634	10	Aluminum / Vinyl	2	1951	1396	1	3	1
7635	10	Aluminum / Vinyl	1	1924	700	1	1	1
7636	10	Aluminum / Vinyl	2	1913	2596	2	6	2
7637	10	Aluminum / Vinyl	2	1924	2427	2	4	2
7638	10	Aluminum / Vinyl	2	1921	2328	2	6	2
7639	10	Frame	1.5	1917	1977	2	4	2
7640	10	Aluminum / Vinyl	1	1920	1763	1	4	2
7641	10	Aluminum / Vinyl	1	1920	1746	1	4	2
7642	10	Aluminum / Vinyl	1	1926	1729	1	3	2
7643	10	Aluminum / Vinyl	1	1920	1727	1	4	1
7644	10	Aluminum / Vinyl	1	1925	1683	1	4	2
7645	10	Frame	1	1916	1617	1	4	2
7646	10	Brick	1	1924	1580	1	2	1
7647	10	Aluminum / Vinyl	1	1930	1491	1	3	1
7648	10	Aluminum / Vinyl	1	1930	1491	1	3	1
7649	10	Frame	1	1928	1336	1	3	2
7650	10	Aluminum / Vinyl	1.5	1919	1329	1	3	1
7651	10	Frame	1	1956	1155	1	3	2
7652	10	Aluminum / Vinyl	1	1953	972	1	3	1
7653	10	Frame	1	1949	914	1	1	1
7654	10	Brick	2	1927	1876	1	2	1
7655	10	Stucco	1	1921	1299	1	3	2
7656	10	Frame	1	1910	977	1	3	1
7657	10	Aluminum / Vinyl	1	1926	932	1	3	1
7658	10	Frame	1	1922	812	1	2	1
7659	10	Aluminum / Vinyl	1	1926	697	1	2	1
7660	11	Fiber-Cement	1	2004	1969	1	3	2
7661	11	Stone	1	1940	1645	1	4	1
7662	11	Stone	1.5	1937	1548	1	4	1
7663	11	Brick	1	1947	1476	1	3	2

7664	11	Brick	1	1954	1437	1	4	1
7665	11	Aluminum / Vinyl	1.5	1940	1223	1	3	1
7666	11	Aluminum / Vinyl	1	1949	1086	1	3	2
7667	11	Aluminum / Vinyl	1.5	1963	1960	2	5	3
7668	11	Prem Wood	1	1968	1873	1	3	1
7669	11	Brick	1	1965	1407	1	3	1
7670	11	Aluminum / Vinyl	1	1954	1396	1	2	2
7671	11	Stone	1	1955	1299	1	3	1
7672	11	Aluminum / Vinyl	1	1959	1266	1	3	1
7673	11	Aluminum / Vinyl	1	1963	1150	1	3	1
7674	11	Aluminum / Vinyl	1	1949	1126	1	2	1
7676	11	Brick	1	1963	1124	1	3	1
7677	11	Brick	1	1955	1069	1	3	1
7678	11	Brick	1	1954	1053	1	3	1
7679	11	Stone	1	1947	1042	1	2	1
7680	11	Brick	1	1958	1041	1	3	1
7681	11	Brick	1	1958	1037	1	3	1
7682	11	Aluminum / Vinyl	1	1958	1035	1	3	1
7683	11	Aluminum / Vinyl	1	1964	1028	1	3	1
7684	11	Aluminum / Vinyl	1	1957	994	1	3	2
7685	11	Brick	1	1956	994	1	3	2
7686	11	Aluminum / Vinyl	1	1954	957	1	3	1
7687	11	Aluminum / Vinyl	1	1955	948	1	3	2
7688	11	Aluminum / Vinyl	1	1953	864	1	3	1
7689	11	Brick	1	1954	838	1	2	1
7690	11	Aluminum / Vinyl	1	1950	638	1	2	1
7691	11	Aluminum / Vinyl	2	1968	2172	2	6	2
7692	12	Aluminum / Vinyl	1	1925	1122	1	3	1
7693	12	Aluminum / Vinyl	1.5	1870	1679	1	4	1
7694	13	Aluminum / Vinyl	1	1976	1850	1	4	2
7695	13	Brick	1	1950	1470	1	3	1
7696	13	Aluminum / Vinyl	1	1958	1464	1	5	2
7697	13	Aluminum / Vinyl	1	1951	1407	1	3	1
7698	13	Aluminum / Vinyl	1	1951	1369	1	3	1
7699	13	Frame	1	1940	1251	1	3	2
7700	13	Brick	1	1951	1166	1	3	1
7701	13	Brick	1	1947	1162	1	3	2
7702	13	Brick	1	1950	1134	1	3	1
7703	13	Brick	1	1952	988	1	3	1
7704	13	Stone	2	1938	1433	1	2	1
7705	13	Aluminum / Vinyl	1.5	1928	1888	2	5	2
7706	13	Frame	1	1928	1511	1	5	1
7707	13	Brick	1	1926	1499	1	3	1

7708	13	Stucco	1	1922	1459	1	4	2
7709	13	Aluminum / Vinyl	1	1961	1223	1	3	1
7710	13	Aluminum / Vinyl	1	1961	1201	1	3	2
7711	13	Brick	1	1952	1108	1	2	2
7712	13	Brick	1	1956	1093	1	2	1
7713	13	Aluminum / Vinyl	1	1970	1080	1	4	1
7714	13	Aluminum / Vinyl	1	1959	1067	1	3	1
7715	13	Aluminum / Vinyl	1	1960	1046	1	3	1
7716	13	Aluminum / Vinyl	1	1959	896	1	3	1
7717	13	Aluminum / Vinyl	1	1944	719	1	2	1
7718	13	Aluminum / Vinyl	1	1943	608	1	1	1
7719	14	Brick	1	1952	1262	1	3	1
7720	14	Aluminum / Vinyl	1	1947	1235	1	3	1
7721	14	Aluminum / Vinyl	1	1950	1148	1	3	1
7722	14	Aluminum / Vinyl	1	1942	979	1	3	2
7723	14	Aluminum / Vinyl	2	1948	1378	1	3	1
7724	14	Aluminum / Vinyl	2	1964	1298	1	3	1
7725	14	Aluminum / Vinyl	2	1941	1213	1	2	1
7726	14	Aluminum / Vinyl	2	1945	990	1	2	1
7727	14	Aluminum / Vinyl	1	1898	998	1	2	1
7728	14	Aluminum / Vinyl	2	1927	2320	2	4	2
7729	14	Aluminum / Vinyl	2	1928	2108	2	4	2
7730	14	Brick	2	1958	2306	2	6	2
7731	14	Brick	1.5	1957	2039	2	5	2
7732	14	Aluminum / Vinyl	2	1906	3183	2	6	2
7733	14	Aluminum / Vinyl	2	1890	2703	2	5	3
7734	14	Frame	2	1894	2321	2	4	2
7735	14	Aluminum / Vinyl	1.5	1925	1860	1	4	2
7736	14	Aluminum / Vinyl	1	1926	1849	1	4	2
7737	14	Brick	1	1928	1791	1	3	2
7738	14	Aluminum / Vinyl	1.5	1915	1715	1	3	2
7739	14	Aluminum / Vinyl	1	1923	1617	1	3	2
7740	14	Aluminum / Vinyl	1	1927	1072	1	3	1
7741	14	Aluminum / Vinyl	2	1924	2096	1	4	3
7742	14	Aluminum / Vinyl	1.5	1917	1665	1	3	2
7743	14	Aluminum / Vinyl	2	1901	1450	1	4	2
7744	14	Aluminum / Vinyl	1	1888	1419	1	4	1
7745	14	Aluminum / Vinyl	1	1910	1352	1	3	2
7746	14	Aluminum / Vinyl	2	1905	1342	1	3	1
7747	14	Aluminum / Vinyl	1	1890	1275	1	2	1
7748	14	Frame	1	1886	1200	1	4	1
7749	14	Aluminum / Vinyl	1	1916	1036	1	3	1

```
     Lotsize Sale_date Sale_price   d_3   r_t
```

1	5040	11719	42000	FALSE	FALSE
2	2880	11808	145000	TRUE	TRUE
3	3185	11839	30000	FALSE	FALSE
4	5781	11961	66500	FALSE	FALSE
5	15600	11992	150500	FALSE	FALSE
6	5075	11992	75000	FALSE	FALSE
7	7750	12022	35000	FALSE	FALSE
8	4800	12204	75000	FALSE	FALSE
9	4200	12265	22000	FALSE	FALSE
10	9480	12387	125000	FALSE	FALSE
11	7800	12387	148500	FALSE	FALSE
12	10428	12387	105000	FALSE	FALSE
13	4560	12539	125000	TRUE	FALSE
14	2280	12600	29000	FALSE	FALSE
15	5668	12631	115000	FALSE	FALSE
16	4520	12784	70000	FALSE	FALSE
17	4230	12843	50000	FALSE	FALSE
18	6450	12843	109900	FALSE	FALSE
19	2950	12874	55000	FALSE	FALSE
20	4200	12904	92000	FALSE	FALSE
21	5400	12935	230000	FALSE	TRUE
22	4150	13027	77000	FALSE	FALSE
23	4551	13027	110000	FALSE	FALSE
24	3500	13118	97000	FALSE	FALSE
25	3500	13149	180000	FALSE	FALSE
26	3252	13180	343100	TRUE	TRUE
27	5250	13180	48000	FALSE	FALSE
28	5080	13208	250000	TRUE	TRUE
29	6000	13269	124100	FALSE	FALSE
30	4230	13300	176800	FALSE	FALSE
31	5040	13300	104000	FALSE	FALSE
32	4680	13300	31100	FALSE	FALSE
33	4551	13330	216000	FALSE	TRUE
34	4520	13330	135000	FALSE	FALSE
35	5400	13361	122000	FALSE	FALSE
36	3600	13361	160900	FALSE	FALSE
37	4200	13361	128750	FALSE	FALSE
38	5600	13361	155000	FALSE	TRUE
39	6160	13392	185000	FALSE	TRUE
40	5504	13545	86900	FALSE	FALSE
41	6450	13545	50000	FALSE	FALSE
42	5250	13726	42500	FALSE	FALSE
43	11129	13726	257000	FALSE	FALSE

44	5535	13757	155000	FALSE	FALSE
45	5418	13879	189000	FALSE	FALSE
46	6351	14153	64000	FALSE	FALSE
47	5838	14245	103000	FALSE	FALSE
48	4797	14245	73450	FALSE	FALSE
49	5040	14245	90000	FALSE	FALSE
50	4800	14245	63300	FALSE	FALSE
51	5945	14245	107500	FALSE	FALSE
52	5400	14245	95000	FALSE	FALSE
53	6660	14245	129000	FALSE	FALSE
54	5590	14245	120900	FALSE	FALSE
55	6000	14245	70000	FALSE	FALSE
56	8190	14245	129900	FALSE	FALSE
57	5750	14245	126500	FALSE	FALSE
58	5544	14245	109000	FALSE	FALSE
59	5250	14245	108000	FALSE	FALSE
60	5400	14245	115000	FALSE	FALSE
61	8040	14245	49000	FALSE	FALSE
62	6500	14245	75000	FALSE	FALSE
63	6000	14245	62500	FALSE	FALSE
64	4800	14245	400000	TRUE	TRUE
65	4080	14245	137000	TRUE	TRUE
66	4020	14245	186500	TRUE	TRUE
67	3600	14245	130000	TRUE	TRUE
68	3600	14245	190000	TRUE	TRUE
69	4800	14245	215000	TRUE	TRUE
70	5280	14245	162250	FALSE	FALSE
71	7000	14245	134900	FALSE	FALSE
72	5400	14245	121600	FALSE	FALSE
73	5400	14245	107000	FALSE	FALSE
74	5000	14245	99900	FALSE	FALSE
75	4800	14245	128000	FALSE	FALSE
76	7200	14245	109500	FALSE	FALSE
77	5320	14245	135000	FALSE	FALSE
78	5700	14245	151000	FALSE	TRUE
79	4800	14245	105000	FALSE	FALSE
80	3968	14245	89000	FALSE	FALSE
81	4290	14245	24000	FALSE	FALSE
82	3570	14245	70000	FALSE	FALSE
83	4200	14245	87000	FALSE	FALSE
85	11250	14245	174000	FALSE	FALSE
86	11328	14245	152900	FALSE	FALSE
87	5600	14245	105000	FALSE	FALSE

88	5200	14245	135000	FALSE	FALSE
89	5160	14245	146000	FALSE	FALSE
90	4920	14245	185000	FALSE	FALSE
91	6000	14245	126000	FALSE	FALSE
92	4524	14245	119900	FALSE	FALSE
93	4080	14245	77000	FALSE	FALSE
94	7440	14245	158400	FALSE	FALSE
95	3240	14245	132500	FALSE	FALSE
96	7930	14245	128300	FALSE	FALSE
97	8040	14245	152000	FALSE	TRUE
98	8062	14245	282000	FALSE	FALSE
99	5400	14245	155000	FALSE	FALSE
100	4800	14245	157000	FALSE	FALSE
101	7150	14245	176000	FALSE	FALSE
102	8040	14245	185000	FALSE	FALSE
103	7625	14245	167000	FALSE	FALSE
104	5050	14245	120000	FALSE	FALSE
105	6650	14245	116000	FALSE	FALSE
106	4725	14245	115000	FALSE	FALSE
107	4880	14245	140000	FALSE	FALSE
108	5060	14245	110000	FALSE	FALSE
109	7104	14245	228000	FALSE	FALSE
110	7500	14245	205000	FALSE	FALSE
111	8480	14245	98750	FALSE	FALSE
112	7200	14245	157500	FALSE	FALSE
113	4720	14245	148000	FALSE	FALSE
114	4235	14245	217000	FALSE	TRUE
115	2660	14245	122500	FALSE	FALSE
117	4290	14245	230000	FALSE	TRUE
118	1710	14245	95000	FALSE	FALSE
119	4900	14276	97000	FALSE	FALSE
120	5043	14276	130000	FALSE	FALSE
121	5440	14276	47600	FALSE	FALSE
122	5400	14276	68000	FALSE	FALSE
123	4800	14276	89000	FALSE	FALSE
124	4800	14276	90000	FALSE	FALSE
125	4800	14276	110000	FALSE	FALSE
126	6350	14276	114900	FALSE	FALSE
127	8100	14276	90000	FALSE	FALSE
128	9648	14276	90500	FALSE	FALSE
129	5520	14276	174200	TRUE	FALSE
130	5715	14276	324000	TRUE	TRUE
131	6000	14276	590000	TRUE	TRUE

132	4800	14276	163500	FALSE	FALSE
133	4800	14276	183000	FALSE	TRUE
134	7200	14276	146000	FALSE	FALSE
135	5000	14276	141600	FALSE	TRUE
136	6700	14276	79800	FALSE	FALSE
137	4800	14276	142650	FALSE	FALSE
138	5400	14276	125000	FALSE	FALSE
139	6720	14276	129800	FALSE	FALSE
140	5508	14276	137000	FALSE	FALSE
141	8990	14276	130000	FALSE	FALSE
142	6000	14276	110000	FALSE	FALSE
143	5960	14276	174800	FALSE	FALSE
144	5125	14276	6678	FALSE	FALSE
145	6050	14276	37500	FALSE	FALSE
146	4250	14276	74516	FALSE	FALSE
147	6500	14276	126500	FALSE	FALSE
148	4551	14276	117500	FALSE	FALSE
149	9507	14276	95000	FALSE	FALSE
150	5000	14276	88000	FALSE	FALSE
151	5808	14276	97000	FALSE	FALSE
152	4800	14276	125000	FALSE	FALSE
153	3795	14276	124600	FALSE	FALSE
154	8910	14276	96000	FALSE	FALSE
155	10650	14276	105000	FALSE	FALSE
156	8100	14276	150000	FALSE	FALSE
157	8400	14276	115000	FALSE	FALSE
158	6000	14276	37000	FALSE	FALSE
159	5400	14276	148900	FALSE	FALSE
160	5400	14276	138900	FALSE	FALSE
161	5600	14276	325000	FALSE	FALSE
162	4680	14276	249980	FALSE	TRUE
163	3588	14276	130000	FALSE	FALSE
164	4960	14276	121500	FALSE	FALSE
165	5355	14276	172000	FALSE	FALSE
166	6500	14276	153900	FALSE	FALSE
167	7540	14276	213000	FALSE	FALSE
168	6760	14276	193000	FALSE	FALSE
169	4920	14276	129000	FALSE	FALSE
170	4920	14276	137000	FALSE	FALSE
171	4760	14276	112000	FALSE	FALSE
172	7038	14276	163300	FALSE	TRUE
173	6375	14276	65000	FALSE	FALSE
174	5635	14276	136000	FALSE	FALSE

175	14300	14276	110000	FALSE	FALSE
176	5040	14276	231000	FALSE	FALSE
177	4270	14276	59000	FALSE	FALSE
178	3500	14276	40000	FALSE	FALSE
179	3696	14276	139900	FALSE	FALSE
180	4800	14276	137500	FALSE	FALSE
181	8050	14276	135000	FALSE	FALSE
182	10395	14276	125000	FALSE	FALSE
183	6095	14276	132000	FALSE	FALSE
184	7436	14276	130000	FALSE	FALSE
185	7209	14276	173000	FALSE	TRUE
186	6579	14276	118000	FALSE	FALSE
187	8100	14276	147000	FALSE	FALSE
188	4000	14276	121000	FALSE	FALSE
189	4305	14276	158500	FALSE	FALSE
190	4800	14276	215400	FALSE	FALSE
191	7200	14276	395000	FALSE	FALSE
192	3900	14276	192500	FALSE	FALSE
193	4410	14276	225000	FALSE	TRUE
194	3510	14276	160000	FALSE	FALSE
195	3200	14276	158000	FALSE	FALSE
196	6400	14276	155000	FALSE	FALSE
197	4720	14276	70000	FALSE	FALSE
199	5000	14276	40000	FALSE	FALSE
200	3600	14276	30000	FALSE	FALSE
201	4960	14304	125000	FALSE	FALSE
202	4840	14304	121400	FALSE	FALSE
203	8967	14304	89375	FALSE	FALSE
204	5250	14304	97000	FALSE	FALSE
205	4800	14304	87600	FALSE	FALSE
206	7245	14304	106400	FALSE	FALSE
207	11970	14304	189000	FALSE	TRUE
208	8040	14304	224999	TRUE	TRUE
209	5535	14304	179875	TRUE	FALSE
210	5080	14304	230000	TRUE	TRUE
211	11700	14304	349200	FALSE	TRUE
212	6784	14304	160000	FALSE	FALSE
213	5120	14304	118000	FALSE	FALSE
214	6550	14304	164900	FALSE	FALSE
215	4800	14304	135000	FALSE	TRUE
216	4960	14304	162000	FALSE	TRUE
217	6700	14304	75500	FALSE	FALSE
218	9020	14304	170000	FALSE	FALSE

219	8700	14304	135000	FALSE	FALSE
220	5280	14304	100000	FALSE	FALSE
221	10125	14304	100000	FALSE	FALSE
222	9042	14304	112250	FALSE	FALSE
223	6500	14304	133800	FALSE	FALSE
224	5080	14304	143500	FALSE	FALSE
225	8060	14304	125000	FALSE	FALSE
226	6459	14304	143000	FALSE	FALSE
227	7015	14304	133100	FALSE	FALSE
228	7080	14304	134000	FALSE	FALSE
229	4800	14304	155000	FALSE	TRUE
230	6050	14304	127000	FALSE	FALSE
231	5000	14304	125000	FALSE	TRUE
232	6776	14304	131500	FALSE	FALSE
233	7245	14304	145000	FALSE	FALSE
234	4700	14304	80000	FALSE	FALSE
235	4800	14304	126000	FALSE	TRUE
236	2340	14304	40000	FALSE	FALSE
237	4200	14304	112900	FALSE	FALSE
238	3960	14304	12500	FALSE	FALSE
239	3960	14304	5000	FALSE	FALSE
240	7080	14304	133000	FALSE	TRUE
241	4480	14304	126000	FALSE	FALSE
242	4800	14304	45000	FALSE	FALSE
243	4800	14304	31000	FALSE	FALSE
244	4800	14304	96000	FALSE	FALSE
245	3600	14304	91000	FALSE	FALSE
246	5715	14304	176400	FALSE	FALSE
247	7742	14304	120000	FALSE	FALSE
248	2400	14304	52100	FALSE	FALSE
249	4200	14304	85000	FALSE	FALSE
250	5625	14304	120000	FALSE	FALSE
251	3600	14304	118900	FALSE	FALSE
252	3998	14304	60000	FALSE	FALSE
253	11335	14304	279500	FALSE	FALSE
254	9000	14304	130000	FALSE	FALSE
255	3800	14304	117500	FALSE	FALSE
256	4216	14304	122000	FALSE	FALSE
257	5360	14304	123500	FALSE	FALSE
258	4800	14304	145000	FALSE	FALSE
259	4520	14304	175000	FALSE	FALSE
260	3630	14304	75000	FALSE	FALSE
261	5400	14304	92000	FALSE	FALSE

262	4800	14304	143900	FALSE	FALSE
263	5640	14304	128000	FALSE	FALSE
264	3600	14304	93000	FALSE	FALSE
265	5000	14304	100000	FALSE	FALSE
266	5160	14304	102000	FALSE	FALSE
267	4800	14304	128000	FALSE	FALSE
268	5400	14304	61000	FALSE	FALSE
269	5500	14304	80000	FALSE	FALSE
270	6360	14304	140000	FALSE	FALSE
271	6000	14304	97500	FALSE	FALSE
272	2640	14304	110000	FALSE	FALSE
273	3720	14304	55000	FALSE	FALSE
274	4200	14304	57000	FALSE	FALSE
275	4760	14304	174000	FALSE	FALSE
276	4800	14304	194000	FALSE	FALSE
277	6000	14304	136000	FALSE	FALSE
278	5170	14304	120000	FALSE	FALSE
279	4920	14304	35000	FALSE	FALSE
280	6837	14304	148000	FALSE	FALSE
281	4840	14304	158000	FALSE	FALSE
282	6840	14304	245000	FALSE	FALSE
283	4800	14304	124000	FALSE	FALSE
284	9000	14304	198000	FALSE	FALSE
285	7110	14304	158000	FALSE	FALSE
286	8280	14304	14700	FALSE	FALSE
287	6578	14304	144000	FALSE	FALSE
288	8580	14304	145500	FALSE	FALSE
289	7728	14304	136000	FALSE	FALSE
290	6450	14304	116000	FALSE	FALSE
291	6000	14304	145000	FALSE	FALSE
292	5535	14304	225000	FALSE	FALSE
293	7062	14304	213500	FALSE	TRUE
294	4720	14304	115000	FALSE	FALSE
295	4960	14304	217900	FALSE	TRUE
296	3720	14304	93000	FALSE	FALSE
297	3600	14304	115000	FALSE	FALSE
298	6165	14304	189000	FALSE	TRUE
299	4800	14304	128000	FALSE	FALSE
300	6210	14304	146900	FALSE	FALSE
301	6550	14304	158900	FALSE	TRUE
302	3510	14304	293500	FALSE	TRUE
303	3078	14304	165000	FALSE	FALSE
304	4290	14304	170000	FALSE	FALSE

305	4720	14304	90000	FALSE	FALSE
306	2970	14304	44800	FALSE	FALSE
307	4920	14304	140000	FALSE	FALSE
308	4800	14335	121300	FALSE	FALSE
309	4800	14335	73000	FALSE	FALSE
310	4800	14335	38000	FALSE	FALSE
311	4356	14335	90000	FALSE	FALSE
312	4920	14335	80000	FALSE	FALSE
313	6600	14335	99900	FALSE	FALSE
314	5805	14335	79026	FALSE	FALSE
315	5265	14335	80000	FALSE	FALSE
316	7526	14335	57000	FALSE	FALSE
317	10680	14335	84500	FALSE	FALSE
318	6018	14335	99900	FALSE	FALSE
319	6250	14335	70450	FALSE	FALSE
320	6500	14335	120100	FALSE	FALSE
321	4920	14335	110000	FALSE	FALSE
322	15030	14335	100000	FALSE	FALSE
323	5000	14335	71000	FALSE	FALSE
324	5080	14335	285000	TRUE	TRUE
325	4800	14335	189000	TRUE	TRUE
326	2640	14335	146000	TRUE	TRUE
327	6960	14335	594225	TRUE	TRUE
328	3600	14335	238000	TRUE	TRUE
329	2670	14335	130000	TRUE	TRUE
330	4800	14335	117000	FALSE	FALSE
331	4800	14335	158000	FALSE	FALSE
332	5000	14335	157500	FALSE	TRUE
333	6500	14335	160000	FALSE	TRUE
334	5280	14335	144875	FALSE	FALSE
335	4800	14335	80000	FALSE	FALSE
336	5200	14335	127500	FALSE	FALSE
337	5320	14335	147400	FALSE	TRUE
338	4116	14335	112500	FALSE	FALSE
339	12600	14335	200000	FALSE	FALSE
340	5700	14335	189900	FALSE	FALSE
341	9525	14335	173000	FALSE	FALSE
342	6250	14335	185000	FALSE	TRUE
343	5160	14335	147000	FALSE	FALSE
344	6400	14335	190000	FALSE	TRUE
345	7809	14335	75000	FALSE	FALSE
346	16768	14335	153500	FALSE	FALSE
347	5000	14335	158000	FALSE	FALSE

348	6000	14335	128000	FALSE	FALSE
349	4800	14335	94000	FALSE	FALSE
350	10166	14335	168000	FALSE	TRUE
351	5610	14335	119900	FALSE	FALSE
352	9000	14335	158500	FALSE	FALSE
353	6240	14335	98000	FALSE	FALSE
354	6240	14335	95000	FALSE	FALSE
355	5400	14335	122500	FALSE	FALSE
356	4200	14335	28000	FALSE	FALSE
357	5400	14335	179000	FALSE	FALSE
358	7038	14335	95000	FALSE	FALSE
359	5125	14335	123000	FALSE	FALSE
360	4920	14335	117000	FALSE	FALSE
361	4551	14335	110000	FALSE	FALSE
362	4920	14335	82000	FALSE	FALSE
363	4800	14335	100000	FALSE	FALSE
364	4324	14335	38500	FALSE	FALSE
365	3600	14335	37000	FALSE	FALSE
366	4165	14335	137000	FALSE	FALSE
368	5500	14335	45000	FALSE	FALSE
369	11456	14335	170000	FALSE	FALSE
370	7717	14335	124000	FALSE	FALSE
371	10125	14335	117000	FALSE	FALSE
372	5560	14335	162000	FALSE	FALSE
373	4320	14335	147500	FALSE	FALSE
374	7080	14335	232000	FALSE	FALSE
375	3900	14335	96500	FALSE	FALSE
376	5400	14335	121000	FALSE	FALSE
377	5280	14335	105000	FALSE	FALSE
378	4800	14335	225000	FALSE	FALSE
379	4050	14335	195000	FALSE	FALSE
380	4294	14335	139000	FALSE	FALSE
381	5400	14335	160000	FALSE	FALSE
382	5200	14335	120000	FALSE	FALSE
383	6542	14335	140000	FALSE	FALSE
384	6075	14335	132957	FALSE	FALSE
385	8400	14335	85000	FALSE	FALSE
386	3870	14335	134000	FALSE	FALSE
387	5360	14335	124500	FALSE	FALSE
388	4332	14335	125000	FALSE	FALSE
389	3660	14335	82900	FALSE	FALSE
390	7564	14335	125000	FALSE	FALSE
391	4800	14335	100000	FALSE	FALSE

392	6750	14335	127000	FALSE	FALSE
393	4950	14335	143000	FALSE	FALSE
394	4838	14335	128000	FALSE	FALSE
395	4920	14335	125000	FALSE	FALSE
396	5651	14335	314000	FALSE	FALSE
397	4800	14335	129000	FALSE	FALSE
398	6700	14335	150000	FALSE	FALSE
399	7638	14335	199000	FALSE	TRUE
400	7560	14335	199000	FALSE	TRUE
401	10315	14335	149000	FALSE	FALSE
402	6750	14335	155300	FALSE	FALSE
403	7260	14335	179300	FALSE	TRUE
404	6000	14335	125000	FALSE	FALSE
405	6000	14335	140000	FALSE	FALSE
406	6240	14335	149500	FALSE	FALSE
407	5750	14335	143000	FALSE	FALSE
408	5940	14335	115000	FALSE	FALSE
409	7488	14335	168000	FALSE	TRUE
410	5400	14335	155500	FALSE	TRUE
411	5280	14335	82500	FALSE	FALSE
412	6000	14335	130000	FALSE	TRUE
413	4800	14335	149000	FALSE	FALSE
414	5400	14335	119500	FALSE	FALSE
415	4884	14335	125000	FALSE	FALSE
416	9900	14335	131000	FALSE	FALSE
417	5625	14335	219000	FALSE	FALSE
418	5400	14335	114000	FALSE	FALSE
419	5074	14335	125000	FALSE	FALSE
420	8820	14335	190000	FALSE	FALSE
421	15014	14335	199000	FALSE	FALSE
422	8235	14335	170000	FALSE	FALSE
423	8228	14335	190000	FALSE	TRUE
424	8060	14335	170000	FALSE	FALSE
425	8400	14335	128500	FALSE	FALSE
426	7440	14335	180000	FALSE	FALSE
427	7480	14335	138000	FALSE	FALSE
428	6450	14335	128500	FALSE	FALSE
429	7800	14335	159500	FALSE	FALSE
430	8280	14335	141000	FALSE	FALSE
431	8350	14335	115000	FALSE	FALSE
432	5633	14335	119000	FALSE	FALSE
433	4578	14335	127500	FALSE	FALSE
434	3600	14335	124500	FALSE	FALSE

435	5000	14335	128600	FALSE	FALSE
436	5040	14335	139000	FALSE	FALSE
437	3502	14335	138500	FALSE	FALSE
438	4636	14335	140000	FALSE	FALSE
439	4514	14335	198000	FALSE	TRUE
440	3182	14335	156200	FALSE	FALSE
441	3750	14335	140000	FALSE	FALSE
442	5035	14335	135000	FALSE	FALSE
443	7340	14335	140000	FALSE	FALSE
444	7200	14335	221900	FALSE	FALSE
445	3600	14335	170000	FALSE	FALSE
446	2881	14335	212500	FALSE	TRUE
447	5250	14365	110000	FALSE	FALSE
448	5777	14365	158000	FALSE	TRUE
449	5355	14365	47000	FALSE	FALSE
450	4800	14365	61000	FALSE	FALSE
451	6350	14365	55000	FALSE	FALSE
452	6000	14365	92000	FALSE	FALSE
453	5440	14365	110000	FALSE	FALSE
454	11475	14365	130000	FALSE	FALSE
455	5940	14365	59000	FALSE	FALSE
456	9600	14365	175000	FALSE	FALSE
457	6480	14365	96000	FALSE	FALSE
458	4920	14365	99000	FALSE	FALSE
459	6000	14365	113900	FALSE	FALSE
460	4800	14365	5000	FALSE	FALSE
461	4800	14365	5000	FALSE	FALSE
462	4800	14365	10000	FALSE	FALSE
463	4800	14365	5000	FALSE	FALSE
464	7980	14365	117000	FALSE	FALSE
465	5200	14365	52500	FALSE	FALSE
466	9625	14365	138600	FALSE	FALSE
467	5310	14365	221000	TRUE	TRUE
468	7050	14365	370000	TRUE	TRUE
469	7050	14365	370000	TRUE	TRUE
470	7050	14365	269900	TRUE	TRUE
471	1980	14365	195000	TRUE	TRUE
472	10800	14365	910000	TRUE	TRUE
473	3540	14365	185000	TRUE	TRUE
474	5640	14365	532450	TRUE	TRUE
475	3210	14365	220000	TRUE	FALSE
476	7680	14365	250000	TRUE	TRUE
477	6650	14365	292500	TRUE	TRUE

478	3600	14365	260000	TRUE	TRUE
479	2071	14365	149900	TRUE	FALSE
480	5080	14365	214500	TRUE	TRUE
481	3600	14365	168000	TRUE	TRUE
482	4782	14365	200000	TRUE	TRUE
483	1902	14365	151700	TRUE	TRUE
484	4800	14365	205000	FALSE	FALSE
485	6120	14365	186500	FALSE	FALSE
486	6016	14365	180000	FALSE	FALSE
487	5280	14365	186500	FALSE	TRUE
488	5400	14365	169800	FALSE	FALSE
489	6480	14365	152000	FALSE	TRUE
490	6000	14365	132000	FALSE	FALSE
491	5120	14365	154900	FALSE	FALSE
492	4816	14365	139900	FALSE	FALSE
493	5400	14365	157900	FALSE	TRUE
494	5000	14365	155000	FALSE	TRUE
495	4960	14365	112900	FALSE	FALSE
496	6250	14365	165000	FALSE	TRUE
497	5080	14365	118000	FALSE	FALSE
498	5106	14365	145000	FALSE	FALSE
499	5120	14365	122000	FALSE	FALSE
500	5106	14365	143000	FALSE	FALSE
501	5000	14365	126000	FALSE	FALSE
502	6204	14365	82500	FALSE	FALSE
503	4800	14365	89900	FALSE	FALSE
504	13200	14365	252000	FALSE	FALSE
505	4800	14365	139900	FALSE	FALSE
506	5000	14365	198500	FALSE	FALSE
507	5120	14365	178000	FALSE	FALSE
508	6500	14365	189000	FALSE	FALSE
509	8664	14365	179900	FALSE	FALSE
510	9515	14365	122000	FALSE	FALSE
511	5120	14365	147000	FALSE	FALSE
512	9535	14365	196500	FALSE	TRUE
513	9882	14365	135500	FALSE	FALSE
514	7296	14365	135340	FALSE	FALSE
515	6660	14365	120000	FALSE	FALSE
516	6750	14365	135000	FALSE	FALSE
517	5120	14365	112000	FALSE	FALSE
518	7040	14365	169900	FALSE	TRUE
519	6840	14365	148900	FALSE	FALSE
520	4823	14365	133000	FALSE	FALSE

521	7920	14365	137000	FALSE	FALSE
522	8450	14365	142000	FALSE	TRUE
523	6750	14365	110000	FALSE	FALSE
524	3236	14365	245000	FALSE	FALSE
525	5080	14365	40000	FALSE	FALSE
526	3280	14365	59550	FALSE	FALSE
527	4800	14365	127000	FALSE	TRUE
528	5760	14365	86000	FALSE	FALSE
529	5160	14365	75000	FALSE	FALSE
530	5160	14365	85000	FALSE	FALSE
531	3795	14365	97000	FALSE	FALSE
532	5494	14365	85000	FALSE	FALSE
533	6890	14365	70900	FALSE	FALSE
534	5160	14365	105900	FALSE	FALSE
535	4429	14365	134000	FALSE	FALSE
536	3960	14365	55000	FALSE	FALSE
537	7920	14365	155000	FALSE	TRUE
538	4200	14365	59500	FALSE	FALSE
539	3570	14365	70000	FALSE	FALSE
540	3558	14365	83900	FALSE	FALSE
541	3750	14365	110000	FALSE	FALSE
542	5850	14365	187500	FALSE	FALSE
543	7920	14365	162000	FALSE	FALSE
544	9000	14365	110450	FALSE	FALSE
545	7200	14365	90000	FALSE	FALSE
546	5160	14365	195000	FALSE	FALSE
547	9000	14365	165000	FALSE	FALSE
548	4920	14365	85000	FALSE	FALSE
549	5208	14365	142500	FALSE	FALSE
550	5460	14365	122500	FALSE	FALSE
551	6175	14365	137000	FALSE	FALSE
552	4320	14365	127500	FALSE	FALSE
553	5400	14365	84000	FALSE	FALSE
554	4800	14365	241500	FALSE	FALSE
555	5085	14365	224500	FALSE	TRUE
556	5400	14365	189000	FALSE	FALSE
557	4816	14365	99900	FALSE	FALSE
558	4080	14365	50000	FALSE	FALSE
559	5160	14365	137900	FALSE	FALSE
560	6480	14365	50100	FALSE	FALSE
561	7011	14365	190000	FALSE	TRUE
562	6950	14365	155000	FALSE	FALSE
563	6149	14365	130000	FALSE	FALSE

564	4800	14365	147000	FALSE	FALSE
565	5120	14365	155500	FALSE	FALSE
566	4800	14365	145000	FALSE	FALSE
567	6700	14365	145000	FALSE	FALSE
568	5000	14365	130000	FALSE	FALSE
570	8092	14365	175000	FALSE	FALSE
571	5040	14365	228000	FALSE	TRUE
572	12000	14365	212000	FALSE	FALSE
573	6000	14365	139800	FALSE	FALSE
574	7590	14365	156500	FALSE	FALSE
575	5520	14365	121000	FALSE	FALSE
576	4960	14365	149500	FALSE	FALSE
577	6625	14365	188000	FALSE	TRUE
578	6072	14365	163000	FALSE	FALSE
579	5040	14365	132000	FALSE	FALSE
580	6650	14365	174000	FALSE	TRUE
581	7654	14365	151000	FALSE	FALSE
582	7198	14365	172000	FALSE	TRUE
583	7440	14365	140000	FALSE	FALSE
584	6240	14365	144000	FALSE	FALSE
585	6579	14365	130000	FALSE	FALSE
586	4884	14365	133000	FALSE	FALSE
587	7524	14365	154000	FALSE	TRUE
588	5050	14365	125000	FALSE	FALSE
589	6405	14365	52000	FALSE	FALSE
590	8576	14365	113000	FALSE	FALSE
591	5985	14365	152500	FALSE	TRUE
592	5400	14365	128500	FALSE	FALSE
593	5985	14365	121500	FALSE	FALSE
594	21780	14365	197000	FALSE	FALSE
595	8640	14365	230000	FALSE	FALSE
596	6450	14365	165000	FALSE	FALSE
598	5074	14365	117400	FALSE	FALSE
599	7000	14365	176000	FALSE	FALSE
600	9450	14365	230000	FALSE	FALSE
601	7236	14365	213000	FALSE	FALSE
602	5805	14365	174900	FALSE	FALSE
603	9750	14365	238500	FALSE	FALSE
604	9296	14365	200000	FALSE	FALSE
605	6095	14365	189900	FALSE	FALSE
606	6450	14365	109200	FALSE	FALSE
607	7427	14365	129000	FALSE	FALSE
608	6500	14365	163000	FALSE	FALSE

609	7200	14365	154900	FALSE	FALSE
610	8418	14365	176000	FALSE	FALSE
611	7290	14365	150000	FALSE	FALSE
612	5980	14365	154500	FALSE	FALSE
613	6474	14365	140000	FALSE	FALSE
615	5160	14365	162900	FALSE	FALSE
616	8268	14365	177000	FALSE	FALSE
617	4277	14365	147750	FALSE	FALSE
618	5490	14365	92500	FALSE	FALSE
619	5822	14365	154500	FALSE	FALSE
620	5625	14365	149900	FALSE	FALSE
621	5160	14365	178000	FALSE	FALSE
622	4050	14365	139000	FALSE	FALSE
623	5850	14365	177000	FALSE	FALSE
624	6320	14365	235000	FALSE	TRUE
625	3510	14365	204000	FALSE	TRUE
626	4340	14365	169000	FALSE	TRUE
627	4514	14365	160000	FALSE	FALSE
628	4712	14365	147500	FALSE	FALSE
629	4640	14365	168500	FALSE	FALSE
630	5246	14365	173800	FALSE	TRUE
631	5250	14365	114000	FALSE	FALSE
632	6204	14365	120000	FALSE	FALSE
633	1750	14365	130000	FALSE	FALSE
634	4463	14365	169500	FALSE	FALSE
635	3600	14365	123000	FALSE	FALSE
636	3600	14365	149900	FALSE	TRUE
637	2400	14365	90000	FALSE	FALSE
638	19769	14396	122000	FALSE	FALSE
639	4800	14396	66500	FALSE	FALSE
640	5500	14396	80000	FALSE	FALSE
641	5125	14396	109900	FALSE	FALSE
642	6000	14396	80000	FALSE	FALSE
643	6400	14396	100000	FALSE	FALSE
644	4960	14396	99900	FALSE	FALSE
645	5360	14396	85000	FALSE	FALSE
646	10960	14396	155000	FALSE	FALSE
647	10960	14396	122600	FALSE	FALSE
648	3600	14396	47000	FALSE	FALSE
649	8840	14396	121900	FALSE	FALSE
650	7560	14396	29000	FALSE	FALSE
651	7863	14396	117000	FALSE	FALSE
652	5375	14396	96000	FALSE	FALSE

653	4920	14396	108000	FALSE	FALSE
654	7200	14396	110000	FALSE	FALSE
655	5330	14396	82400	FALSE	FALSE
656	8450	14396	93000	FALSE	FALSE
657	10105	14396	257400	FALSE	FALSE
658	5330	14396	134000	FALSE	FALSE
659	4920	14396	127650	FALSE	FALSE
660	4800	14396	73500	FALSE	FALSE
661	5400	14396	92500	FALSE	FALSE
662	5240	14396	87500	FALSE	FALSE
663	6480	14396	98000	FALSE	FALSE
664	4920	14396	98000	FALSE	FALSE
665	6050	14396	73000	FALSE	FALSE
666	5500	14396	110000	FALSE	FALSE
667	5460	14396	70000	FALSE	FALSE
668	5640	14396	30000	TRUE	FALSE
669	2325	14396	197900	TRUE	TRUE
670	5400	14396	455000	TRUE	TRUE
671	1800	14396	269000	TRUE	TRUE
672	4200	14396	325000	TRUE	TRUE
673	3660	14396	130000	TRUE	FALSE
675	21780	14396	250000	FALSE	TRUE
676	5400	14396	187000	FALSE	FALSE
677	5000	14396	195000	FALSE	TRUE
678	6250	14396	193500	FALSE	FALSE
679	5428	14396	142500	FALSE	FALSE
680	6528	14396	170000	FALSE	FALSE
681	5000	14396	165000	FALSE	TRUE
682	5160	14396	165000	FALSE	TRUE
683	4800	14396	132500	FALSE	FALSE
684	5355	14396	138000	FALSE	FALSE
685	6050	14396	124500	FALSE	FALSE
686	6000	14396	123000	FALSE	FALSE
687	5610	14396	159000	FALSE	TRUE
688	5120	14396	124000	FALSE	FALSE
689	5120	14396	152400	FALSE	FALSE
690	5400	14396	135000	FALSE	FALSE
691	5040	14396	150000	FALSE	TRUE
692	5170	14396	137500	FALSE	FALSE
693	6000	14396	147700	FALSE	TRUE
694	5000	14396	152900	FALSE	TRUE
695	9000	14396	120000	FALSE	FALSE
696	5280	14396	210000	FALSE	FALSE

697	5117	14396	125000	FALSE	FALSE
698	6000	14396	154000	FALSE	TRUE
699	11072	14396	170000	FALSE	FALSE
700	5106	14396	141300	FALSE	FALSE
701	9000	14396	129900	FALSE	FALSE
702	7320	14396	176500	FALSE	TRUE
703	6500	14396	119000	FALSE	FALSE
704	5248	14396	98000	FALSE	FALSE
705	8710	14396	174000	FALSE	TRUE
706	4800	14396	160000	FALSE	TRUE
707	5000	14396	123000	FALSE	FALSE
708	7380	14396	127000	FALSE	FALSE
709	7830	14396	105000	FALSE	FALSE
710	6250	14396	153000	FALSE	TRUE
711	5900	14396	112000	FALSE	FALSE
712	4800	14396	135000	FALSE	TRUE
713	6625	14396	94900	FALSE	FALSE
714	4428	14396	205000	FALSE	TRUE
715	11550	14396	200500	FALSE	FALSE
716	5996	14396	99000	FALSE	FALSE
717	4250	14396	134600	FALSE	FALSE
718	1700	14396	105000	FALSE	FALSE
720	8800	14396	165000	FALSE	FALSE
721	5600	14396	127000	FALSE	FALSE
722	5760	14396	136800	FALSE	FALSE
723	4956	14396	45000	FALSE	FALSE
724	8448	14396	99000	FALSE	FALSE
725	5000	14396	77500	FALSE	FALSE
726	7360	14396	93000	FALSE	FALSE
727	5160	14396	139900	FALSE	FALSE
728	9680	14396	95000	FALSE	FALSE
729	4800	14396	118000	FALSE	FALSE
730	4420	14396	84900	FALSE	FALSE
731	4200	14396	122000	FALSE	FALSE
732	3300	14396	90000	FALSE	FALSE
733	5355	14396	100000	FALSE	FALSE
734	3240	14396	103500	FALSE	FALSE
735	4375	14396	113250	FALSE	FALSE
736	3750	14396	45000	FALSE	FALSE
737	20064	14396	119000	FALSE	FALSE
738	13000	14396	265000	FALSE	FALSE
739	5940	14396	77000	FALSE	FALSE
740	16335	14396	19900	FALSE	FALSE

741	9656	14396	219900	FALSE	FALSE
742	8085	14396	48800	FALSE	FALSE
743	15150	14396	155000	FALSE	FALSE
744	10150	14396	155000	FALSE	FALSE
745	6000	14396	109500	FALSE	FALSE
746	7236	14396	137000	FALSE	FALSE
747	8166	14396	117500	FALSE	FALSE
748	7200	14396	60500	FALSE	FALSE
749	12600	14396	122000	FALSE	FALSE
750	7208	14396	125900	FALSE	FALSE
751	19835	14396	43000	FALSE	FALSE
752	9563	14396	105000	FALSE	FALSE
753	8604	14396	151000	FALSE	FALSE
754	5500	14396	166900	FALSE	FALSE
755	5400	14396	189900	FALSE	FALSE
756	5600	14396	155000	FALSE	FALSE
757	5950	14396	179000	FALSE	FALSE
758	6900	14396	185000	FALSE	FALSE
759	5850	14396	87500	FALSE	FALSE
760	5040	14396	136500	FALSE	FALSE
761	5400	14396	120000	FALSE	FALSE
762	5760	14396	85000	FALSE	FALSE
763	5460	14396	127000	FALSE	FALSE
764	4662	14396	128000	FALSE	FALSE
765	5688	14396	154900	FALSE	TRUE
766	5200	14396	80000	FALSE	FALSE
767	4524	14396	69000	FALSE	FALSE
768	5080	14396	195000	FALSE	TRUE
769	8650	14396	208000	FALSE	TRUE
770	4800	14396	147000	FALSE	FALSE
771	6500	14396	110000	FALSE	FALSE
772	8008	14396	240000	FALSE	TRUE
773	6650	14396	278000	FALSE	TRUE
774	3520	14396	174000	FALSE	TRUE
775	4800	14396	145000	FALSE	FALSE
776	3720	14396	128500	FALSE	FALSE
777	4655	14396	33500	FALSE	FALSE
778	8100	14396	76500	FALSE	FALSE
779	5400	14396	134900	FALSE	FALSE
780	5200	14396	124900	FALSE	FALSE
781	2680	14396	87000	FALSE	FALSE
782	5400	14396	120000	FALSE	FALSE
783	8760	14396	191100	FALSE	FALSE

784	4800	14396	175000	FALSE	FALSE
785	4800	14396	126000	FALSE	FALSE
786	5440	14396	135000	FALSE	FALSE
787	6000	14396	145500	FALSE	FALSE
788	4920	14396	134900	FALSE	FALSE
789	4840	14396	151000	FALSE	FALSE
790	6500	14396	103000	FALSE	FALSE
791	4840	14396	120000	FALSE	FALSE
792	6144	14396	205000	FALSE	FALSE
793	13562	14396	298000	FALSE	FALSE
794	8520	14396	283000	FALSE	FALSE
795	7980	14396	100000	FALSE	FALSE
796	4836	14396	176000	FALSE	FALSE
797	8576	14396	149900	FALSE	FALSE
798	9432	14396	180000	FALSE	FALSE
799	5680	14396	117600	FALSE	FALSE
800	11456	14396	161700	FALSE	FALSE
801	9200	14396	199800	FALSE	TRUE
802	6600	14396	149400	FALSE	FALSE
803	6681	14396	151500	FALSE	FALSE
804	7260	14396	149000	FALSE	FALSE
805	6604	14396	120000	FALSE	FALSE
806	6000	14396	133000	FALSE	FALSE
807	7480	14396	135000	FALSE	FALSE
808	7420	14396	142000	FALSE	FALSE
809	7198	14396	155900	FALSE	TRUE
810	6480	14396	135000	FALSE	FALSE
811	7040	14396	135000	FALSE	FALSE
812	5289	14396	132000	FALSE	FALSE
813	5040	14396	130000	FALSE	FALSE
814	10836	14396	236500	FALSE	FALSE
815	6600	14396	225700	FALSE	TRUE
816	7440	14396	219900	FALSE	TRUE
817	2820	14396	127100	FALSE	FALSE
818	3600	14396	71000	FALSE	FALSE
819	4200	14396	110000	FALSE	FALSE
820	4340	14396	85000	FALSE	FALSE
821	11952	14396	185000	FALSE	FALSE
822	8245	14396	163000	FALSE	FALSE
823	6150	14396	125500	FALSE	FALSE
824	4200	14396	60000	FALSE	FALSE
825	6120	14396	80000	FALSE	FALSE
826	23205	14396	256000	FALSE	FALSE

827	8556	14396	160000	FALSE	FALSE
828	9594	14396	193000	FALSE	FALSE
829	9380	14396	147000	FALSE	FALSE
830	6150	14396	137000	FALSE	FALSE
831	8260	14396	155000	FALSE	FALSE
832	7740	14396	158000	FALSE	FALSE
833	8220	14396	156000	FALSE	FALSE
834	6900	14396	163500	FALSE	FALSE
835	8491	14396	153000	FALSE	FALSE
836	7866	14396	155000	FALSE	FALSE
837	10200	14396	156900	FALSE	FALSE
838	7260	14396	158000	FALSE	FALSE
839	5805	14396	117900	FALSE	FALSE
840	9050	14396	135000	FALSE	FALSE
841	8000	14396	84500	FALSE	FALSE
842	14310	14396	165000	FALSE	FALSE
843	5040	14396	130400	FALSE	FALSE
844	5040	14396	162500	FALSE	FALSE
845	6674	14396	128900	FALSE	FALSE
846	4800	14396	257500	FALSE	TRUE
847	3840	14396	126500	FALSE	FALSE
848	4880	14396	231000	FALSE	TRUE
849	3120	14396	140000	FALSE	FALSE
850	6435	14396	233000	FALSE	FALSE
851	6050	14396	181000	FALSE	FALSE
852	3458	14396	217000	FALSE	TRUE
853	7200	14396	144200	FALSE	FALSE
854	4200	14396	225000	FALSE	TRUE
855	3600	14396	114000	FALSE	FALSE
856	4800	14396	192900	FALSE	TRUE
857	11031	14396	164000	FALSE	FALSE
858	4620	14396	127000	FALSE	FALSE
859	8256	14396	134500	FALSE	FALSE
860	5400	14396	124000	FALSE	FALSE
861	4800	14396	105000	FALSE	FALSE
862	5400	14396	190000	FALSE	FALSE
863	5120	14396	165000	FALSE	FALSE
864	3600	14396	159000	FALSE	FALSE
865	3156	14396	148000	FALSE	FALSE
866	3870	14396	215000	FALSE	FALSE
867	1900	14396	211500	FALSE	TRUE
868	3510	14396	160000	FALSE	FALSE
869	3000	14396	195000	FALSE	TRUE

870	3480	14396	157000	FALSE	FALSE
871	3750	14396	129900	FALSE	FALSE
872	3600	14396	123000	FALSE	FALSE
873	3600	14396	96500	FALSE	FALSE
874	4800	14396	160000	FALSE	FALSE
875	3660	14396	97400	FALSE	FALSE
876	4720	14396	34500	FALSE	FALSE
877	3600	14396	22525	FALSE	FALSE
878	4720	14396	96000	FALSE	FALSE
879	5000	14396	97000	FALSE	FALSE
880	4800	14426	60000	FALSE	FALSE
881	6600	14426	60000	FALSE	FALSE
882	5875	14426	69000	FALSE	FALSE
883	6000	14426	101000	FALSE	FALSE
884	7317	14426	175000	FALSE	FALSE
885	5885	14426	87500	FALSE	FALSE
886	7750	14426	97900	FALSE	FALSE
887	8905	14426	82500	FALSE	FALSE
888	5200	14426	100000	FALSE	FALSE
889	7800	14426	130000	FALSE	FALSE
890	4920	14426	92000	FALSE	FALSE
891	5900	14426	120000	FALSE	FALSE
892	4800	14426	40000	FALSE	FALSE
893	4800	14426	94500	FALSE	FALSE
894	5500	14426	89900	FALSE	FALSE
895	6000	14426	136000	FALSE	TRUE
896	7300	14426	95000	FALSE	FALSE
897	4875	14426	63900	FALSE	FALSE
898	6120	14426	72900	FALSE	FALSE
899	5400	14426	110000	FALSE	FALSE
900	12137	14426	20000	FALSE	FALSE
901	4800	14426	318000	TRUE	TRUE
902	5640	14426	170000	TRUE	FALSE
903	4340	14426	190000	TRUE	TRUE
904	5120	14426	168900	TRUE	TRUE
905	5400	14426	285000	TRUE	TRUE
906	4800	14426	274000	TRUE	TRUE
907	5400	14426	230000	TRUE	TRUE
908	3600	14426	161000	TRUE	TRUE
910	6600	14426	395000	TRUE	TRUE
911	4800	14426	196900	FALSE	FALSE
912	7800	14426	167650	FALSE	TRUE
913	6625	14426	144000	FALSE	FALSE

914	5805	14426	159900	FALSE	FALSE
915	7000	14426	139000	FALSE	FALSE
916	5760	14426	75000	FALSE	FALSE
917	5080	14426	153000	FALSE	FALSE
918	4800	14426	137000	FALSE	FALSE
919	5240	14426	127000	FALSE	FALSE
920	4920	14426	136500	FALSE	FALSE
921	6250	14426	164900	FALSE	TRUE
922	5248	14426	84500	FALSE	FALSE
923	5000	14426	121500	FALSE	FALSE
924	5760	14426	167900	FALSE	TRUE
925	5000	14426	156500	FALSE	TRUE
926	5250	14426	146900	FALSE	TRUE
927	5390	14426	114000	FALSE	FALSE
928	5500	14426	129000	FALSE	FALSE
929	4960	14426	135000	FALSE	TRUE
930	4800	14426	112000	FALSE	FALSE
931	5125	14426	110000	FALSE	FALSE
932	5376	14426	149000	FALSE	TRUE
933	5088	14426	147800	FALSE	TRUE
934	5400	14426	131500	FALSE	FALSE
935	5625	14426	129900	FALSE	TRUE
936	5200	14426	135000	FALSE	FALSE
937	6000	14426	134000	FALSE	FALSE
938	5040	14426	121500	FALSE	FALSE
939	9230	14426	187000	FALSE	FALSE
940	12000	14426	158000	FALSE	FALSE
941	6762	14426	127900	FALSE	FALSE
942	6500	14426	178000	FALSE	FALSE
943	5120	14426	115000	FALSE	FALSE
944	4800	14426	124000	FALSE	FALSE
945	20400	14426	147500	FALSE	FALSE
946	4800	14426	90000	FALSE	FALSE
947	6000	14426	127900	FALSE	FALSE
948	7200	14426	102500	FALSE	FALSE
949	6160	14426	120000	FALSE	FALSE
950	6240	14426	90000	FALSE	FALSE
951	5400	14426	153000	FALSE	TRUE
952	8280	14426	135000	FALSE	FALSE
953	10512	14426	75000	FALSE	FALSE
954	7420	14426	112000	FALSE	FALSE
955	6765	14426	133900	FALSE	FALSE
956	6780	14426	52500	FALSE	FALSE

957	8004	14426	143000	FALSE	TRUE
958	5150	14426	102000	FALSE	FALSE
959	5000	14426	105900	FALSE	FALSE
960	3690	14426	30200	FALSE	FALSE
961	6300	14426	59900	FALSE	FALSE
962	4601	14426	55200	FALSE	FALSE
963	2660	14426	64900	FALSE	FALSE
964	3775	14426	199000	FALSE	FALSE
965	2400	14426	88000	FALSE	FALSE
966	3502	14426	177000	FALSE	TRUE
967	3630	14426	77500	FALSE	FALSE
968	4920	14426	142000	FALSE	FALSE
969	5040	14426	135000	FALSE	FALSE
970	6579	14426	114900	FALSE	FALSE
971	5192	14426	65000	FALSE	FALSE
972	3600	14426	48600	FALSE	FALSE
973	5160	14426	103000	FALSE	FALSE
974	5031	14426	54000	FALSE	FALSE
975	4800	14426	54500	FALSE	FALSE
976	2400	14426	60000	FALSE	FALSE
977	4200	14426	90000	FALSE	FALSE
978	7800	14426	140000	FALSE	FALSE
979	4170	14426	86858	FALSE	FALSE
980	3150	14426	66800	FALSE	FALSE
981	3844	14426	65900	FALSE	FALSE
982	3500	14426	133000	FALSE	FALSE
983	5969	14426	98000	FALSE	FALSE
984	24527	14426	323800	FALSE	FALSE
985	14948	14426	74000	FALSE	FALSE
986	7200	14426	184000	FALSE	FALSE
987	7500	14426	9800	FALSE	FALSE
988	20296	14426	134700	FALSE	FALSE
989	11250	14426	139900	FALSE	FALSE
990	10000	14426	162000	FALSE	FALSE
991	15862	14426	165900	FALSE	TRUE
992	7500	14426	142500	FALSE	FALSE
993	13500	14426	124000	FALSE	FALSE
994	7200	14426	88500	FALSE	FALSE
995	9028	14426	92000	FALSE	FALSE
996	7200	14426	80000	FALSE	FALSE
997	8580	14426	119000	FALSE	FALSE
998	7440	14426	105000	FALSE	FALSE
999	7200	14426	128000	FALSE	FALSE

1000	5328	14426	194000	FALSE	FALSE
1001	5500	14426	85000	FALSE	FALSE
1002	5160	14426	167000	FALSE	FALSE
1003	10613	14426	153000	FALSE	FALSE
1004	7440	14426	154000	FALSE	FALSE
1005	5600	14426	38000	FALSE	FALSE
1006	7095	14426	155500	FALSE	FALSE
1007	4960	14426	128000	FALSE	FALSE
1008	5652	14426	140000	FALSE	FALSE
1009	6375	14426	200000	FALSE	FALSE
1010	4320	14426	215000	FALSE	FALSE
1011	2880	14426	170000	FALSE	FALSE
1012	4800	14426	183500	FALSE	FALSE
1013	4800	14426	175000	FALSE	FALSE
1014	4662	14426	135000	FALSE	FALSE
1015	6750	14426	144000	FALSE	TRUE
1016	7500	14426	123500	FALSE	FALSE
1017	5400	14426	190000	FALSE	TRUE
1018	5040	14426	142500	FALSE	FALSE
1019	4800	14426	126000	FALSE	FALSE
1020	6440	14426	162800	FALSE	FALSE
1021	5760	14426	80000	FALSE	FALSE
1022	3660	14426	96000	FALSE	FALSE
1023	3706	14426	139500	FALSE	FALSE
1024	4840	14426	84450	FALSE	FALSE
1025	8100	14426	105000	FALSE	FALSE
1026	8400	14426	190000	FALSE	FALSE
1027	3840	14426	162000	FALSE	FALSE
1028	4181	14426	120000	FALSE	FALSE
1029	3600	14426	154000	FALSE	FALSE
1030	3600	14426	120000	FALSE	FALSE
1031	4920	14426	165000	FALSE	FALSE
1032	4960	14426	162000	FALSE	FALSE
1033	8568	14426	154000	FALSE	FALSE
1034	8680	14426	160000	FALSE	FALSE
1035	5500	14426	149500	FALSE	FALSE
1036	6608	14426	139900	FALSE	FALSE
1037	6160	14426	143900	FALSE	FALSE
1038	5280	14426	116000	FALSE	FALSE
1039	4386	14426	134500	FALSE	FALSE
1040	5475	14426	195000	FALSE	FALSE
1041	8505	14426	225000	FALSE	FALSE
1042	4800	14426	136000	FALSE	FALSE

1043	6510	14426	213000	FALSE	FALSE
1044	8253	14426	165500	FALSE	FALSE
1045	7686	14426	153000	FALSE	FALSE
1046	6120	14426	150000	FALSE	FALSE
1047	8160	14426	124000	FALSE	FALSE
1048	8550	14426	168000	FALSE	FALSE
1049	6681	14426	178000	FALSE	TRUE
1050	6700	14426	115000	FALSE	FALSE
1051	9940	14426	139000	FALSE	FALSE
1052	5500	14426	173000	FALSE	FALSE
1053	7280	14426	113500	FALSE	FALSE
1054	7625	14426	160000	FALSE	FALSE
1055	8040	14426	181000	FALSE	TRUE
1056	6095	14426	143000	FALSE	FALSE
1057	5576	14426	82500	FALSE	FALSE
1058	8370	14426	162900	FALSE	FALSE
1059	7560	14426	172000	FALSE	TRUE
1060	7500	14426	82000	FALSE	FALSE
1061	7500	14426	82000	FALSE	FALSE
1062	7140	14426	148000	FALSE	FALSE
1063	7132	14426	118000	FALSE	FALSE
1064	7560	14426	109000	FALSE	FALSE
1065	5980	14426	129200	FALSE	FALSE
1066	6800	14426	146000	FALSE	TRUE
1067	6750	14426	133000	FALSE	FALSE
1068	5000	14426	107000	FALSE	FALSE
1069	6264	14426	118900	FALSE	FALSE
1070	9272	14426	242000	FALSE	TRUE
1071	4200	14426	52000	FALSE	FALSE
1072	3720	14426	72000	FALSE	FALSE
1073	6720	14426	164000	FALSE	FALSE
1074	3750	14426	75000	FALSE	FALSE
1075	3600	14426	133000	FALSE	FALSE
1077	6450	14426	133500	FALSE	FALSE
1078	4838	14426	159000	FALSE	FALSE
1079	8450	14426	135800	FALSE	FALSE
1080	5160	14426	144600	FALSE	FALSE
1081	7290	14426	110000	FALSE	FALSE
1082	4920	14426	125000	FALSE	FALSE
1083	5350	14426	138000	FALSE	TRUE
1084	3658	14426	127000	FALSE	FALSE
1085	5400	14426	130000	FALSE	FALSE
1086	4720	14426	126500	FALSE	FALSE

1087	9750	14426	246400	FALSE	FALSE
1088	10980	14426	209500	FALSE	FALSE
1089	7740	14426	182000	FALSE	FALSE
1090	7480	14426	161000	FALSE	FALSE
1091	7250	14426	167000	FALSE	FALSE
1092	6460	14426	166900	FALSE	FALSE
1093	6000	14426	160500	FALSE	FALSE
1094	9840	14426	142339	FALSE	FALSE
1095	21441	14426	182000	FALSE	TRUE
1096	8520	14426	149900	FALSE	FALSE
1097	7780	14426	75000	FALSE	FALSE
1098	6950	14426	145000	FALSE	FALSE
1099	6837	14426	121000	FALSE	FALSE
1100	5896	14426	135000	FALSE	FALSE
1101	6579	14426	105500	FALSE	FALSE
1102	4860	14426	79000	FALSE	FALSE
1103	16320	14426	149900	FALSE	FALSE
1104	5250	14426	83750	FALSE	FALSE
1105	6534	14426	172500	FALSE	FALSE
1106	5580	14426	145000	FALSE	FALSE
1107	3720	14426	130000	FALSE	FALSE
1108	5310	14426	134000	FALSE	FALSE
1109	4142	14426	168000	FALSE	TRUE
1110	4142	14426	148000	FALSE	FALSE
1111	3450	14426	159000	FALSE	FALSE
1112	1350	14426	75000	FALSE	FALSE
1113	1350	14426	75000	FALSE	FALSE
1114	4180	14426	94000	FALSE	FALSE
1115	4800	14426	165500	FALSE	FALSE
1116	5625	14426	160000	FALSE	FALSE
1117	4880	14426	106500	FALSE	FALSE
1118	4800	14426	140000	FALSE	FALSE
1119	8235	14426	413000	FALSE	TRUE
1120	3120	14426	144000	FALSE	FALSE
1121	3120	14426	147500	FALSE	TRUE
1122	6050	14426	181000	FALSE	FALSE
1123	5016	14426	215000	FALSE	FALSE
1124	6480	14426	226000	FALSE	TRUE
1125	4320	14426	184900	FALSE	FALSE
1126	3690	14426	157000	FALSE	FALSE
1127	7440	14426	159000	FALSE	FALSE
1128	3840	14426	144000	FALSE	FALSE
1129	4880	14426	125000	FALSE	FALSE

1130	5040	14426	152000	FALSE	TRUE
1131	11075	14426	100000	FALSE	FALSE
1132	11160	14426	199700	FALSE	FALSE
1133	3840	14426	110700	FALSE	FALSE
1134	3390	14426	129900	FALSE	FALSE
1135	1750	14426	128000	FALSE	FALSE
1136	6081	14426	146000	FALSE	FALSE
1137	6240	14426	64000	FALSE	FALSE
1138	4800	14426	55000	FALSE	FALSE
1139	3600	14426	5500	FALSE	FALSE
1140	5040	14457	65000	FALSE	FALSE
1141	6150	14457	50000	FALSE	FALSE
1142	4800	14457	90000	FALSE	FALSE
1143	3840	14457	83000	FALSE	FALSE
1144	9000	14457	12000	FALSE	FALSE
1145	3600	14457	84000	FALSE	FALSE
1146	21025	14457	140000	FALSE	FALSE
1147	6120	14457	117900	FALSE	FALSE
1148	5940	14457	30000	FALSE	FALSE
1149	3750	14457	55000	FALSE	FALSE
1150	6750	14457	28900	FALSE	FALSE
1151	8540	14457	73500	FALSE	FALSE
1152	10133	14457	234900	FALSE	FALSE
1153	10921	14457	285000	FALSE	FALSE
1154	10080	14457	100000	FALSE	FALSE
1155	6500	14457	105000	FALSE	FALSE
1156	5040	14457	110000	FALSE	FALSE
1157	10070	14457	104000	FALSE	FALSE
1158	7254	14457	97000	FALSE	FALSE
1159	6600	14457	100000	FALSE	FALSE
1160	7245	14457	90000	FALSE	FALSE
1161	7200	14457	83000	FALSE	FALSE
1162	7245	14457	74000	FALSE	FALSE
1163	7725	14457	90000	FALSE	FALSE
1164	8244	14457	90900	FALSE	FALSE
1165	4880	14457	47000	FALSE	FALSE
1166	7200	14457	520000	TRUE	TRUE
1167	3600	14457	213000	TRUE	TRUE
1168	5520	14457	280000	TRUE	TRUE
1169	4800	14457	338000	TRUE	TRUE
1170	3600	14457	199900	TRUE	TRUE
1171	5310	14457	201000	FALSE	FALSE
1172	5715	14457	174500	FALSE	FALSE

1173	5240	14457	169000	FALSE	FALSE
1174	6960	14457	120700	FALSE	FALSE
1175	4800	14457	136000	FALSE	FALSE
1176	6000	14457	128000	FALSE	FALSE
1177	5400	14457	115000	FALSE	FALSE
1178	5280	14457	123000	FALSE	FALSE
1179	5280	14457	63000	FALSE	FALSE
1180	5400	14457	154000	FALSE	FALSE
1181	6936	14457	115000	FALSE	FALSE
1182	5000	14457	138000	FALSE	FALSE
1183	4800	14457	157500	FALSE	TRUE
1184	5280	14457	145000	FALSE	FALSE
1185	6250	14457	179000	FALSE	TRUE
1186	5000	14457	143000	FALSE	FALSE
1187	4800	14457	130000	FALSE	FALSE
1188	6000	14457	160000	FALSE	TRUE
1189	5000	14457	156000	FALSE	TRUE
1190	4884	14457	141000	FALSE	TRUE
1191	5000	14457	146000	FALSE	TRUE
1192	5684	14457	103000	FALSE	FALSE
1193	4800	14457	118000	FALSE	FALSE
1194	6000	14457	111500	FALSE	FALSE
1195	16468	14457	350000	FALSE	FALSE
1196	7200	14457	165000	FALSE	FALSE
1197	5400	14457	161000	FALSE	TRUE
1198	5720	14457	140000	FALSE	FALSE
1199	4800	14457	159000	FALSE	FALSE
1200	5080	14457	167000	FALSE	FALSE
1201	16956	14457	291000	FALSE	TRUE
1202	4800	14457	92500	FALSE	FALSE
1203	7200	14457	126000	FALSE	FALSE
1204	6000	14457	93000	FALSE	FALSE
1205	9088	14457	170000	FALSE	TRUE
1206	5080	14457	135500	FALSE	FALSE
1207	10781	14457	113000	FALSE	FALSE
1208	5600	14457	89700	FALSE	FALSE
1209	6625	14457	145000	FALSE	TRUE
1210	5000	14457	128900	FALSE	FALSE
1211	6042	14457	112000	FALSE	FALSE
1212	7200	14457	105000	FALSE	FALSE
1213	4800	14457	94888	FALSE	FALSE
1214	6480	14457	115000	FALSE	FALSE
1215	6875	14457	121250	FALSE	FALSE

1216	5000	14457	60000	FALSE	FALSE
1217	4375	14457	60000	FALSE	FALSE
1218	3900	14457	126000	FALSE	FALSE
1219	4800	14457	160000	FALSE	FALSE
1220	5104	14457	109500	FALSE	FALSE
1221	4042	14457	70000	FALSE	FALSE
1222	7314	14457	86200	FALSE	FALSE
1223	6840	14457	110000	FALSE	FALSE
1224	4680	14457	28500	FALSE	FALSE
1225	4800	14457	140000	FALSE	FALSE
1226	4840	14457	160000	FALSE	FALSE
1227	7172	14457	80000	FALSE	FALSE
1228	5160	14457	79000	FALSE	FALSE
1229	5074	14457	66000	FALSE	FALSE
1230	4500	14457	83000	FALSE	FALSE
1231	7620	14457	64000	FALSE	FALSE
1232	4167	14457	123500	FALSE	FALSE
1233	4484	14457	147000	FALSE	TRUE
1234	3600	14457	70000	FALSE	FALSE
1235	4810	14457	65000	FALSE	FALSE
1236	5600	14457	60000	FALSE	FALSE
1237	5555	14457	73000	FALSE	FALSE
1238	11129	14457	17000	FALSE	FALSE
1239	11856	14457	154900	FALSE	FALSE
1240	11687	14457	200000	FALSE	FALSE
1241	20296	14457	159000	FALSE	FALSE
1242	7812	14457	95000	FALSE	FALSE
1243	9440	14457	136500	FALSE	FALSE
1244	11950	14457	129900	FALSE	FALSE
1245	17663	14457	132000	FALSE	FALSE
1246	7200	14457	95000	FALSE	FALSE
1247	7800	14457	84000	FALSE	FALSE
1248	7739	14457	129000	FALSE	FALSE
1249	5400	14457	190000	FALSE	FALSE
1250	5760	14457	175000	FALSE	FALSE
1251	6000	14457	102000	FALSE	FALSE
1252	5400	14457	168000	FALSE	FALSE
1253	5512	14457	195000	FALSE	TRUE
1254	6450	14457	143000	FALSE	FALSE
1255	4320	14457	149900	FALSE	FALSE
1256	6600	14457	144500	FALSE	FALSE
1257	6650	14457	100000	FALSE	FALSE
1258	6992	14457	162500	FALSE	FALSE

1259	3570	14457	165000	FALSE	FALSE
1260	8400	14457	156000	FALSE	FALSE
1261	4800	14457	133500	FALSE	FALSE
1262	4640	14457	145400	FALSE	FALSE
1263	4800	14457	149000	FALSE	FALSE
1264	4800	14457	104900	FALSE	FALSE
1265	5160	14457	150000	FALSE	FALSE
1266	4758	14457	156000	FALSE	FALSE
1267	4800	14457	210000	FALSE	TRUE
1268	5000	14457	173000	FALSE	FALSE
1269	4480	14457	133000	FALSE	FALSE
1270	5625	14457	105000	FALSE	FALSE
1271	5767	14457	106000	FALSE	FALSE
1272	4800	14457	49700	FALSE	FALSE
1273	4284	14457	107000	FALSE	FALSE
1274	4046	14457	75000	FALSE	FALSE
1275	6018	14457	121500	FALSE	FALSE
1276	4560	14457	170000	FALSE	FALSE
1277	5760	14457	75000	FALSE	FALSE
1278	5580	14457	101000	FALSE	FALSE
1279	6650	14457	134000	FALSE	FALSE
1280	4760	14457	89000	FALSE	FALSE
1281	4641	14457	117900	FALSE	FALSE
1282	5200	14457	64500	FALSE	FALSE
1283	3840	14457	89900	FALSE	FALSE
1284	6192	14457	220000	FALSE	FALSE
1285	6192	14457	220000	FALSE	FALSE
1286	5610	14457	120000	FALSE	FALSE
1287	4800	14457	112000	FALSE	FALSE
1288	7750	14457	168000	FALSE	FALSE
1289	6930	14457	220000	FALSE	FALSE
1290	8510	14457	215000	FALSE	FALSE
1291	6500	14457	208000	FALSE	FALSE
1292	4920	14457	169900	FALSE	FALSE
1293	6700	14457	110000	FALSE	FALSE
1294	3240	14457	120000	FALSE	FALSE
1295	5208	14457	158000	FALSE	TRUE
1296	7100	14457	122500	FALSE	FALSE
1297	7200	14457	152000	FALSE	FALSE
1298	4797	14457	230000	FALSE	FALSE
1299	6500	14457	166000	FALSE	FALSE
1300	4040	14457	143000	FALSE	FALSE
1301	7980	14457	190000	FALSE	TRUE

1302	6600	14457	195000	FALSE	FALSE
1303	7500	14457	173000	FALSE	FALSE
1304	5750	14457	143000	FALSE	FALSE
1305	7440	14457	154900	FALSE	FALSE
1306	7200	14457	180000	FALSE	FALSE
1307	7560	14457	195000	FALSE	TRUE
1308	8850	14457	150000	FALSE	FALSE
1309	7000	14457	170400	FALSE	FALSE
1310	8400	14457	144000	FALSE	FALSE
1311	9660	14457	149000	FALSE	FALSE
1312	7560	14457	180000	FALSE	FALSE
1313	8040	14457	164000	FALSE	FALSE
1314	6600	14457	193000	FALSE	TRUE
1315	7280	14457	157000	FALSE	FALSE
1316	6920	14457	146500	FALSE	FALSE
1317	7620	14457	138900	FALSE	FALSE
1318	7800	14457	142000	FALSE	FALSE
1319	7590	14457	161000	FALSE	FALSE
1320	12580	14457	138500	FALSE	FALSE
1321	6360	14457	117000	FALSE	FALSE
1322	9600	14457	140000	FALSE	FALSE
1323	9425	14457	130000	FALSE	FALSE
1324	11200	14457	145000	FALSE	TRUE
1325	7280	14457	126500	FALSE	FALSE
1326	6360	14457	125000	FALSE	FALSE
1327	9540	14457	180500	FALSE	TRUE
1328	4800	14457	117000	FALSE	FALSE
1329	8245	14457	82000	FALSE	FALSE
1330	6000	14457	99000	FALSE	FALSE
1331	5000	14457	129900	FALSE	FALSE
1332	5000	14457	151000	FALSE	TRUE
1333	4800	14457	83000	FALSE	FALSE
1334	5360	14457	134500	FALSE	FALSE
1335	4446	14457	134900	FALSE	FALSE
1336	3000	14457	69900	FALSE	FALSE
1337	2880	14457	25000	FALSE	FALSE
1338	3600	14457	70000	FALSE	FALSE
1339	2850	14457	91000	FALSE	FALSE
1340	4200	14457	74900	FALSE	FALSE
1341	3600	14457	40000	FALSE	FALSE
1343	2800	14457	74000	FALSE	FALSE
1344	20000	14457	185900	FALSE	FALSE
1345	3600	14457	114000	FALSE	FALSE

1346	10560	14457	196000	FALSE	FALSE
1347	4960	14457	149800	FALSE	FALSE
1348	4800	14457	167400	FALSE	TRUE
1349	6576	14457	153000	FALSE	FALSE
1350	5400	14457	124700	FALSE	FALSE
1351	4375	14457	147000	FALSE	FALSE
1352	4880	14457	127500	FALSE	FALSE
1353	4800	14457	150000	FALSE	FALSE
1354	4800	14457	95000	FALSE	FALSE
1355	3600	14457	118000	FALSE	FALSE
1356	9200	14457	98000	FALSE	FALSE
1357	4320	14457	144000	FALSE	FALSE
1358	5040	14457	186000	FALSE	TRUE
1359	5900	14457	130000	FALSE	FALSE
1360	19932	14457	198000	FALSE	TRUE
1361	13404	14457	251000	FALSE	TRUE
1362	9000	14457	211850	FALSE	TRUE
1363	8833	14457	165000	FALSE	FALSE
1364	5160	14457	135000	FALSE	FALSE
1365	9600	14457	145000	FALSE	FALSE
1366	7480	14457	165000	FALSE	FALSE
1367	6812	14457	164800	FALSE	FALSE
1368	7820	14457	124000	FALSE	FALSE
1369	5160	14457	112400	FALSE	FALSE
1370	6345	14457	117000	FALSE	FALSE
1371	5280	14457	123500	FALSE	FALSE
1372	5080	14457	148000	FALSE	FALSE
1373	4845	14457	130000	FALSE	FALSE
1374	6150	14457	249500	FALSE	FALSE
1375	7488	14457	232000	FALSE	FALSE
1376	5828	14457	154900	FALSE	FALSE
1377	5200	14457	142000	FALSE	FALSE
1378	3600	14457	132000	FALSE	FALSE
1379	7500	14457	165000	FALSE	TRUE
1380	4636	14457	147000	FALSE	FALSE
1381	5400	14457	215000	FALSE	TRUE
1382	4520	14457	180000	FALSE	TRUE
1383	4800	14457	116500	FALSE	FALSE
1384	5400	14457	215500	FALSE	TRUE
1385	3660	14457	40000	FALSE	FALSE
1386	5700	14457	261700	FALSE	FALSE
1387	4800	14457	182000	FALSE	FALSE
1388	4200	14457	159000	FALSE	FALSE

1389	2850	14457	165000	FALSE	FALSE
1390	4636	14457	131500	FALSE	FALSE
1391	12220	14457	510000	FALSE	TRUE
1392	7100	14457	134000	FALSE	FALSE
1393	3090	14457	129503	FALSE	FALSE
1394	5560	14457	126000	FALSE	FALSE
1395	6696	14457	148900	FALSE	TRUE
1396	4720	14457	134000	FALSE	FALSE
1397	5640	14457	115600	FALSE	FALSE
1398	4800	14457	99900	FALSE	FALSE
1399	6000	14457	303000	FALSE	TRUE
1400	6000	14457	250000	FALSE	TRUE
1401	4800	14457	220000	FALSE	FALSE
1402	6000	14457	220000	FALSE	TRUE
1403	4050	14457	126000	FALSE	FALSE
1404	3600	14457	74900	FALSE	FALSE
1405	3150	14457	126000	FALSE	FALSE
1406	4200	14457	85000	FALSE	FALSE
1407	3120	14457	114300	FALSE	FALSE
1408	5160	14457	152000	FALSE	FALSE
1409	3600	14457	26500	FALSE	FALSE
1410	3780	14457	35500	FALSE	FALSE
1411	7500	14457	72300	FALSE	FALSE
1412	8415	14457	190000	FALSE	FALSE
1413	6480	14488	165000	FALSE	FALSE
1414	6528	14488	120000	FALSE	FALSE
1415	4920	14488	79000	FALSE	FALSE
1416	8650	14488	125000	FALSE	FALSE
1417	3960	14488	29000	FALSE	FALSE
1418	5460	14488	94000	FALSE	FALSE
1419	10935	14488	108000	FALSE	FALSE
1420	6600	14488	40000	FALSE	FALSE
1421	5460	14488	130000	FALSE	FALSE
1422	5330	14488	108000	FALSE	FALSE
1423	5460	14488	38000	FALSE	FALSE
1424	5400	14488	41900	FALSE	FALSE
1425	9972	14488	82000	FALSE	FALSE
1426	4800	14488	98000	FALSE	FALSE
1427	4800	14488	65000	FALSE	FALSE
1428	8715	14488	103000	FALSE	FALSE
1429	6592	14488	97000	FALSE	FALSE
1430	6760	14488	64000	FALSE	FALSE
1431	3465	14488	235000	TRUE	FALSE

1432	3600	14488	109000	TRUE	FALSE
1433	3600	14488	94000	TRUE	FALSE
1434	3600	14488	201000	TRUE	TRUE
1435	3600	14488	60000	TRUE	FALSE
1436	4560	14488	184000	TRUE	TRUE
1437	4800	14488	372000	TRUE	FALSE
1438	7200	14488	512400	TRUE	TRUE
1439	7320	14488	480000	TRUE	TRUE
1440	3000	14488	243683	TRUE	TRUE
1441	3600	14488	144000	TRUE	FALSE
1442	8700	14488	399500	TRUE	FALSE
1443	9000	14488	180000	FALSE	FALSE
1444	4756	14488	190000	FALSE	FALSE
1445	5040	14488	177000	FALSE	TRUE
1446	4960	14488	110200	FALSE	FALSE
1447	4800	14488	145000	FALSE	FALSE
1448	5000	14488	144000	FALSE	FALSE
1449	21780	14488	163000	FALSE	TRUE
1450	5000	14488	157000	FALSE	TRUE
1451	7000	14488	145000	FALSE	FALSE
1452	4520	14488	100000	FALSE	FALSE
1453	5000	14488	146000	FALSE	TRUE
1454	5376	14488	140000	FALSE	TRUE
1455	4988	14488	156000	FALSE	FALSE
1456	5280	14488	165000	FALSE	FALSE
1457	10800	14488	190000	FALSE	TRUE
1458	10848	14488	145000	FALSE	FALSE
1459	6250	14488	205000	FALSE	TRUE
1460	21780	14488	166500	FALSE	FALSE
1461	21780	14488	195000	FALSE	FALSE
1462	30122	14488	187000	FALSE	FALSE
1463	6000	14488	151000	FALSE	FALSE
1464	9375	14488	90000	FALSE	FALSE
1465	7980	14488	134001	FALSE	FALSE
1466	7224	14488	127000	FALSE	FALSE
1467	10270	14488	142000	FALSE	FALSE
1468	5760	14488	132000	FALSE	FALSE
1469	6120	14488	149900	FALSE	TRUE
1470	5106	14488	144000	FALSE	TRUE
1471	5000	14488	145000	FALSE	FALSE
1472	4800	14488	158000	FALSE	TRUE
1473	6750	14488	86500	FALSE	FALSE
1474	4960	14488	145000	FALSE	FALSE

1475	5000	14488	108000	FALSE	FALSE
1476	5080	14488	147500	FALSE	TRUE
1477	5535	14488	80000	FALSE	FALSE
1478	4375	14488	13000	FALSE	FALSE
1479	4440	14488	12000	FALSE	FALSE
1480	2720	14488	20000	FALSE	FALSE
1481	4640	14488	82500	FALSE	FALSE
1482	3800	14488	25000	FALSE	FALSE
1483	5440	14488	78000	FALSE	FALSE
1484	4920	14488	138900	FALSE	FALSE
1485	4445	14488	79000	FALSE	FALSE
1486	3600	14488	10000	FALSE	FALSE
1487	3564	14488	25000	FALSE	FALSE
1488	4760	14488	107500	FALSE	FALSE
1489	5250	14488	154500	FALSE	FALSE
1490	3780	14488	61500	FALSE	FALSE
1491	4765	14488	105000	FALSE	FALSE
1492	3960	14488	119000	FALSE	FALSE
1493	3795	14488	115000	FALSE	FALSE
1494	4200	14488	74900	FALSE	FALSE
1495	3540	14488	55000	FALSE	FALSE
1496	3133	14488	78000	FALSE	FALSE
1497	3750	14488	85000	FALSE	FALSE
1498	3750	14488	79000	FALSE	FALSE
1499	3600	14488	115000	FALSE	FALSE
1500	10004	14488	149000	FALSE	FALSE
1501	6600	14488	109900	FALSE	FALSE
1502	10425	14488	131000	FALSE	FALSE
1503	8637	14488	140000	FALSE	FALSE
1504	8750	14488	127000	FALSE	FALSE
1505	8640	14488	36000	FALSE	FALSE
1506	6120	14488	127000	FALSE	FALSE
1507	5160	14488	152900	FALSE	FALSE
1508	4800	14488	75000	FALSE	FALSE
1509	7200	14488	185000	FALSE	TRUE
1510	5200	14488	105000	FALSE	FALSE
1511	6500	14488	209000	FALSE	FALSE
1512	5550	14488	224000	FALSE	FALSE
1513	4800	14488	213000	FALSE	FALSE
1514	5336	14488	69000	FALSE	FALSE
1515	3400	14488	84900	FALSE	FALSE
1516	4800	14488	195000	FALSE	TRUE
1517	5625	14488	140000	FALSE	FALSE

1518	4200	14488	145000	FALSE	FALSE
1519	4473	14488	112000	FALSE	FALSE
1520	5200	14488	180000	FALSE	FALSE
1521	5200	14488	120000	FALSE	FALSE
1522	5200	14488	120000	FALSE	FALSE
1523	7200	14488	156500	FALSE	FALSE
1524	4800	14488	93000	FALSE	FALSE
1525	3955	14488	132000	FALSE	FALSE
1526	6148	14488	181500	FALSE	FALSE
1527	5289	14488	161800	FALSE	TRUE
1528	4920	14488	105000	FALSE	FALSE
1529	5160	14488	85500	FALSE	FALSE
1530	4760	14488	49900	FALSE	FALSE
1531	3680	14488	81496	FALSE	FALSE
1532	4800	14488	148000	FALSE	FALSE
1533	6076	14488	419000	FALSE	FALSE
1534	8700	14488	208000	FALSE	TRUE
1535	5625	14488	181000	FALSE	FALSE
1536	7200	14488	167000	FALSE	FALSE
1537	6255	14488	101000	FALSE	FALSE
1538	4800	14488	117000	FALSE	FALSE
1539	5400	14488	145000	FALSE	FALSE
1540	5544	14488	138000	FALSE	FALSE
1541	5922	14488	126300	FALSE	FALSE
1542	5000	14488	167500	FALSE	FALSE
1543	5400	14488	143500	FALSE	FALSE
1544	4920	14488	153000	FALSE	FALSE
1545	4998	14488	140000	FALSE	FALSE
1546	6345	14488	109900	FALSE	FALSE
1547	4920	14488	132000	FALSE	FALSE
1548	4920	14488	125700	FALSE	FALSE
1549	7936	14488	225000	FALSE	FALSE
1550	7700	14488	267000	FALSE	FALSE
1551	6250	14488	168000	FALSE	FALSE
1552	10107	14488	220500	FALSE	FALSE
1553	4875	14488	163000	FALSE	FALSE
1554	5125	14488	180000	FALSE	TRUE
1555	6000	14488	172000	FALSE	FALSE
1556	7320	14488	190000	FALSE	TRUE
1557	7896	14488	167000	FALSE	FALSE
1558	5400	14488	130000	FALSE	FALSE
1559	4960	14488	162000	FALSE	FALSE
1560	4800	14488	145000	FALSE	FALSE

1561	8540	14488	218000	FALSE	FALSE
1562	7896	14488	187000	FALSE	FALSE
1563	8211	14488	160000	FALSE	FALSE
1564	6240	14488	150000	FALSE	FALSE
1565	6800	14488	155000	FALSE	FALSE
1566	7480	14488	160000	FALSE	FALSE
1567	12870	14488	175000	FALSE	FALSE
1568	8400	14488	159900	FALSE	FALSE
1569	7105	14488	150000	FALSE	FALSE
1570	7535	14488	170000	FALSE	FALSE
1571	5000	14488	115000	FALSE	FALSE
1572	6188	14488	128500	FALSE	FALSE
1573	7480	14488	103500	FALSE	FALSE
1574	6035	14488	132000	FALSE	FALSE
1575	4800	14488	137500	FALSE	FALSE
1576	10020	14488	162000	FALSE	FALSE
1577	4950	14488	101000	FALSE	FALSE
1578	7500	14488	146100	FALSE	FALSE
1579	6960	14488	116750	FALSE	FALSE
1580	7644	14488	152000	FALSE	TRUE
1581	6750	14488	124000	FALSE	FALSE
1582	4800	14488	110000	FALSE	FALSE
1583	5600	14488	68500	FALSE	FALSE
1584	4879	14488	205900	FALSE	FALSE
1585	7200	14488	150000	FALSE	FALSE
1586	3600	14488	48000	FALSE	FALSE
1587	9000	14488	179900	FALSE	FALSE
1588	5500	14488	160000	FALSE	FALSE
1589	7320	14488	189000	FALSE	FALSE
1590	5670	14488	121500	FALSE	FALSE
1591	5460	14488	150000	FALSE	FALSE
1592	4840	14488	133000	FALSE	FALSE
1593	6450	14488	152500	FALSE	FALSE
1594	4920	14488	152000	FALSE	FALSE
1595	4800	14488	130000	FALSE	FALSE
1596	5760	14488	153300	FALSE	FALSE
1597	7200	14488	111000	FALSE	FALSE
1598	5280	14488	159000	FALSE	FALSE
1599	3520	14488	110000	FALSE	FALSE
1600	3600	14488	119000	FALSE	FALSE
1601	9460	14488	190000	FALSE	FALSE
1602	8040	14488	184000	FALSE	FALSE
1603	6000	14488	165000	FALSE	FALSE

1604	7228	14488	132200	FALSE	FALSE
1605	7125	14488	154900	FALSE	FALSE
1606	7930	14488	128800	FALSE	FALSE
1607	6400	14488	127000	FALSE	FALSE
1608	4840	14488	104900	FALSE	FALSE
1609	9840	14488	113000	FALSE	FALSE
1610	5400	14488	105000	FALSE	FALSE
1611	6600	14488	145000	FALSE	TRUE
1612	4800	14488	100000	FALSE	FALSE
1613	9460	14488	89900	FALSE	FALSE
1614	4200	14488	171900	FALSE	FALSE
1615	5904	14488	138000	FALSE	FALSE
1616	5040	14488	152500	FALSE	FALSE
1617	5400	14488	128000	FALSE	FALSE
1618	5940	14488	153200	FALSE	FALSE
1619	6480	14488	145000	FALSE	FALSE
1620	3986	14488	245000	FALSE	FALSE
1621	3840	14488	154900	FALSE	FALSE
1622	3750	14488	140000	FALSE	FALSE
1623	5200	14488	206600	FALSE	FALSE
1624	3660	14488	158000	FALSE	FALSE
1625	3600	14488	150000	FALSE	FALSE
1626	3922	14488	178000	FALSE	FALSE
1627	5040	14488	238000	FALSE	TRUE
1628	5280	14488	147000	FALSE	FALSE
1629	3600	14488	136000	FALSE	FALSE
1630	3660	14488	95000	FALSE	FALSE
1631	3200	14488	198000	FALSE	TRUE
1632	7260	14488	369000	FALSE	TRUE
1633	4880	14488	150000	FALSE	FALSE
1634	3600	14488	240000	FALSE	TRUE
1635	5538	14488	145000	FALSE	FALSE
1636	3540	14488	115000	FALSE	FALSE
1637	4998	14488	171000	FALSE	FALSE
1638	5880	14488	155000	FALSE	FALSE
1639	6950	14488	130000	FALSE	FALSE
1640	4800	14488	137000	FALSE	TRUE
1641	3750	14488	250000	FALSE	TRUE
1642	4020	14488	190000	FALSE	FALSE
1643	3870	14488	150000	FALSE	FALSE
1644	3000	14488	105900	FALSE	FALSE
1645	4880	14488	85000	FALSE	FALSE
1646	3810	14488	129000	FALSE	FALSE

1647	4800	14488	100000	FALSE	FALSE
1648	3720	14488	20000	FALSE	FALSE
1649	7320	14518	65000	FALSE	FALSE
1650	8601	14518	109000	FALSE	FALSE
1651	4800	14518	71500	FALSE	FALSE
1652	6075	14518	88900	FALSE	FALSE
1653	3600	14518	51000	FALSE	FALSE
1654	7100	14518	65000	FALSE	FALSE
1655	4800	14518	18000	FALSE	FALSE
1656	7636	14518	65000	FALSE	FALSE
1657	5400	14518	52000	FALSE	FALSE
1658	6028	14518	66000	FALSE	FALSE
1659	7905	14518	102500	FALSE	FALSE
1660	5500	14518	82000	FALSE	FALSE
1661	5124	14518	30000	FALSE	FALSE
1662	6550	14518	66000	FALSE	FALSE
1663	4800	14518	89600	FALSE	FALSE
1664	8494	14518	39000	FALSE	FALSE
1665	10230	14518	25000	FALSE	FALSE
1666	4400	14518	84000	FALSE	FALSE
1667	13028	14518	88000	FALSE	FALSE
1668	5320	14518	92900	FALSE	FALSE
1669	5080	14518	125000	FALSE	FALSE
1670	5400	14518	84900	FALSE	FALSE
1671	5256	14518	105300	FALSE	FALSE
1672	6000	14518	114000	FALSE	FALSE
1673	4920	14518	104500	FALSE	FALSE
1674	4800	14518	58000	FALSE	FALSE
1675	6480	14518	107000	FALSE	FALSE
1676	5000	14518	78000	FALSE	FALSE
1677	7272	14518	68000	FALSE	FALSE
1678	6500	14518	78000	FALSE	FALSE
1679	6000	14518	85000	FALSE	FALSE
1680	4800	14518	105000	TRUE	FALSE
1681	3600	14518	117000	TRUE	TRUE
1682	4400	14518	167900	TRUE	TRUE
1683	3600	14518	94000	TRUE	FALSE
1684	4305	14518	159000	TRUE	TRUE
1685	5640	14518	179000	TRUE	TRUE
1686	6000	14518	380000	TRUE	TRUE
1687	4800	14518	267500	TRUE	TRUE
1688	3720	14518	260000	TRUE	TRUE
1689	3720	14518	258800	TRUE	TRUE

1690	4120	14518	210000	TRUE	TRUE
1691	3240	14518	133600	TRUE	TRUE
1692	5250	14518	125000	FALSE	FALSE
1693	6710	14518	183000	FALSE	TRUE
1694	5120	14518	145000	FALSE	FALSE
1695	5376	14518	153000	FALSE	FALSE
1696	4520	14518	143000	FALSE	FALSE
1697	6490	14518	90000	FALSE	FALSE
1698	4687	14518	188000	FALSE	TRUE
1699	4800	14518	131500	FALSE	FALSE
1700	8680	14518	80000	FALSE	FALSE
1701	8680	14518	127000	FALSE	FALSE
1702	5120	14518	96000	FALSE	FALSE
1703	7080	14518	211900	FALSE	FALSE
1704	8100	14518	121000	FALSE	FALSE
1705	9822	14518	160000	FALSE	FALSE
1706	12000	14518	146500	FALSE	FALSE
1707	6000	14518	168500	FALSE	FALSE
1708	6480	14518	165000	FALSE	FALSE
1709	9535	14518	182000	FALSE	FALSE
1710	5520	14518	136500	FALSE	FALSE
1711	5160	14518	137000	FALSE	FALSE
1712	4800	14518	88000	FALSE	FALSE
1713	5000	14518	80000	FALSE	FALSE
1714	7800	14518	153900	FALSE	FALSE
1715	5678	14518	110800	FALSE	FALSE
1716	6630	14518	114000	FALSE	FALSE
1717	4920	14518	89000	FALSE	FALSE
1718	5300	14518	107500	FALSE	FALSE
1719	5600	14518	86990	FALSE	FALSE
1720	5400	14518	93000	FALSE	FALSE
1721	6240	14518	112000	FALSE	FALSE
1722	8555	14518	122000	FALSE	FALSE
1723	6000	14518	100500	FALSE	FALSE
1724	5400	14518	148000	FALSE	TRUE
1725	5280	14518	81300	FALSE	FALSE
1726	5000	14518	129000	FALSE	FALSE
1727	6325	14518	129000	FALSE	FALSE
1728	5760	14518	135000	FALSE	TRUE
1729	9491	14518	122000	FALSE	FALSE
1730	7245	14518	170000	FALSE	FALSE
1731	3026	14518	62000	FALSE	FALSE
1732	5760	14518	48000	FALSE	FALSE

1733	4059	14518	28500	FALSE	FALSE
1734	4560	14518	79900	FALSE	FALSE
1735	7020	14518	74000	FALSE	FALSE
1736	3600	14518	63500	FALSE	FALSE
1737	10070	14518	89400	FALSE	FALSE
1738	3690	14518	15000	FALSE	FALSE
1739	4800	14518	116700	FALSE	FALSE
1740	4800	14518	106500	FALSE	FALSE
1741	6160	14518	79900	FALSE	FALSE
1742	4800	14518	88000	FALSE	FALSE
1743	4920	14518	142500	FALSE	FALSE
1744	6020	14518	25000	FALSE	FALSE
1745	5280	14518	5000	FALSE	FALSE
1746	4920	14518	86000	FALSE	FALSE
1747	3750	14518	67600	FALSE	FALSE
1748	11745	14518	108000	FALSE	FALSE
1749	7280	14518	64000	FALSE	FALSE
1750	3720	14518	67500	FALSE	FALSE
1751	4800	14518	135000	FALSE	FALSE
1752	5080	14518	107000	FALSE	FALSE
1753	5152	14518	65000	FALSE	FALSE
1754	6000	14518	111880	FALSE	FALSE
1755	5080	14518	78000	FALSE	FALSE
1756	3582	14518	62000	FALSE	FALSE
1757	3780	14518	56000	FALSE	FALSE
1758	5360	14518	95000	FALSE	FALSE
1759	4360	14518	149000	FALSE	FALSE
1760	3780	14518	90000	FALSE	FALSE
1761	3570	14518	112000	FALSE	FALSE
1762	4625	14518	115000	FALSE	FALSE
1763	2556	14518	90000	FALSE	FALSE
1764	5400	14518	89900	FALSE	FALSE
1765	3998	14518	90000	FALSE	FALSE
1767	21780	14518	127000	FALSE	FALSE
1768	7200	14518	110000	FALSE	FALSE
1769	5500	14518	56400	FALSE	FALSE
1770	9255	14518	251200	FALSE	FALSE
1771	8192	14518	163300	FALSE	FALSE
1772	17379	14518	145000	FALSE	FALSE
1773	14658	14518	168000	FALSE	FALSE
1774	8479	14518	112000	FALSE	FALSE
1775	7320	14518	90000	FALSE	FALSE
1776	10087	14518	121000	FALSE	FALSE

1777	7950	14518	95900	FALSE	FALSE
1778	8298	14518	114500	FALSE	FALSE
1779	10800	14518	115000	FALSE	FALSE
1780	7200	14518	172000	FALSE	FALSE
1781	8100	14518	154000	FALSE	FALSE
1782	7200	14518	87000	FALSE	FALSE
1783	6375	14518	35000	FALSE	FALSE
1784	6375	14518	27500	FALSE	FALSE
1785	7200	14518	97000	FALSE	FALSE
1786	7200	14518	119900	FALSE	FALSE
1787	9000	14518	114200	FALSE	FALSE
1788	5400	14518	169000	FALSE	FALSE
1789	5400	14518	186000	FALSE	FALSE
1790	9000	14518	182000	FALSE	FALSE
1791	4920	14518	75500	FALSE	FALSE
1792	7440	14518	126000	FALSE	FALSE
1793	7248	14518	252500	FALSE	FALSE
1794	4800	14518	219000	FALSE	FALSE
1795	4800	14518	46000	FALSE	FALSE
1796	4800	14518	145000	FALSE	FALSE
1797	4560	14518	178500	FALSE	TRUE
1798	6400	14518	185000	FALSE	TRUE
1799	4800	14518	208000	FALSE	TRUE
1800	4800	14518	58500	FALSE	FALSE
1801	5900	14518	164300	FALSE	FALSE
1802	6000	14518	146000	FALSE	FALSE
1803	4080	14518	122400	FALSE	FALSE
1804	3660	14518	151000	FALSE	FALSE
1805	6680	14518	139000	FALSE	FALSE
1806	4280	14518	64000	FALSE	FALSE
1807	3810	14518	146000	FALSE	TRUE
1808	13440	14518	139000	FALSE	FALSE
1809	5900	14518	120000	FALSE	FALSE
1810	7280	14518	60000	FALSE	FALSE
1811	5000	14518	83000	FALSE	FALSE
1812	3720	14518	124900	FALSE	FALSE
1813	4524	14518	39600	FALSE	FALSE
1814	5670	14518	115000	FALSE	FALSE
1815	3990	14518	67000	FALSE	FALSE
1816	4200	14518	115000	FALSE	FALSE
1817	4880	14518	90000	FALSE	FALSE
1818	8640	14518	389500	FALSE	TRUE
1819	6750	14518	155000	FALSE	FALSE

1820	8308	14518	135000	FALSE	FALSE
1821	8160	14518	168500	FALSE	FALSE
1822	6000	14518	124900	FALSE	FALSE
1823	4920	14518	155000	FALSE	FALSE
1824	8878	14518	142000	FALSE	FALSE
1825	7860	14518	115000	FALSE	FALSE
1826	7938	14518	124000	FALSE	FALSE
1827	7068	14518	105000	FALSE	FALSE
1828	8040	14518	164900	FALSE	FALSE
1829	7740	14518	179000	FALSE	FALSE
1830	5125	14518	151000	FALSE	FALSE
1831	7980	14518	172700	FALSE	TRUE
1832	5500	14518	171500	FALSE	TRUE
1833	8200	14518	125000	FALSE	FALSE
1834	9106	14518	148000	FALSE	TRUE
1835	8040	14518	175000	FALSE	FALSE
1836	10541	14518	201500	FALSE	FALSE
1837	6550	14518	166500	FALSE	FALSE
1838	8250	14518	153500	FALSE	FALSE
1839	8100	14518	149900	FALSE	FALSE
1840	8100	14518	90000	FALSE	FALSE
1841	7245	14518	161000	FALSE	FALSE
1842	5500	14518	115300	FALSE	FALSE
1843	7100	14518	154900	FALSE	FALSE
1844	7315	14518	129000	FALSE	FALSE
1845	8370	14518	147000	FALSE	FALSE
1846	8308	14518	145500	FALSE	FALSE
1847	7560	14518	166000	FALSE	FALSE
1848	5000	14518	143000	FALSE	FALSE
1849	6960	14518	157500	FALSE	FALSE
1850	7980	14518	120000	FALSE	FALSE
1851	5500	14518	132900	FALSE	FALSE
1852	4800	14518	125000	FALSE	FALSE
1853	5712	14518	112500	FALSE	FALSE
1854	5818	14518	128000	FALSE	FALSE
1855	8760	14518	129000	FALSE	FALSE
1856	9144	14518	104000	FALSE	FALSE
1857	6750	14518	130000	FALSE	FALSE
1858	7440	14518	84000	FALSE	FALSE
1859	6380	14518	99900	FALSE	FALSE
1860	4920	14518	122500	FALSE	FALSE
1861	5760	14518	122000	FALSE	FALSE
1862	4800	14518	90000	FALSE	FALSE

1863	6700	14518	55000	FALSE	FALSE
1864	6765	14518	140000	FALSE	TRUE
1865	8496	14518	156000	FALSE	FALSE
1866	7011	14518	230000	FALSE	FALSE
1867	2820	14518	98000	FALSE	FALSE
1868	5124	14518	130000	FALSE	FALSE
1869	4800	14518	162000	FALSE	FALSE
1870	6768	14518	137500	FALSE	FALSE
1871	6075	14518	82000	FALSE	FALSE
1872	4080	14518	122500	FALSE	FALSE
1873	5336	14518	135000	FALSE	FALSE
1874	8060	14518	254900	FALSE	FALSE
1875	9125	14518	198000	FALSE	TRUE
1876	7500	14518	176000	FALSE	TRUE
1877	6930	14518	165000	FALSE	TRUE
1878	3600	14518	122000	FALSE	FALSE
1879	7300	14518	121500	FALSE	FALSE
1880	8640	14518	158000	FALSE	FALSE
1881	7688	14518	199900	FALSE	TRUE
1882	9600	14518	150000	FALSE	FALSE
1883	7112	14518	149000	FALSE	FALSE
1884	8220	14518	157000	FALSE	FALSE
1885	6450	14518	140000	FALSE	FALSE
1886	7350	14518	150000	FALSE	FALSE
1887	7560	14518	148000	FALSE	FALSE
1888	8030	14518	146600	FALSE	FALSE
1889	6550	14518	161000	FALSE	FALSE
1890	16021	14518	130000	FALSE	FALSE
1891	4800	14518	122000	FALSE	FALSE
1892	6840	14518	155000	FALSE	FALSE
1893	5500	14518	112000	FALSE	FALSE
1894	5840	14518	154900	FALSE	FALSE
1895	7140	14518	104100	FALSE	FALSE
1896	5625	14518	160000	FALSE	FALSE
1897	5400	14518	156000	FALSE	FALSE
1898	7228	14518	163600	FALSE	FALSE
1899	6960	14518	154000	FALSE	FALSE
1900	6075	14518	154000	FALSE	FALSE
1901	4142	14518	145500	FALSE	FALSE
1902	5002	14518	114000	FALSE	FALSE
1903	5080	14518	131820	FALSE	FALSE
1904	6250	14518	255000	FALSE	FALSE
1905	5580	14518	215000	FALSE	FALSE

1906	3540	14518	247000	FALSE	FALSE
1907	5400	14518	202000	FALSE	FALSE
1908	4200	14518	127000	FALSE	FALSE
1909	4800	14518	145000	FALSE	FALSE
1910	4062	14518	254000	FALSE	TRUE
1911	6240	14518	129900	FALSE	FALSE
1912	4880	14518	170000	FALSE	FALSE
1913	4880	14518	150000	FALSE	FALSE
1914	4200	14518	119900	FALSE	FALSE
1915	3600	14518	170100	FALSE	TRUE
1916	4500	14518	153000	FALSE	FALSE
1917	4800	14518	117000	FALSE	FALSE
1918	4000	14518	105000	FALSE	FALSE
1919	4920	14518	91000	FALSE	FALSE
1920	4500	14518	236000	FALSE	TRUE
1921	8160	14518	126300	FALSE	FALSE
1922	3600	14518	109000	FALSE	FALSE
1923	3000	14518	149800	FALSE	FALSE
1924	3600	14518	145500	FALSE	FALSE
1925	3600	14518	69900	FALSE	FALSE
1926	4200	14518	205000	FALSE	TRUE
1927	3960	14518	142700	FALSE	FALSE
1928	4200	14518	230000	FALSE	TRUE
1929	3750	14518	75000	FALSE	FALSE
1930	6400	14518	160000	FALSE	FALSE
1931	6400	14549	77500	FALSE	FALSE
1932	9279	14549	57000	FALSE	FALSE
1933	12542	14549	106500	FALSE	FALSE
1934	5125	14549	67000	FALSE	FALSE
1935	3240	14549	25000	FALSE	FALSE
1936	6125	14549	24500	FALSE	FALSE
1937	7236	14549	80000	FALSE	FALSE
1938	6000	14549	72500	FALSE	FALSE
1939	5720	14549	19000	FALSE	FALSE
1940	5720	14549	10000	FALSE	FALSE
1941	5084	14549	6500	FALSE	FALSE
1942	7015	14549	90000	FALSE	FALSE
1943	5460	14549	116000	FALSE	FALSE
1944	6000	14549	128500	FALSE	FALSE
1945	4920	14549	82000	FALSE	FALSE
1946	5400	14549	82500	FALSE	FALSE
1947	5761	14549	59500	FALSE	FALSE
1948	4800	14549	115000	FALSE	FALSE

1949	7644	14549	87000	FALSE	FALSE
1950	4800	14549	107000	FALSE	FALSE
1951	7245	14549	86900	FALSE	FALSE
1952	9920	14549	39000	FALSE	FALSE
1953	4800	14549	71000	FALSE	FALSE
1954	4800	14549	84000	FALSE	FALSE
1955	4920	14549	70000	FALSE	FALSE
1956	7040	14549	82500	FALSE	FALSE
1957	5940	14549	87000	FALSE	FALSE
1958	9504	14549	32750	FALSE	FALSE
1959	8280	14549	50400	FALSE	FALSE
1960	5040	14549	89900	FALSE	FALSE
1961	5400	14549	39000	FALSE	FALSE
1962	5400	14549	24000	FALSE	FALSE
1963	8128	14549	95000	FALSE	FALSE
1964	7100	14549	99900	FALSE	FALSE
1965	6060	14549	115000	FALSE	FALSE
1966	7200	14549	85000	FALSE	FALSE
1967	6600	14549	65000	FALSE	FALSE
1968	5940	14549	19000	FALSE	FALSE
1969	5940	14549	37500	FALSE	FALSE
1970	6000	14549	68500	FALSE	FALSE
1971	6000	14549	66000	FALSE	FALSE
1972	7700	14549	84000	FALSE	FALSE
1973	5500	14549	114900	FALSE	FALSE
1974	6240	14549	151000	FALSE	FALSE
1975	8100	14549	504000	TRUE	TRUE
1976	5904	14549	215000	TRUE	TRUE
1977	4800	14549	150000	TRUE	TRUE
1978	4800	14549	320000	TRUE	FALSE
1979	3600	14549	115000	TRUE	TRUE
1980	13080	14549	1350000	TRUE	TRUE
1981	2700	14549	229000	TRUE	FALSE
1982	4080	14549	269900	TRUE	TRUE
1983	4800	14549	255000	TRUE	TRUE
1984	3600	14549	57000	TRUE	FALSE
1985	4560	14549	78600	FALSE	FALSE
1986	4880	14549	70000	FALSE	FALSE
1987	2100	14549	79900	FALSE	FALSE
1988	5000	14549	176000	FALSE	FALSE
1989	6050	14549	104900	FALSE	FALSE
1990	5060	14549	170000	FALSE	FALSE
1991	5400	14549	173000	FALSE	FALSE

1992	6250	14549	192000	FALSE	TRUE
1993	4800	14549	130000	FALSE	FALSE
1994	5120	14549	179000	FALSE	FALSE
1995	5240	14549	179000	FALSE	FALSE
1996	5875	14549	172000	FALSE	FALSE
1997	5000	14549	165000	FALSE	FALSE
1998	4800	14549	75500	FALSE	FALSE
1999	4800	14549	148900	FALSE	FALSE
2000	5000	14549	115000	FALSE	FALSE
2001	6840	14549	147000	FALSE	TRUE
2002	4879	14549	120000	FALSE	FALSE
2003	5120	14549	155000	FALSE	TRUE
2004	4800	14549	70000	FALSE	FALSE
2005	5000	14549	153500	FALSE	TRUE
2006	4773	14549	111000	FALSE	FALSE
2007	6000	14549	136500	FALSE	FALSE
2008	5000	14549	110000	FALSE	FALSE
2009	5248	14549	97445	FALSE	FALSE
2010	16662	14549	325000	FALSE	FALSE
2011	7440	14549	149000	FALSE	FALSE
2012	8760	14549	162500	FALSE	TRUE
2013	5240	14549	149000	FALSE	FALSE
2014	15085	14549	81000	FALSE	FALSE
2015	7200	14549	160000	FALSE	FALSE
2016	9054	14549	155500	FALSE	FALSE
2017	4920	14549	118000	FALSE	FALSE
2018	6350	14549	139000	FALSE	FALSE
2019	7482	14549	107000	FALSE	FALSE
2020	7809	14549	130000	FALSE	FALSE
2021	10530	14549	165000	FALSE	TRUE
2022	5040	14549	82000	FALSE	FALSE
2023	6394	14549	113500	FALSE	FALSE
2024	6050	14549	86000	FALSE	FALSE
2025	6250	14549	144900	FALSE	TRUE
2026	6000	14549	129900	FALSE	FALSE
2027	5400	14549	99000	FALSE	FALSE
2028	7936	14549	135000	FALSE	FALSE
2029	5080	14549	120000	FALSE	FALSE
2030	8505	14549	92600	FALSE	FALSE
2031	8208	14549	143000	FALSE	FALSE
2032	7980	14549	110000	FALSE	FALSE
2033	7200	14549	105000	FALSE	FALSE
2034	4800	14549	105000	FALSE	FALSE

2035	5120	14549	134000	FALSE	TRUE
2036	5640	14549	82900	FALSE	FALSE
2037	6000	14549	124500	FALSE	TRUE
2038	5000	14549	140800	FALSE	FALSE
2039	6739	14549	154000	FALSE	FALSE
2040	5992	14549	99500	FALSE	FALSE
2041	2800	14549	110000	FALSE	FALSE
2042	5375	14549	100000	FALSE	FALSE
2043	4800	14549	10100	FALSE	FALSE
2044	3584	14549	61000	FALSE	FALSE
2045	3775	14549	229000	FALSE	TRUE
2046	2673	14549	9900	FALSE	FALSE
2047	3300	14549	145000	FALSE	FALSE
2048	6750	14549	164000	FALSE	FALSE
2049	18422	14549	6500	FALSE	FALSE
2050	4608	14549	110000	FALSE	FALSE
2051	5696	14549	125000	FALSE	TRUE
2052	2600	14549	129000	FALSE	FALSE
2053	5842	14549	103500	FALSE	FALSE
2054	4800	14549	110000	FALSE	FALSE
2055	6160	14549	32700	FALSE	FALSE
2056	4551	14549	103000	FALSE	FALSE
2057	5160	14549	119900	FALSE	FALSE
2058	9840	14549	84500	FALSE	FALSE
2059	4920	14549	131900	FALSE	FALSE
2060	8064	14549	140000	FALSE	FALSE
2061	5160	14549	85000	FALSE	FALSE
2062	3840	14549	21000	FALSE	FALSE
2063	4800	14549	25000	FALSE	FALSE
2064	4800	14549	32000	FALSE	FALSE
2065	3600	14549	30000	FALSE	FALSE
2066	3540	14549	69900	FALSE	FALSE
2067	4200	14549	50000	FALSE	FALSE
2068	5160	14549	79000	FALSE	FALSE
2069	8120	14549	60000	FALSE	FALSE
2070	3420	14549	49900	FALSE	FALSE
2071	5400	14549	134900	FALSE	FALSE
2072	4025	14549	85000	FALSE	FALSE
2073	4200	14549	55000	FALSE	FALSE
2074	4200	14549	121000	FALSE	FALSE
2075	4200	14549	67000	FALSE	FALSE
2076	4387	14549	70000	FALSE	FALSE
2077	3300	14549	80000	FALSE	FALSE

2078	3600	14549	75000	FALSE	FALSE
2079	5400	14549	100000	FALSE	FALSE
2080	3750	14549	35000	FALSE	FALSE
2081	2415	14549	61000	FALSE	FALSE
2082	7275	14549	107000	FALSE	FALSE
2083	5500	14549	25750	FALSE	FALSE
2084	30486	14549	279000	FALSE	FALSE
2085	9175	14549	237192	FALSE	FALSE
2086	6120	14549	75000	FALSE	FALSE
2087	5940	14549	105000	FALSE	FALSE
2088	15608	14549	146500	FALSE	FALSE
2089	8245	14549	16000	FALSE	FALSE
2090	5940	14549	75000	FALSE	FALSE
2091	10125	14549	119900	FALSE	FALSE
2092	7200	14549	84150	FALSE	FALSE
2093	6588	14549	94700	FALSE	FALSE
2094	7964	14549	87500	FALSE	FALSE
2095	7200	14549	53000	FALSE	FALSE
2096	8964	14549	101000	FALSE	FALSE
2097	5940	14549	75000	FALSE	FALSE
2098	7245	14549	117450	FALSE	FALSE
2099	10000	14549	141700	FALSE	FALSE
2100	6250	14549	130000	FALSE	FALSE
2101	5358	14549	99900	FALSE	FALSE
2102	5500	14549	169900	FALSE	FALSE
2103	4320	14549	154500	FALSE	FALSE
2104	3660	14549	54000	FALSE	FALSE
2105	5500	14549	103000	FALSE	FALSE
2106	4920	14549	170000	FALSE	FALSE
2107	4440	14549	179900	FALSE	FALSE
2108	5080	14549	176500	FALSE	FALSE
2109	4816	14549	108656	FALSE	FALSE
2110	4800	14549	161500	FALSE	FALSE
2111	4960	14549	134000	FALSE	FALSE
2112	6000	14549	238900	FALSE	FALSE
2113	5085	14549	224000	FALSE	FALSE
2114	4520	14549	205000	FALSE	FALSE
2115	3488	14549	50000	FALSE	FALSE
2116	6055	14549	138800	FALSE	FALSE
2117	3379	14549	117000	FALSE	FALSE
2118	4998	14549	182500	FALSE	TRUE
2119	4640	14549	97000	FALSE	FALSE
2120	4960	14549	134000	FALSE	FALSE

2121	4181	14549	78000	FALSE	FALSE
2122	8100	14549	180000	FALSE	FALSE
2123	5500	14549	144400	FALSE	TRUE
2124	6404	14549	141000	FALSE	FALSE
2125	3720	14549	109000	FALSE	FALSE
2126	4800	14549	168000	FALSE	FALSE
2127	3600	14549	32000	FALSE	FALSE
2128	5000	14549	108500	FALSE	FALSE
2129	4800	14549	69300	FALSE	FALSE
2130	16642	14549	460000	FALSE	FALSE
2131	4960	14549	193000	FALSE	FALSE
2132	8400	14549	239900	FALSE	TRUE
2133	10062	14549	179500	FALSE	FALSE
2134	6120	14549	166000	FALSE	FALSE
2135	6700	14549	157000	FALSE	TRUE
2136	4840	14549	115850	FALSE	FALSE
2137	7973	14549	124425	FALSE	FALSE
2138	5280	14549	160500	FALSE	FALSE
2139	6750	14549	94500	FALSE	FALSE
2140	4920	14549	147000	FALSE	FALSE
2141	6840	14549	83500	FALSE	FALSE
2142	6240	14549	142500	FALSE	FALSE
2143	7920	14549	127000	FALSE	FALSE
2144	8418	14549	180000	FALSE	FALSE
2145	5700	14549	156800	FALSE	FALSE
2146	5500	14549	158000	FALSE	FALSE
2147	11997	14549	172000	FALSE	FALSE
2148	6566	14549	160000	FALSE	FALSE
2149	5355	14549	148900	FALSE	FALSE
2150	7200	14549	146000	FALSE	FALSE
2151	6732	14549	159000	FALSE	FALSE
2152	9300	14549	139900	FALSE	FALSE
2153	5500	14549	132500	FALSE	FALSE
2154	7625	14549	184000	FALSE	FALSE
2155	6384	14549	118900	FALSE	FALSE
2156	6450	14549	150000	FALSE	FALSE
2157	6000	14549	130000	FALSE	FALSE
2158	5985	14549	150000	FALSE	FALSE
2159	8976	14549	126750	FALSE	FALSE
2160	5984	14549	116000	FALSE	FALSE
2161	5500	14549	134800	FALSE	FALSE
2162	6120	14549	143500	FALSE	FALSE
2163	8100	14549	160000	FALSE	FALSE

2164	9400	14549	129000	FALSE	FALSE
2165	5580	14549	120000	FALSE	FALSE
2166	6240	14549	129400	FALSE	FALSE
2167	5360	14549	148000	FALSE	FALSE
2168	6120	14549	114900	FALSE	FALSE
2169	6615	14549	130400	FALSE	FALSE
2170	6615	14549	125500	FALSE	FALSE
2171	8512	14549	45000	FALSE	FALSE
2172	10507	14549	101000	FALSE	FALSE
2173	3750	14549	112500	FALSE	FALSE
2174	3600	14549	57000	FALSE	FALSE
2175	4050	14549	80000	FALSE	FALSE
2176	4392	14549	65000	FALSE	FALSE
2177	3090	14549	72000	FALSE	FALSE
2178	3600	14549	60000	FALSE	FALSE
2180	11600	14549	220000	FALSE	FALSE
2181	4800	14549	115400	FALSE	FALSE
2182	4800	14549	174500	FALSE	FALSE
2183	4000	14549	115000	FALSE	FALSE
2184	5940	14549	165000	FALSE	FALSE
2185	5400	14549	127000	FALSE	FALSE
2186	5040	14549	129400	FALSE	FALSE
2187	6600	14549	115000	FALSE	FALSE
2188	6600	14549	140000	FALSE	FALSE
2189	5300	14549	122000	FALSE	FALSE
2190	8970	14549	285000	FALSE	FALSE
2191	9000	14549	254900	FALSE	FALSE
2192	4800	14549	148500	FALSE	FALSE
2193	11130	14549	98000	FALSE	FALSE
2194	5500	14549	145000	FALSE	FALSE
2195	4838	14549	142000	FALSE	FALSE
2196	6450	14549	150000	FALSE	FALSE
2197	6255	14549	163865	FALSE	FALSE
2198	7800	14549	169900	FALSE	FALSE
2199	5805	14549	138000	FALSE	FALSE
2200	6450	14549	172900	FALSE	FALSE
2201	6550	14549	160000	FALSE	FALSE
2202	4850	14549	124000	FALSE	FALSE
2203	6600	14549	163974	FALSE	FALSE
2204	6468	14549	133000	FALSE	FALSE
2205	8000	14549	135000	FALSE	FALSE
2206	4800	14549	102000	FALSE	FALSE
2207	14976	14549	108300	FALSE	FALSE

2208	6192	14549	111000	FALSE	FALSE
2209	4988	14549	95000	FALSE	FALSE
2210	4800	14549	95850	FALSE	FALSE
2211	7650	14549	159000	FALSE	FALSE
2212	5142	14549	98000	FALSE	FALSE
2213	3750	14549	117000	FALSE	FALSE
2214	3600	14549	200000	FALSE	TRUE
2215	7500	14549	113500	FALSE	FALSE
2216	3600	14549	54000	FALSE	FALSE
2217	4180	14549	25000	FALSE	FALSE
2218	4880	14549	148000	FALSE	TRUE
2219	6200	14549	168500	FALSE	FALSE
2220	4340	14549	221500	FALSE	FALSE
2221	4200	14549	200000	FALSE	TRUE
2222	3660	14549	89900	FALSE	FALSE
2223	7020	14549	130000	FALSE	FALSE
2224	3540	14549	225000	FALSE	FALSE
2225	4880	14549	100000	FALSE	FALSE
2226	3665	14549	113000	FALSE	FALSE
2227	4800	14549	159000	FALSE	FALSE
2228	7198	14549	130000	FALSE	FALSE
2229	4920	14549	115000	FALSE	FALSE
2230	4290	14549	127500	FALSE	FALSE
2231	5160	14549	232500	FALSE	TRUE
2232	3120	14549	125000	FALSE	FALSE
2233	4960	14549	119700	FALSE	FALSE
2234	3660	14549	37000	FALSE	FALSE
2235	4200	14549	85000	FALSE	FALSE
2237	2989	14549	90000	FALSE	FALSE
2238	3840	14549	97800	FALSE	FALSE
2239	4720	14549	89000	FALSE	FALSE
2240	3750	14549	12000	FALSE	FALSE
2241	3750	14549	12000	FALSE	FALSE
2242	4680	14549	30000	FALSE	FALSE
2243	10450	14549	79900	FALSE	FALSE
2244	4290	14549	24900	FALSE	FALSE
2245	3600	14549	112000	FALSE	FALSE
2246	5000	14549	9000	FALSE	FALSE
2247	6350	14579	21000	FALSE	FALSE
2248	4797	14579	37500	FALSE	FALSE
2249	7120	14579	23000	FALSE	FALSE
2250	4800	14579	12000	FALSE	FALSE
2251	5544	14579	21000	FALSE	FALSE

2252	4800	14579	9000	FALSE	FALSE
2253	8550	14579	13000	FALSE	FALSE
2254	5080	14579	70000	FALSE	FALSE
2255	10479	14579	165000	FALSE	FALSE
2256	4960	14579	120000	FALSE	FALSE
2257	5280	14579	96700	FALSE	FALSE
2258	5895	14579	17000	FALSE	FALSE
2259	5900	14579	119900	FALSE	FALSE
2260	5940	14579	87700	FALSE	FALSE
2261	5250	14579	89900	FALSE	FALSE
2262	3900	14579	91000	FALSE	FALSE
2263	4800	14579	85000	FALSE	FALSE
2264	4920	14579	45000	FALSE	FALSE
2265	6350	14579	50500	FALSE	FALSE
2266	4800	14579	85000	FALSE	FALSE
2267	5400	14579	114000	FALSE	FALSE
2268	6000	14579	21200	FALSE	FALSE
2269	6000	14579	58000	FALSE	FALSE
2270	5330	14579	85000	FALSE	FALSE
2271	5000	14579	80800	FALSE	FALSE
2272	5203	14579	32500	FALSE	FALSE
2273	10640	14579	72000	FALSE	FALSE
2274	5324	14579	52000	FALSE	FALSE
2275	7920	14579	89900	FALSE	FALSE
2276	8100	14579	48500	FALSE	FALSE
2277	8100	14579	86300	FALSE	FALSE
2278	8400	14579	45000	FALSE	FALSE
2279	5490	14579	44000	FALSE	FALSE
2280	4800	14579	18000	FALSE	FALSE
2281	4920	14579	45000	FALSE	FALSE
2282	2645	14579	124000	TRUE	FALSE
2283	5715	14579	250000	TRUE	FALSE
2284	5080	14579	276000	TRUE	TRUE
2285	4720	14579	223000	TRUE	TRUE
2286	5640	14579	119000	TRUE	FALSE
2287	3600	14579	49000	TRUE	FALSE
2288	4880	14579	465000	TRUE	TRUE
2289	3600	14579	330000	TRUE	TRUE
2290	2400	14579	209800	TRUE	TRUE
2292	4520	14579	93000	FALSE	FALSE
2293	3600	14579	20000	FALSE	FALSE
2294	10540	14579	45000	FALSE	FALSE
2295	6150	14579	172000	FALSE	FALSE

2296	5400	14579	143750	FALSE	FALSE
2297	5000	14579	157103	FALSE	TRUE
2298	10178	14579	250000	FALSE	FALSE
2299	5120	14579	144000	FALSE	FALSE
2300	8468	14579	180000	FALSE	FALSE
2301	4800	14579	102000	FALSE	FALSE
2302	9300	14579	146200	FALSE	FALSE
2303	10125	14579	150000	FALSE	FALSE
2304	5000	14579	86000	FALSE	FALSE
2305	5000	14579	175000	FALSE	FALSE
2306	9535	14579	141000	FALSE	FALSE
2307	4800	14579	110000	FALSE	FALSE
2308	9906	14579	120000	FALSE	FALSE
2309	6780	14579	99900	FALSE	FALSE
2310	5000	14579	105000	FALSE	FALSE
2311	3780	14579	9000	FALSE	FALSE
2312	3600	14579	74000	FALSE	FALSE
2313	9240	14579	61000	FALSE	FALSE
2314	4720	14579	99000	FALSE	FALSE
2315	6360	14579	33000	FALSE	FALSE
2316	5740	14579	35000	FALSE	FALSE
2317	7638	14579	14000	FALSE	FALSE
2318	3750	14579	34500	FALSE	FALSE
2319	3630	14579	100000	FALSE	FALSE
2320	3960	14579	15200	FALSE	FALSE
2321	4800	14579	55000	FALSE	FALSE
2322	4800	14579	90000	FALSE	FALSE
2323	7200	14579	30000	FALSE	FALSE
2324	5600	14579	89900	FALSE	FALSE
2325	5040	14579	120000	FALSE	FALSE
2326	5590	14579	101000	FALSE	FALSE
2327	5590	14579	101000	FALSE	FALSE
2328	3542	14579	92000	FALSE	FALSE
2329	4050	14579	64900	FALSE	FALSE
2330	5972	14579	88000	FALSE	FALSE
2331	5625	14579	55000	FALSE	FALSE
2332	2550	14579	38000	FALSE	FALSE
2333	4470	14579	59900	FALSE	FALSE
2334	4200	14579	115000	FALSE	FALSE
2335	3570	14579	36000	FALSE	FALSE
2336	4550	14579	115000	FALSE	FALSE
2337	3600	14579	103000	FALSE	FALSE
2338	6732	14579	79000	FALSE	FALSE

2341	2786	14579	75000	FALSE	FALSE
2342	18534	14579	47000	FALSE	FALSE
2343	8105	14579	66000	FALSE	FALSE
2344	9000	14579	169350	FALSE	FALSE
2345	7560	14579	60000	FALSE	FALSE
2346	7208	14579	184400	FALSE	TRUE
2347	9569	14579	49000	FALSE	FALSE
2348	10248	14579	70000	FALSE	FALSE
2349	6300	14579	38000	FALSE	FALSE
2350	5500	14579	10500	FALSE	FALSE
2351	7719	14579	101000	FALSE	FALSE
2352	5400	14579	210000	FALSE	FALSE
2353	6000	14579	105300	FALSE	FALSE
2354	4520	14579	164000	FALSE	FALSE
2355	5680	14579	110000	FALSE	FALSE
2356	3905	14579	33000	FALSE	FALSE
2357	6000	14579	135000	FALSE	FALSE
2358	8000	14579	187500	FALSE	TRUE
2359	5100	14579	113000	FALSE	FALSE
2360	5400	14579	84900	FALSE	FALSE
2361	5124	14579	94000	FALSE	FALSE
2362	5124	14579	46500	FALSE	FALSE
2363	5520	14579	87800	FALSE	FALSE
2364	6042	14579	119750	FALSE	FALSE
2365	5500	14579	50000	FALSE	FALSE
2366	5160	14579	74000	FALSE	FALSE
2367	8586	14579	400000	FALSE	TRUE
2368	3844	14579	54000	FALSE	FALSE
2369	5760	14579	155500	FALSE	FALSE
2370	6900	14579	125000	FALSE	FALSE
2371	8721	14579	101500	FALSE	FALSE
2372	8024	14579	120000	FALSE	FALSE
2373	5750	14579	170000	FALSE	TRUE
2374	8750	14579	173500	FALSE	TRUE
2375	5500	14579	105000	FALSE	FALSE
2376	8174	14579	173900	FALSE	TRUE
2377	7920	14579	128700	FALSE	FALSE
2378	8100	14579	139500	FALSE	FALSE
2379	8040	14579	137000	FALSE	FALSE
2380	6118	14579	50000	FALSE	FALSE
2381	6696	14579	103000	FALSE	FALSE
2382	2850	14579	70000	FALSE	FALSE
2383	3030	14579	61400	FALSE	FALSE

2384	30627	14579	180000	FALSE	FALSE
2385	5130	14579	137000	FALSE	FALSE
2386	4880	14579	90000	FALSE	FALSE
2387	4800	14579	138000	FALSE	FALSE
2388	9360	14579	260000	FALSE	FALSE
2389	7155	14579	65000	FALSE	FALSE
2390	6161	14579	157500	FALSE	FALSE
2391	4232	14579	134000	FALSE	FALSE
2392	3540	14579	117100	FALSE	FALSE
2393	5400	14579	173900	FALSE	FALSE
2394	4800	14579	137500	FALSE	FALSE
2395	4200	14579	120000	FALSE	FALSE
2396	3660	14579	102000	FALSE	FALSE
2397	4800	14579	155000	FALSE	FALSE
2398	10419	14579	72000	FALSE	FALSE
2399	3600	14579	205000	FALSE	TRUE
2400	7150	14579	224900	FALSE	TRUE
2401	5400	14579	100000	FALSE	FALSE
2402	4440	14579	210000	FALSE	FALSE
2403	20050	14579	149900	FALSE	FALSE
2404	6050	14579	225000	FALSE	TRUE
2405	3660	14579	115000	FALSE	FALSE
2406	2852	14579	8000	FALSE	FALSE
2407	4800	14579	80000	FALSE	FALSE
2408	3600	14579	19000	FALSE	FALSE
2409	3600	14579	24500	FALSE	FALSE
2410	4375	14579	29900	FALSE	FALSE
2411	2948	14579	23500	FALSE	FALSE
2412	3660	14579	15000	FALSE	FALSE
2413	3600	14579	8800	FALSE	FALSE
2414	4797	14610	105000	FALSE	FALSE
2415	13357	14610	145000	FALSE	FALSE
2416	8700	14610	835000	TRUE	TRUE
2417	2800	14610	179000	TRUE	FALSE
2418	7020	14610	919400	TRUE	TRUE
2419	4080	14610	239000	TRUE	FALSE
2420	4800	14610	297925	TRUE	FALSE
2421	5386	14610	224820	FALSE	FALSE
2422	5040	14610	151000	FALSE	FALSE
2423	5120	14610	138000	FALSE	FALSE
2424	5400	14610	81000	FALSE	FALSE
2425	4800	14610	177500	FALSE	FALSE
2426	5080	14610	122500	FALSE	FALSE

2427	5152	14610	108500	FALSE	FALSE
2428	3600	14610	265000	FALSE	FALSE
2429	3600	14610	60000	FALSE	FALSE
2430	3750	14610	28000	FALSE	FALSE
2431	4800	14610	94800	FALSE	FALSE
2432	4800	14610	100000	FALSE	FALSE
2433	5160	14610	84900	FALSE	FALSE
2434	5760	14610	117500	FALSE	FALSE
2435	3510	14610	75000	FALSE	FALSE
2436	5928	14610	151000	FALSE	FALSE
2437	3720	14610	11500	FALSE	FALSE
2438	3720	14610	114900	FALSE	FALSE
2439	4840	14610	121500	FALSE	FALSE
2440	6000	14610	103500	FALSE	FALSE
2441	4800	14610	235000	FALSE	FALSE
2442	7772	14610	183000	FALSE	TRUE
2443	7980	14610	78300	FALSE	FALSE
2444	7490	14610	124000	FALSE	FALSE
2445	7475	14610	144000	FALSE	FALSE
2446	8040	14610	145000	FALSE	FALSE
2447	7200	14610	179600	FALSE	FALSE
2448	7200	14610	156000	FALSE	FALSE
2449	17088	14610	127000	FALSE	FALSE
2450	4875	14610	135000	FALSE	FALSE
2451	3000	14610	20000	FALSE	FALSE
2452	3600	14610	100000	FALSE	FALSE
2453	3000	14610	103000	FALSE	FALSE
2454	6708	14610	140700	FALSE	FALSE
2455	7800	14610	133900	FALSE	FALSE
2456	4800	14610	93500	FALSE	FALSE
2457	7655	14610	139900	FALSE	FALSE
2458	6000	14610	134000	FALSE	FALSE
2459	4760	14610	112000	FALSE	FALSE
2460	4640	14610	179000	FALSE	FALSE
2461	4560	14610	216900	FALSE	TRUE
2462	6780	14610	202500	FALSE	FALSE
2463	8190	14610	154300	FALSE	FALSE
2464	3870	14610	190000	FALSE	FALSE
2465	2610	14610	125000	FALSE	FALSE
2466	4800	14641	48000	FALSE	FALSE
2467	5740	14641	72500	FALSE	FALSE
2468	7200	14641	75000	FALSE	FALSE
2469	17085	14641	89000	FALSE	FALSE

2470	2500	14641	145000	TRUE	TRUE
2471	3600	14641	210000	TRUE	TRUE
2472	7050	14641	313000	TRUE	TRUE
2473	4080	14641	184000	TRUE	FALSE
2474	22400	14641	320000	FALSE	FALSE
2475	4800	14641	142900	FALSE	FALSE
2476	7560	14641	154900	FALSE	FALSE
2477	4800	14641	110000	FALSE	FALSE
2478	4800	14641	115000	FALSE	FALSE
2479	4859	14641	195000	FALSE	TRUE
2480	5500	14641	125000	FALSE	FALSE
2481	19080	14641	250000	FALSE	TRUE
2482	7200	14641	161000	FALSE	FALSE
2483	7614	14641	150000	FALSE	FALSE
2484	7440	14641	149000	FALSE	TRUE
2485	6120	14641	131500	FALSE	FALSE
2486	6160	14641	110000	FALSE	FALSE
2487	4800	14641	119000	FALSE	FALSE
2488	7560	14641	91000	FALSE	FALSE
2489	5640	14641	85000	FALSE	FALSE
2490	2000	14641	132000	FALSE	TRUE
2491	5696	14641	64000	FALSE	FALSE
2492	4375	14641	11210	FALSE	FALSE
2493	3600	14641	152500	FALSE	FALSE
2494	3750	14641	16800	FALSE	FALSE
2495	4800	14641	158000	FALSE	FALSE
2496	4773	14641	135000	FALSE	FALSE
2497	5600	14641	95900	FALSE	FALSE
2498	3588	14641	78000	FALSE	FALSE
2499	3840	14641	104000	FALSE	FALSE
2500	4720	14641	31000	FALSE	FALSE
2501	4810	14641	90000	FALSE	FALSE
2502	5805	14641	49000	FALSE	FALSE
2503	3600	14641	100000	FALSE	FALSE
2504	4200	14641	34500	FALSE	FALSE
2505	7650	14641	165000	FALSE	FALSE
2506	8085	14641	120000	FALSE	FALSE
2507	3660	14641	135000	FALSE	FALSE
2508	3480	14641	81000	FALSE	FALSE
2509	4800	14641	109000	FALSE	FALSE
2510	4840	14641	87300	FALSE	FALSE
2511	5350	14641	48000	FALSE	FALSE
2512	4400	14641	174500	FALSE	TRUE

2513	4520	14641	105000	FALSE	FALSE
2514	6475	14641	162500	FALSE	FALSE
2515	3780	14641	140500	FALSE	FALSE
2516	5330	14641	99000	FALSE	FALSE
2517	5289	14641	145000	FALSE	FALSE
2518	4816	14641	86200	FALSE	FALSE
2519	4218	14641	147000	FALSE	FALSE
2520	6110	14641	156700	FALSE	FALSE
2521	7980	14641	189900	FALSE	FALSE
2522	6750	14641	133900	FALSE	TRUE
2523	6250	14641	130000	FALSE	FALSE
2524	7440	14641	170000	FALSE	FALSE
2525	9945	14641	139000	FALSE	FALSE
2526	6750	14641	127000	FALSE	FALSE
2527	5184	14641	96000	FALSE	FALSE
2528	7645	14641	122000	FALSE	FALSE
2529	2030	14641	54900	FALSE	FALSE
2530	5664	14641	170000	FALSE	FALSE
2531	12584	14641	130000	FALSE	FALSE
2532	9000	14641	110000	FALSE	FALSE
2533	7800	14641	175000	FALSE	FALSE
2534	6000	14641	148000	FALSE	FALSE
2535	5000	14641	108000	FALSE	FALSE
2536	5590	14641	175000	FALSE	TRUE
2537	4815	14641	185000	FALSE	TRUE
2538	4800	14641	144000	FALSE	FALSE
2539	4800	14641	179000	FALSE	FALSE
2540	3660	14641	55000	FALSE	FALSE
2541	5400	14641	135000	FALSE	FALSE
2542	3150	14641	165000	FALSE	FALSE
2543	3660	14641	97000	FALSE	FALSE
2544	5681	14641	189000	FALSE	FALSE
2545	5424	14641	57000	FALSE	FALSE
2546	4800	14669	97000	FALSE	FALSE
2547	5120	14669	115000	FALSE	FALSE
2548	3600	14669	75000	FALSE	FALSE
2549	5680	14669	84000	FALSE	FALSE
2550	4200	14669	25000	FALSE	FALSE
2551	6160	14669	136000	FALSE	FALSE
2552	8190	14669	112000	FALSE	FALSE
2553	5000	14669	129000	FALSE	FALSE
2554	4800	14669	79000	FALSE	FALSE
2555	5080	14669	82000	FALSE	FALSE

2556	4800	14669	79000	FALSE	FALSE
2557	9760	14669	97000	FALSE	FALSE
2558	5080	14669	80000	FALSE	FALSE
2559	7488	14669	75000	FALSE	FALSE
2560	6885	14669	125000	FALSE	FALSE
2561	8625	14669	130000	FALSE	FALSE
2562	4720	14669	246600	TRUE	TRUE
2563	3600	14669	167900	TRUE	TRUE
2564	4800	14669	223000	TRUE	TRUE
2565	8880	14669	525000	TRUE	TRUE
2566	4880	14669	357900	TRUE	TRUE
2567	6650	14669	287500	TRUE	TRUE
2568	4880	14669	365000	TRUE	TRUE
2569	4020	14669	270000	TRUE	TRUE
2570	6890	14669	290000	TRUE	TRUE
2571	5120	14669	165000	FALSE	FALSE
2572	5000	14669	169900	FALSE	FALSE
2573	6250	14669	195000	FALSE	TRUE
2574	5000	14669	158500	FALSE	FALSE
2575	4800	14669	145000	FALSE	FALSE
2576	4800	14669	105800	FALSE	FALSE
2577	6000	14669	122400	FALSE	FALSE
2578	5200	14669	127000	FALSE	FALSE
2579	4800	14669	153000	FALSE	FALSE
2580	4800	14669	110000	FALSE	FALSE
2581	4800	14669	203000	FALSE	TRUE
2582	10461	14669	200000	FALSE	FALSE
2583	7500	14669	118000	FALSE	FALSE
2584	7200	14669	134000	FALSE	FALSE
2585	10950	14669	155000	FALSE	FALSE
2586	5080	14669	119500	FALSE	FALSE
2587	8256	14669	114000	FALSE	FALSE
2588	5715	14669	114900	FALSE	FALSE
2589	8400	14669	121000	FALSE	FALSE
2590	5508	14669	94000	FALSE	FALSE
2591	4800	14669	110000	FALSE	FALSE
2592	4500	14669	56700	FALSE	FALSE
2593	3450	14669	46000	FALSE	FALSE
2594	3450	14669	135000	FALSE	TRUE
2595	4800	14669	64000	FALSE	FALSE
2596	5535	14669	105000	FALSE	FALSE
2597	4320	14669	84000	FALSE	FALSE
2598	4800	14669	58500	FALSE	FALSE

2599	3510	14669	35100	FALSE	FALSE
2600	5449	14669	113000	FALSE	FALSE
2601	6600	14669	29900	FALSE	FALSE
2602	4768	14669	125000	FALSE	FALSE
2603	9375	14669	156000	FALSE	FALSE
2604	12975	14669	100000	FALSE	FALSE
2605	6600	14669	119000	FALSE	FALSE
2606	7200	14669	95000	FALSE	FALSE
2607	7200	14669	114500	FALSE	FALSE
2608	23205	14669	85000	FALSE	FALSE
2609	7200	14669	110000	FALSE	FALSE
2610	4800	14669	195000	FALSE	FALSE
2611	5400	14669	156900	FALSE	FALSE
2612	5600	14669	133500	FALSE	FALSE
2613	5160	14669	138000	FALSE	FALSE
2614	4320	14669	15000	FALSE	FALSE
2615	5400	14669	169000	FALSE	FALSE
2616	7140	14669	145000	FALSE	TRUE
2617	4520	14669	146500	FALSE	FALSE
2618	4800	14669	125000	FALSE	FALSE
2619	4800	14669	173000	FALSE	FALSE
2620	3034	14669	165000	FALSE	FALSE
2621	5400	14669	205000	FALSE	TRUE
2622	3488	14669	138600	FALSE	FALSE
2623	2960	14669	119950	FALSE	FALSE
2624	4080	14669	175000	FALSE	FALSE
2625	6500	14669	150000	FALSE	FALSE
2626	8640	14669	162500	FALSE	FALSE
2627	7440	14669	133000	FALSE	FALSE
2628	4800	14669	136000	FALSE	FALSE
2629	7980	14669	167900	FALSE	FALSE
2630	13650	14669	115000	FALSE	FALSE
2631	8100	14669	156500	FALSE	FALSE
2632	7920	14669	148000	FALSE	FALSE
2633	7200	14669	120000	FALSE	FALSE
2634	5750	14669	133000	FALSE	FALSE
2635	7200	14669	145000	FALSE	FALSE
2636	7260	14669	135100	FALSE	FALSE
2637	6138	14669	136300	FALSE	FALSE
2638	6000	14669	125000	FALSE	FALSE
2639	7488	14669	145000	FALSE	FALSE
2640	9176	14669	143500	FALSE	FALSE
2641	7920	14669	115000	FALSE	FALSE

2642	4920	14669	110000	FALSE	FALSE
2643	3030	14669	90000	FALSE	FALSE
2644	7000	14669	57000	FALSE	FALSE
2645	9000	14669	255000	FALSE	FALSE
2646	7950	14669	110500	FALSE	FALSE
2647	16320	14669	243000	FALSE	FALSE
2648	8100	14669	193000	FALSE	FALSE
2649	4760	14669	160000	FALSE	FALSE
2650	6480	14669	179900	FALSE	FALSE
2651	13860	14669	206000	FALSE	FALSE
2652	6084	14669	134000	FALSE	FALSE
2653	5900	14669	137000	FALSE	FALSE
2654	6000	14669	156000	FALSE	FALSE
2655	7500	14669	96000	FALSE	FALSE
2656	6708	14669	85000	FALSE	FALSE
2657	6832	14669	154000	FALSE	FALSE
2658	7182	14669	141700	FALSE	FALSE
2659	4860	14669	138000	FALSE	FALSE
2660	4560	14669	222500	FALSE	TRUE
2661	4900	14669	80000	FALSE	FALSE
2662	6710	14669	160000	FALSE	FALSE
2663	3600	14669	224210	FALSE	TRUE
2664	3800	14669	192000	FALSE	TRUE
2665	10425	14669	178800	FALSE	FALSE
2666	3010	14669	15900	FALSE	FALSE
2667	5187	14669	101000	FALSE	FALSE
2668	6029	14669	135960	FALSE	FALSE
2669	5160	14700	97700	FALSE	FALSE
2670	8710	14700	90000	FALSE	FALSE
2671	5896	14700	32000	FALSE	FALSE
2672	5040	14700	70000	FALSE	FALSE
2673	6150	14700	90000	FALSE	FALSE
2674	6000	14700	102000	FALSE	FALSE
2675	8280	14700	150000	FALSE	FALSE
2676	8775	14700	55000	FALSE	FALSE
2677	4800	14700	83000	FALSE	FALSE
2678	4920	14700	112500	FALSE	FALSE
2679	5590	14700	101000	FALSE	FALSE
2680	4800	14700	139000	FALSE	FALSE
2681	5246	14700	123000	FALSE	FALSE
2682	5400	14700	90000	FALSE	FALSE
2683	4800	14700	77000	FALSE	FALSE
2684	4800	14700	95000	FALSE	FALSE

2685	5664	14700	85325	FALSE	FALSE
2686	7236	14700	104000	FALSE	FALSE
2687	5400	14700	88000	FALSE	FALSE
2688	5040	14700	81000	FALSE	FALSE
2689	7700	14700	98000	FALSE	FALSE
2690	7200	14700	77000	FALSE	FALSE
2691	6344	14700	79000	FALSE	FALSE
2692	6000	14700	77000	FALSE	FALSE
2694	4800	14700	92000	FALSE	FALSE
2695	21780	14700	131000	FALSE	FALSE
2696	4556	14700	198000	TRUE	TRUE
2697	3876	14700	200000	TRUE	FALSE
2698	3600	14700	172500	TRUE	TRUE
2699	4200	14700	301000	TRUE	TRUE
2700	3405	14700	205000	TRUE	TRUE
2701	4100	14700	277000	TRUE	TRUE
2702	4800	14700	248000	TRUE	TRUE
2703	3600	14700	105000	TRUE	TRUE
2704	5760	14700	40000	FALSE	FALSE
2705	5280	14700	180500	FALSE	FALSE
2706	10000	14700	188000	FALSE	FALSE
2707	7000	14700	160000	FALSE	FALSE
2708	5120	14700	170000	FALSE	TRUE
2709	5480	14700	108000	FALSE	FALSE
2710	5000	14700	167000	FALSE	TRUE
2711	7000	14700	154000	FALSE	FALSE
2712	6050	14700	112000	FALSE	FALSE
2713	5106	14700	135000	FALSE	FALSE
2714	5120	14700	149000	FALSE	FALSE
2715	6000	14700	154900	FALSE	FALSE
2716	6000	14700	161000	FALSE	TRUE
2717	5720	14700	118500	FALSE	FALSE
2718	8710	14700	200000	FALSE	FALSE
2719	19080	14700	284000	FALSE	TRUE
2720	12000	14700	200000	FALSE	FALSE
2721	15701	14700	163900	FALSE	FALSE
2722	7345	14700	102800	FALSE	FALSE
2723	4800	14700	92000	FALSE	FALSE
2724	10125	14700	165500	FALSE	TRUE
2725	7260	14700	104000	FALSE	FALSE
2726	4800	14700	89000	FALSE	FALSE
2727	6120	14700	103000	FALSE	FALSE
2728	7600	14700	120000	FALSE	FALSE

2729	6750	14700	111000	FALSE	FALSE
2730	5040	14700	146500	FALSE	FALSE
2731	5040	14700	146500	FALSE	FALSE
2732	6550	14700	120000	FALSE	FALSE
2733	5200	14700	98000	FALSE	FALSE
2734	6897	14700	101500	FALSE	FALSE
2735	5080	14700	120000	FALSE	FALSE
2736	5460	14700	148000	FALSE	FALSE
2737	3795	14700	65000	FALSE	FALSE
2738	4680	14700	168900	FALSE	FALSE
2739	5400	14700	72600	FALSE	FALSE
2740	11250	14700	88100	FALSE	FALSE
2741	4840	14700	97000	FALSE	FALSE
2742	4800	14700	89800	FALSE	FALSE
2743	4914	14700	65000	FALSE	FALSE
2744	5166	14700	44000	FALSE	FALSE
2745	4920	14700	155000	FALSE	FALSE
2746	5124	14700	65000	FALSE	FALSE
2747	5166	14700	82500	FALSE	FALSE
2748	4800	14700	70000	FALSE	FALSE
2749	5280	14700	40000	FALSE	FALSE
2750	6160	14700	57500	FALSE	FALSE
2751	5934	14700	67000	FALSE	FALSE
2752	5000	14700	100000	FALSE	FALSE
2753	3420	14700	69900	FALSE	FALSE
2754	5040	14700	120000	FALSE	FALSE
2755	3600	14700	100000	FALSE	FALSE
2756	8040	14700	124000	FALSE	FALSE
2757	3200	14700	140000	FALSE	FALSE
2758	3332	14700	108500	FALSE	FALSE
2759	4200	14700	90000	FALSE	FALSE
2760	3600	14700	94000	FALSE	FALSE
2761	3290	14700	57000	FALSE	FALSE
2762	3560	14700	80000	FALSE	FALSE
2763	9042	14700	35000	FALSE	FALSE
2764	9000	14700	160000	FALSE	FALSE
2765	8750	14700	99000	FALSE	FALSE
2766	10125	14700	129000	FALSE	FALSE
2767	10353	14700	113000	FALSE	FALSE
2768	7200	14700	134000	FALSE	FALSE
2769	5520	14700	170000	FALSE	FALSE
2770	6254	14700	160000	FALSE	FALSE
2771	5400	14700	148500	FALSE	FALSE

2772	5580	14700	125800	FALSE	FALSE
2773	5200	14700	138000	FALSE	FALSE
2774	5364	14700	130000	FALSE	FALSE
2775	4800	14700	168500	FALSE	FALSE
2776	5950	14700	129000	FALSE	FALSE
2777	4982	14700	180000	FALSE	FALSE
2778	2928	14700	173000	FALSE	FALSE
2779	3710	14700	114750	FALSE	FALSE
2780	5936	14700	190000	FALSE	FALSE
2781	6000	14700	143500	FALSE	FALSE
2782	5355	14700	196770	FALSE	FALSE
2783	3295	14700	110000	FALSE	FALSE
2784	4960	14700	108000	FALSE	FALSE
2785	5400	14700	175000	FALSE	TRUE
2786	6000	14700	118000	FALSE	FALSE
2787	3840	14700	82000	FALSE	FALSE
2788	4750	14700	135000	FALSE	FALSE
2789	3840	14700	96000	FALSE	FALSE
2790	6300	14700	120000	FALSE	FALSE
2791	4600	14700	92000	FALSE	FALSE
2792	5500	14700	126500	FALSE	FALSE
2793	5200	14700	135000	FALSE	TRUE
2794	12000	14700	123000	FALSE	FALSE
2795	5400	14700	100000	FALSE	FALSE
2796	4800	14700	185000	FALSE	FALSE
2797	5150	14700	138000	FALSE	FALSE
2798	5400	14700	132800	FALSE	FALSE
2799	7920	14700	140500	FALSE	FALSE
2800	8909	14700	350000	FALSE	FALSE
2801	12035	14700	270000	FALSE	FALSE
2802	4900	14700	163400	FALSE	FALSE
2803	3750	14700	120000	FALSE	FALSE
2804	8999	14700	158000	FALSE	FALSE
2805	5289	14700	155000	FALSE	FALSE
2806	5500	14700	125000	FALSE	FALSE
2807	5500	14700	169000	FALSE	FALSE
2808	6120	14700	147400	FALSE	FALSE
2809	9536	14700	149000	FALSE	FALSE
2810	8060	14700	147000	FALSE	FALSE
2811	4884	14700	135000	FALSE	FALSE
2812	6550	14700	174000	FALSE	TRUE
2813	6681	14700	129000	FALSE	FALSE
2814	6000	14700	170500	FALSE	TRUE

2815	7140	14700	148000	FALSE	FALSE
2816	7070	14700	70000	FALSE	FALSE
2817	7232	14700	138000	FALSE	FALSE
2818	5600	14700	139900	FALSE	FALSE
2819	5808	14700	109400	FALSE	FALSE
2820	4928	14700	109000	FALSE	FALSE
2821	7452	14700	140000	FALSE	FALSE
2822	7920	14700	140000	FALSE	FALSE
2823	6405	14700	127500	FALSE	FALSE
2824	8646	14700	137500	FALSE	FALSE
2825	7200	14700	167500	FALSE	TRUE
2826	6750	14700	128000	FALSE	FALSE
2827	6240	14700	141000	FALSE	FALSE
2828	6831	14700	85000	FALSE	FALSE
2829	5280	14700	98000	FALSE	FALSE
2830	6336	14700	80000	FALSE	FALSE
2831	7473	14700	90000	FALSE	FALSE
2832	5400	14700	99800	FALSE	FALSE
2833	5920	14700	154000	FALSE	FALSE
2834	5440	14700	109900	FALSE	FALSE
2835	8320	14700	233000	FALSE	FALSE
2836	5080	14700	175000	FALSE	FALSE
2837	3750	14700	95000	FALSE	FALSE
2838	3500	14700	25000	FALSE	FALSE
2839	2040	14700	51000	FALSE	FALSE
2840	3500	14700	66500	FALSE	FALSE
2841	3500	14700	75000	FALSE	FALSE
2842	3600	14700	85000	FALSE	FALSE
2843	4725	14700	145000	FALSE	FALSE
2844	4080	14700	156000	FALSE	FALSE
2845	5040	14700	147000	FALSE	FALSE
2846	4800	14700	125000	FALSE	FALSE
2847	5805	14700	119000	FALSE	FALSE
2848	7900	14700	120000	FALSE	FALSE
2849	6450	14700	108300	FALSE	FALSE
2850	5400	14700	100000	FALSE	FALSE
2851	4800	14700	108000	FALSE	FALSE
2852	5900	14700	163500	FALSE	FALSE
2853	4200	14700	124100	FALSE	FALSE
2854	13130	14700	221000	FALSE	TRUE
2855	7800	14700	169000	FALSE	FALSE
2856	6720	14700	145900	FALSE	FALSE
2857	5330	14700	126500	FALSE	FALSE

2858	5125	14700	109900	FALSE	FALSE
2859	4760	14700	91500	FALSE	FALSE
2860	5400	14700	127000	FALSE	FALSE
2861	3949	14700	87000	FALSE	FALSE
2862	3540	14700	300000	FALSE	TRUE
2863	7000	14700	150500	FALSE	FALSE
2864	3600	14700	58000	FALSE	FALSE
2865	4800	14700	195000	FALSE	TRUE
2866	6384	14700	180000	FALSE	TRUE
2867	3390	14700	117000	FALSE	FALSE
2868	3795	14700	174000	FALSE	TRUE
2869	3750	14700	143000	FALSE	TRUE
2870	4800	14700	220000	FALSE	TRUE
2871	5320	14700	162000	FALSE	FALSE
2872	3600	14700	157500	FALSE	FALSE
2873	3720	14700	145000	FALSE	FALSE
2874	4880	14700	85000	FALSE	FALSE
2875	3120	14700	160000	FALSE	FALSE
2876	3000	14700	168000	FALSE	FALSE
2877	4800	14700	205000	FALSE	TRUE
2878	4998	14700	144000	FALSE	FALSE
2879	1800	14700	72000	FALSE	FALSE
2880	6030	14700	90000	FALSE	FALSE
2881	7500	14700	165000	FALSE	FALSE
2882	3600	14700	29500	FALSE	FALSE
2883	3780	14700	17000	FALSE	FALSE
2884	4560	14700	33000	FALSE	FALSE
2886	4800	14730	100000	FALSE	FALSE
2887	7068	14730	127000	FALSE	FALSE
2888	5984	14730	89000	FALSE	FALSE
2889	4932	14730	77000	FALSE	FALSE
2890	6750	14730	140000	FALSE	FALSE
2891	3600	14730	47000	FALSE	FALSE
2892	4920	14730	118000	FALSE	FALSE
2893	5360	14730	89000	FALSE	FALSE
2894	5880	14730	134000	FALSE	FALSE
2895	4800	14730	130000	FALSE	FALSE
2896	4800	14730	109900	FALSE	FALSE
2897	6125	14730	100000	FALSE	FALSE
2898	5460	14730	70000	FALSE	FALSE
2899	7215	14730	73000	FALSE	FALSE
2900	6000	14730	113500	FALSE	FALSE
2901	8513	14730	82000	FALSE	FALSE

2902	6000	14730	72800	FALSE	FALSE
2903	2540	14730	149700	TRUE	TRUE
2904	3000	14730	108000	TRUE	TRUE
2905	4800	14730	338000	TRUE	TRUE
2906	3600	14730	260000	TRUE	TRUE
2907	4920	14730	150000	FALSE	FALSE
2908	4800	14730	207500	FALSE	FALSE
2909	7316	14730	213000	FALSE	FALSE
2910	5400	14730	188000	FALSE	TRUE
2911	5000	14730	193500	FALSE	TRUE
2912	5080	14730	112000	FALSE	FALSE
2913	9600	14730	123000	FALSE	FALSE
2914	5000	14730	158000	FALSE	FALSE
2915	5060	14730	145000	FALSE	FALSE
2916	5000	14730	125000	FALSE	FALSE
2917	7200	14730	110000	FALSE	FALSE
2918	4914	14730	180000	FALSE	FALSE
2919	5000	14730	197000	FALSE	FALSE
2920	4800	14730	127000	FALSE	FALSE
2921	4800	14730	150000	FALSE	FALSE
2922	5400	14730	195000	FALSE	FALSE
2923	16090	14730	264000	FALSE	FALSE
2924	7200	14730	195000	FALSE	FALSE
2925	7680	14730	153000	FALSE	FALSE
2926	9100	14730	148000	FALSE	FALSE
2927	7630	14730	129400	FALSE	FALSE
2928	11025	14730	147500	FALSE	FALSE
2929	6890	14730	113900	FALSE	FALSE
2930	10125	14730	168500	FALSE	TRUE
2931	7680	14730	135000	FALSE	FALSE
2932	5520	14730	85000	FALSE	FALSE
2933	8400	14730	101000	FALSE	FALSE
2934	7424	14730	114000	FALSE	FALSE
2935	5080	14730	124000	FALSE	FALSE
2936	4800	14730	130000	FALSE	FALSE
2937	6000	14730	91000	FALSE	FALSE
2938	21780	14730	45000	FALSE	FALSE
2939	6120	14730	128000	FALSE	FALSE
2940	6000	14730	137500	FALSE	FALSE
2941	5800	14730	125000	FALSE	FALSE
2942	8040	14730	132000	FALSE	FALSE
2943	8400	14730	120000	FALSE	FALSE
2944	4800	14730	120000	FALSE	FALSE

2945	4480	14730	68000	FALSE	FALSE
2946	6120	14730	145000	FALSE	FALSE
2947	5679	14730	302500	FALSE	FALSE
2948	5500	14730	219500	FALSE	TRUE
2949	3500	14730	34900	FALSE	FALSE
2950	9652	14730	10000	FALSE	FALSE
2951	3775	14730	99000	FALSE	FALSE
2952	3300	14730	125000	FALSE	FALSE
2953	2500	14730	155000	FALSE	TRUE
2954	3270	14730	24500	FALSE	FALSE
2955	3480	14730	139000	FALSE	TRUE
2956	4992	14730	167000	FALSE	FALSE
2957	5400	14730	120000	FALSE	FALSE
2958	4920	14730	60000	FALSE	FALSE
2959	5805	14730	50000	FALSE	FALSE
2960	4800	14730	117000	FALSE	FALSE
2961	5000	14730	69700	FALSE	FALSE
2962	3600	14730	67500	FALSE	FALSE
2963	2752	14730	44500	FALSE	FALSE
2964	5080	14730	115000	FALSE	FALSE
2965	3600	14730	96000	FALSE	FALSE
2966	4880	14730	120000	FALSE	FALSE
2967	4165	14730	110000	FALSE	FALSE
2968	4200	14730	80000	FALSE	FALSE
2969	4200	14730	80000	FALSE	FALSE
2970	2700	14730	86500	FALSE	FALSE
2971	2700	14730	86500	FALSE	FALSE
2972	10607	14730	60000	FALSE	FALSE
2973	11421	14730	194000	FALSE	FALSE
2974	7200	14730	145500	FALSE	FALSE
2975	10000	14730	115000	FALSE	FALSE
2976	12900	14730	99000	FALSE	FALSE
2977	19185	14730	70000	FALSE	FALSE
2978	7200	14730	114000	FALSE	FALSE
2979	5124	14730	180000	FALSE	FALSE
2980	5460	14730	110000	FALSE	FALSE
2981	4800	14730	155200	FALSE	FALSE
2982	6075	14730	112000	FALSE	FALSE
2983	5076	14730	215000	FALSE	FALSE
2984	5400	14730	157000	FALSE	FALSE
2985	7336	14730	199000	FALSE	FALSE
2986	4690	14730	116000	FALSE	FALSE
2987	3600	14730	75000	FALSE	FALSE

2988	4800	14730	208500	FALSE	FALSE
2989	5000	14730	208000	FALSE	FALSE
2990	5125	14730	110000	FALSE	FALSE
2991	4760	14730	100000	FALSE	FALSE
2992	4800	14730	107000	FALSE	FALSE
2993	4290	14730	126500	FALSE	FALSE
2994	5490	14730	117900	FALSE	FALSE
2995	6950	14730	135000	FALSE	FALSE
2996	4680	14730	113000	FALSE	FALSE
2997	5300	14730	111900	FALSE	FALSE
2998	5490	14730	81500	FALSE	FALSE
2999	4600	14730	280000	FALSE	FALSE
3000	3480	14730	80000	FALSE	FALSE
3001	4760	14730	161200	FALSE	FALSE
3002	4960	14730	156000	FALSE	FALSE
3003	8400	14730	125000	FALSE	FALSE
3004	5400	14730	88500	FALSE	FALSE
3005	4920	14730	111000	FALSE	FALSE
3006	6500	14730	183500	FALSE	FALSE
3007	9225	14730	315000	FALSE	FALSE
3008	4480	14730	115500	FALSE	FALSE
3009	14534	14730	220000	FALSE	FALSE
3010	7625	14730	175000	FALSE	FALSE
3011	7072	14730	159000	FALSE	FALSE
3012	6800	14730	135000	FALSE	FALSE
3013	4880	14730	128000	FALSE	FALSE
3014	5000	14730	121840	FALSE	FALSE
3015	7700	14730	267500	FALSE	FALSE
3016	13632	14730	181800	FALSE	FALSE
3017	11220	14730	242000	FALSE	FALSE
3018	7750	14730	198900	FALSE	TRUE
3019	6188	14730	168000	FALSE	TRUE
3020	7410	14730	257000	FALSE	FALSE
3021	7840	14730	180000	FALSE	FALSE
3022	6240	14730	139900	FALSE	FALSE
3023	8370	14730	155000	FALSE	FALSE
3024	5500	14730	144900	FALSE	FALSE
3025	4884	14730	131000	FALSE	FALSE
3026	6630	14730	139000	FALSE	FALSE
3027	9536	14730	85000	FALSE	FALSE
3028	7920	14730	138000	FALSE	FALSE
3029	7245	14730	133000	FALSE	FALSE
3030	6960	14730	134900	FALSE	FALSE

3031	7700	14730	137000	FALSE	FALSE
3032	8040	14730	145000	FALSE	FALSE
3033	3750	14730	42000	FALSE	FALSE
3034	1480	14730	57000	FALSE	FALSE
3035	2820	14730	50000	FALSE	FALSE
3036	4050	14730	90000	FALSE	FALSE
3037	3920	14730	46500	FALSE	FALSE
3039	5125	14730	134900	FALSE	FALSE
3040	4800	14730	160100	FALSE	FALSE
3041	7224	14730	130000	FALSE	FALSE
3042	4212	14730	107500	FALSE	FALSE
3043	4800	14730	115000	FALSE	FALSE
3044	4800	14730	148500	FALSE	FALSE
3045	4856	14730	123500	FALSE	FALSE
3046	18720	14730	210000	FALSE	FALSE
3047	19184	14730	195000	FALSE	FALSE
3048	8479	14730	168000	FALSE	FALSE
3049	6500	14730	154500	FALSE	FALSE
3050	6240	14730	127500	FALSE	FALSE
3051	6000	14730	157000	FALSE	FALSE
3052	9891	14730	154800	FALSE	FALSE
3053	8250	14730	150000	FALSE	FALSE
3054	5900	14730	130000	FALSE	FALSE
3055	7200	14730	105000	FALSE	FALSE
3056	5250	14730	108000	FALSE	FALSE
3057	5200	14730	86000	FALSE	FALSE
3058	5125	14730	83000	FALSE	FALSE
3059	7320	14730	131400	FALSE	FALSE
3060	3600	14730	145000	FALSE	FALSE
3061	5160	14730	168000	FALSE	FALSE
3062	7245	14730	129000	FALSE	FALSE
3063	4000	14730	107400	FALSE	FALSE
3064	5280	14730	190000	FALSE	FALSE
3065	3600	14730	37000	FALSE	FALSE
3066	4500	14730	128000	FALSE	TRUE
3067	6300	14730	99500	FALSE	FALSE
3068	3500	14730	220000	FALSE	FALSE
3069	4667	14730	174000	FALSE	FALSE
3070	4800	14730	190000	FALSE	TRUE
3071	6300	14730	125000	FALSE	FALSE
3072	4050	14730	160000	FALSE	TRUE
3073	2400	14730	172000	FALSE	TRUE
3074	4200	14730	80000	FALSE	FALSE

3075	4080	14730	260000	FALSE	TRUE
3076	4440	14730	232500	FALSE	FALSE
3077	4960	14730	192500	FALSE	FALSE
3078	3630	14730	175000	FALSE	FALSE
3079	4720	14730	123000	FALSE	FALSE
3080	4025	14730	148000	FALSE	FALSE
3081	2450	14730	133400	FALSE	FALSE
3082	7200	14730	185000	FALSE	FALSE
3083	6000	14730	172500	FALSE	FALSE
3084	6000	14730	172500	FALSE	FALSE
3085	4446	14730	212000	FALSE	FALSE
3086	3600	14730	185000	FALSE	FALSE
3087	4332	14730	150000	FALSE	FALSE
3088	3720	14730	92500	FALSE	FALSE
3089	3120	14730	129900	FALSE	FALSE
3090	6338	14730	179000	FALSE	FALSE
3091	3600	14730	119000	FALSE	FALSE
3092	4875	14730	74000	FALSE	FALSE
3093	5994	14730	75000	FALSE	FALSE
3094	6936	14761	102000	FALSE	FALSE
3095	5265	14761	82000	FALSE	FALSE
3096	8450	14761	65000	FALSE	FALSE
3097	4603	14761	15100	FALSE	FALSE
3098	4603	14761	27900	FALSE	FALSE
3099	5000	14761	53000	FALSE	FALSE
3100	3600	14761	30000	FALSE	FALSE
3101	4720	14761	80000	FALSE	FALSE
3102	4800	14761	82000	FALSE	FALSE
3103	4752	14761	10000	FALSE	FALSE
3104	5160	14761	13000	FALSE	FALSE
3105	4950	14761	115000	FALSE	FALSE
3106	4920	14761	97000	FALSE	FALSE
3107	4800	14761	59800	FALSE	FALSE
3108	3600	14761	55000	FALSE	FALSE
3109	5160	14761	88000	FALSE	FALSE
3110	5460	14761	82500	FALSE	FALSE
3111	6000	14761	90000	FALSE	FALSE
3112	5400	14761	100000	FALSE	FALSE
3113	5500	14761	90000	FALSE	FALSE
3114	5400	14761	142000	FALSE	FALSE
3115	5160	14761	82500	FALSE	FALSE
3116	6125	14761	49900	FALSE	FALSE
3117	6750	14761	102200	FALSE	FALSE

3118	5500	14761	109000	FALSE	FALSE
3119	5500	14761	75500	FALSE	FALSE
3120	10470	14761	154000	FALSE	TRUE
3121	6100	14761	87500	FALSE	FALSE
3122	6000	14761	93000	FALSE	FALSE
3123	6000	14761	72000	FALSE	FALSE
3124	5460	14761	68000	FALSE	FALSE
3125	6000	14761	102000	FALSE	FALSE
3126	5610	14761	66000	FALSE	FALSE
3127	6500	14761	285000	TRUE	TRUE
3128	5570	14761	354000	TRUE	TRUE
3129	3360	14761	205000	TRUE	TRUE
3130	5715	14761	381000	TRUE	TRUE
3131	5080	14761	237500	TRUE	TRUE
3132	3600	14761	195000	TRUE	TRUE
3133	4720	14761	230000	TRUE	TRUE
3134	4200	14761	249400	TRUE	TRUE
3135	4080	14761	220000	TRUE	TRUE
3136	4572	14761	170000	TRUE	TRUE
3137	1590	14761	219000	TRUE	TRUE
3138	3840	14761	140000	TRUE	TRUE
3139	3600	14761	50500	FALSE	FALSE
3140	9333	14761	135000	FALSE	FALSE
3141	4797	14761	159500	FALSE	FALSE
3142	6360	14761	200000	FALSE	FALSE
3143	4800	14761	145500	FALSE	FALSE
3144	6250	14761	224000	FALSE	TRUE
3145	5376	14761	154900	FALSE	FALSE
3146	4879	14761	124000	FALSE	FALSE
3147	5040	14761	126500	FALSE	FALSE
3148	6000	14761	188500	FALSE	TRUE
3149	5000	14761	149900	FALSE	FALSE
3150	6480	14761	102800	FALSE	FALSE
3151	11351	14761	368000	FALSE	FALSE
3152	15679	14761	243900	FALSE	FALSE
3153	5400	14761	240000	FALSE	TRUE
3154	4800	14761	119900	FALSE	FALSE
3155	11760	14761	230500	FALSE	FALSE
3156	4800	14761	125000	FALSE	FALSE
3157	5080	14761	132500	FALSE	FALSE
3158	6240	14761	119900	FALSE	FALSE
3159	5240	14761	115000	FALSE	FALSE
3160	7200	14761	183500	FALSE	TRUE

3161	7900	14761	111000	FALSE	FALSE
3162	6380	14761	91000	FALSE	FALSE
3163	5192	14761	123500	FALSE	FALSE
3164	8400	14761	103500	FALSE	FALSE
3165	8400	14761	130000	FALSE	FALSE
3166	8835	14761	106000	FALSE	FALSE
3167	5628	14761	118000	FALSE	FALSE
3168	6000	14761	138500	FALSE	FALSE
3169	6000	14761	114300	FALSE	FALSE
3170	5080	14761	132000	FALSE	FALSE
3171	18095	14761	230000	FALSE	FALSE
3172	6290	14761	38000	FALSE	FALSE
3174	9165	14761	234000	FALSE	TRUE
3175	4760	14761	137400	FALSE	FALSE
3176	6960	14761	95000	FALSE	FALSE
3177	6160	14761	99400	FALSE	FALSE
3178	4920	14761	93000	FALSE	FALSE
3179	5180	14761	81000	FALSE	FALSE
3180	4840	14761	65000	FALSE	FALSE
3181	5000	14761	98000	FALSE	FALSE
3182	4324	14761	55000	FALSE	FALSE
3183	4800	14761	83000	FALSE	FALSE
3184	7200	14761	69000	FALSE	FALSE
3185	7843	14761	99500	FALSE	FALSE
3186	6471	14761	59000	FALSE	FALSE
3187	4879	14761	107000	FALSE	FALSE
3188	7200	14761	110000	FALSE	FALSE
3189	3600	14761	40000	FALSE	FALSE
3190	3360	14761	55500	FALSE	FALSE
3191	5000	14761	117500	FALSE	FALSE
3192	4880	14761	94000	FALSE	FALSE
3193	4410	14761	128000	FALSE	FALSE
3194	2850	14761	45000	FALSE	FALSE
3195	13524	14761	112500	FALSE	FALSE
3196	10284	14761	279900	FALSE	FALSE
3197	9387	14761	218155	FALSE	FALSE
3198	7200	14761	128000	FALSE	FALSE
3199	11692	14761	136900	FALSE	FALSE
3200	8400	14761	72500	FALSE	FALSE
3201	7200	14761	119000	FALSE	FALSE
3202	4800	14761	143600	FALSE	FALSE
3203	4960	14761	126500	FALSE	FALSE
3204	4025	14761	120000	FALSE	FALSE

3205	6084	14761	220000	FALSE	FALSE
3206	5320	14761	147000	FALSE	FALSE
3207	5336	14761	75100	FALSE	FALSE
3208	3870	14761	50000	FALSE	FALSE
3209	7500	14761	155900	FALSE	FALSE
3210	8000	14761	275000	FALSE	FALSE
3211	4602	14761	211000	FALSE	FALSE
3212	4760	14761	170000	FALSE	FALSE
3213	4455	14761	113000	FALSE	FALSE
3214	9800	14761	150000	FALSE	FALSE
3215	5936	14761	176500	FALSE	TRUE
3216	4838	14761	224500	FALSE	FALSE
3217	10496	14761	260000	FALSE	TRUE
3218	5400	14761	187500	FALSE	FALSE
3219	3600	14761	158500	FALSE	FALSE
3220	7200	14761	149000	FALSE	FALSE
3221	5439	14761	148800	FALSE	FALSE
3222	9782	14761	86000	FALSE	FALSE
3223	7920	14761	170000	FALSE	FALSE
3224	7093	14761	210000	FALSE	FALSE
3225	6000	14761	148500	FALSE	FALSE
3226	7920	14761	144500	FALSE	TRUE
3227	5120	14761	144000	FALSE	FALSE
3228	6820	14761	188900	FALSE	FALSE
3229	7729	14761	123000	FALSE	FALSE
3230	6120	14761	182000	FALSE	FALSE
3231	4920	14761	163000	FALSE	FALSE
3232	4600	14761	144000	FALSE	FALSE
3233	8580	14761	155000	FALSE	FALSE
3234	6800	14761	153000	FALSE	FALSE
3235	7320	14761	135000	FALSE	FALSE
3236	8352	14761	145000	FALSE	FALSE
3237	6615	14761	137500	FALSE	FALSE
3238	5500	14761	123500	FALSE	FALSE
3239	5712	14761	127500	FALSE	FALSE
3240	8400	14761	119500	FALSE	FALSE
3241	5400	14761	90000	FALSE	FALSE
3242	5040	14761	222000	FALSE	FALSE
3243	5120	14761	71500	FALSE	FALSE
3244	4125	14761	73300	FALSE	FALSE
3245	4200	14761	120500	FALSE	FALSE
3246	3000	14761	69000	FALSE	FALSE
3247	4080	14761	105000	FALSE	FALSE

3248	19096	14761	148000	FALSE	FALSE
3249	5676	14761	119900	FALSE	FALSE
3250	5080	14761	154000	FALSE	FALSE
3251	4880	14761	117000	FALSE	FALSE
3252	11440	14761	87000	FALSE	FALSE
3253	9520	14761	235000	FALSE	TRUE
3254	6636	14761	158000	FALSE	TRUE
3255	7290	14761	180000	FALSE	FALSE
3256	9375	14761	180000	FALSE	FALSE
3257	8760	14761	188000	FALSE	FALSE
3258	7320	14761	155000	FALSE	FALSE
3259	7200	14761	136400	FALSE	FALSE
3260	5200	14761	117400	FALSE	FALSE
3261	5400	14761	187000	FALSE	FALSE
3262	6076	14761	170000	FALSE	FALSE
3263	7680	14761	142000	FALSE	FALSE
3264	6750	14761	395000	FALSE	TRUE
3265	10595	14761	155000	FALSE	FALSE
3266	4080	14761	134000	FALSE	FALSE
3267	3600	14761	114500	FALSE	FALSE
3268	3720	14761	194500	FALSE	TRUE
3269	4880	14761	110000	FALSE	FALSE
3270	2343	14761	125500	FALSE	FALSE
3271	6000	14761	205000	FALSE	FALSE
3272	5040	14761	64000	FALSE	FALSE
3273	4370	14761	142900	FALSE	FALSE
3274	3600	14761	233000	FALSE	TRUE
3275	5490	14761	122000	FALSE	FALSE
3276	3600	14761	153000	FALSE	FALSE
3277	3750	14761	122800	FALSE	FALSE
3278	2520	14761	135000	FALSE	FALSE
3279	6696	14761	128000	FALSE	FALSE
3280	5160	14761	115000	FALSE	FALSE
3281	4960	14761	118000	FALSE	FALSE
3282	4216	14761	130000	FALSE	FALSE
3283	5600	14761	249900	FALSE	FALSE
3284	6104	14761	375000	FALSE	TRUE
3285	8580	14761	168000	FALSE	FALSE
3286	5490	14761	124900	FALSE	FALSE
3287	7620	14761	90000	FALSE	FALSE
3288	3600	14761	1000	FALSE	FALSE
3289	5040	14791	103800	FALSE	FALSE
3290	10080	14791	77500	FALSE	FALSE

3291	6528	14791	132000	FALSE	FALSE
3292	6840	14791	82000	FALSE	FALSE
3293	5640	14791	34900	FALSE	FALSE
3294	4920	14791	80000	FALSE	FALSE
3295	5664	14791	95000	FALSE	FALSE
3296	5330	14791	119900	FALSE	FALSE
3297	6000	14791	119400	FALSE	FALSE
3298	8040	14791	89500	FALSE	FALSE
3299	4800	14791	79900	FALSE	FALSE
3300	6050	14791	110000	FALSE	FALSE
3301	6120	14791	70000	FALSE	FALSE
3302	5250	14791	66000	FALSE	FALSE
3303	4800	14791	33900	FALSE	FALSE
3304	3600	14791	134900	TRUE	TRUE
3305	3600	14791	135000	TRUE	TRUE
3306	3600	14791	73900	TRUE	FALSE
3307	3600	14791	139000	TRUE	TRUE
3308	4800	14791	205000	TRUE	FALSE
3309	3180	14791	198900	TRUE	TRUE
3310	5250	14791	79000	FALSE	FALSE
3311	5504	14791	153600	FALSE	FALSE
3312	4920	14791	146000	FALSE	FALSE
3313	4800	14791	105900	FALSE	FALSE
3314	10950	14791	169000	FALSE	FALSE
3315	4840	14791	167000	FALSE	FALSE
3316	5400	14791	164000	FALSE	FALSE
3317	5080	14791	130000	FALSE	FALSE
3318	10950	14791	140000	FALSE	FALSE
3319	5080	14791	118000	FALSE	FALSE
3320	6250	14791	93500	FALSE	FALSE
3321	9295	14791	147900	FALSE	FALSE
3322	4440	14791	50000	FALSE	FALSE
3323	6160	14791	62000	FALSE	FALSE
3324	6560	14791	102000	FALSE	FALSE
3325	5000	14791	106900	FALSE	FALSE
3326	4920	14791	115000	FALSE	FALSE
3327	4920	14791	74000	FALSE	FALSE
3328	6160	14791	85000	FALSE	FALSE
3329	3600	14791	28500	FALSE	FALSE
3330	4720	14791	112000	FALSE	FALSE
3331	3600	14791	67500	FALSE	FALSE
3332	3600	14791	65000	FALSE	FALSE
3333	3750	14791	34000	FALSE	FALSE

3334	5751	14791	100000	FALSE	FALSE
3335	3600	14791	65000	FALSE	FALSE
3336	5715	14791	82500	FALSE	FALSE
3337	5778	14791	120000	FALSE	FALSE
3338	3959	14791	85873	FALSE	FALSE
3339	3840	14791	111850	FALSE	FALSE
3340	3600	14791	75000	FALSE	FALSE
3341	3750	14791	70000	FALSE	FALSE
3342	8156	14791	98000	FALSE	FALSE
3343	12100	14791	127500	FALSE	FALSE
3344	7200	14791	95500	FALSE	FALSE
3345	19835	14791	101400	FALSE	FALSE
3346	5265	14791	270000	FALSE	FALSE
3347	7392	14791	39000	FALSE	FALSE
3348	8800	14791	180000	FALSE	FALSE
3349	4800	14791	190000	FALSE	FALSE
3350	5080	14791	163000	FALSE	FALSE
3351	4760	14791	80000	FALSE	FALSE
3352	2850	14791	100000	FALSE	FALSE
3353	4522	14791	150000	FALSE	FALSE
3354	5400	14791	178000	FALSE	FALSE
3355	5330	14791	79900	FALSE	FALSE
3356	5580	14791	58000	FALSE	FALSE
3357	3600	14791	43500	FALSE	FALSE
3358	3600	14791	97500	FALSE	FALSE
3359	5400	14791	149900	FALSE	FALSE
3360	4445	14791	70000	FALSE	FALSE
3361	5280	14791	123000	FALSE	FALSE
3362	5160	14791	115000	FALSE	FALSE
3363	5805	14791	220000	FALSE	FALSE
3364	6885	14791	101900	FALSE	FALSE
3365	5625	14791	133500	FALSE	FALSE
3366	4800	14791	138000	FALSE	FALSE
3367	4800	14791	119000	FALSE	FALSE
3368	7260	14791	168000	FALSE	FALSE
3369	7344	14791	100000	FALSE	FALSE
3370	6240	14791	133000	FALSE	FALSE
3371	8100	14791	120000	FALSE	FALSE
3372	9450	14791	230000	FALSE	FALSE
3373	4840	14791	129500	FALSE	FALSE
3374	7200	14791	125500	FALSE	FALSE
3375	5160	14791	130000	FALSE	FALSE
3376	5376	14791	140000	FALSE	FALSE

3377	6165	14791	132900	FALSE	FALSE
3378	8970	14791	262500	FALSE	FALSE
3379	4920	14791	125000	FALSE	TRUE
3380	3600	14791	119900	FALSE	FALSE
3381	6480	14791	140000	FALSE	FALSE
3382	5633	14791	128000	FALSE	FALSE
3383	14791	14791	170000	FALSE	FALSE
3384	4800	14791	205000	FALSE	FALSE
3385	3600	14791	75000	FALSE	FALSE
3386	6348	14791	165000	FALSE	FALSE
3387	5074	14791	135000	FALSE	FALSE
3388	3600	14791	130000	FALSE	FALSE
3389	4998	14791	115000	FALSE	FALSE
3390	1050	14791	59900	FALSE	FALSE
3391	17400	14791	429900	FALSE	TRUE
3392	4880	14791	140000	FALSE	FALSE
3393	3904	14791	219000	FALSE	TRUE
3394	4680	14791	191000	FALSE	FALSE
3395	4485	14791	99000	FALSE	FALSE
3396	4200	14791	118000	FALSE	FALSE
3397	5760	14791	114900	FALSE	FALSE
3398	4750	14791	120000	FALSE	FALSE
3399	2856	14791	215300	FALSE	FALSE
3400	4900	14791	133000	FALSE	FALSE
3401	3120	14791	162000	FALSE	FALSE
3402	2920	14791	135900	FALSE	FALSE
3403	3600	14791	40550	FALSE	FALSE
3404	4585	14791	115000	FALSE	FALSE
3405	4655	14791	58000	FALSE	FALSE
3406	3600	14791	33000	FALSE	FALSE
3407	3960	14791	22500	FALSE	FALSE
3408	5715	14822	79500	FALSE	FALSE
3409	5040	14822	35000	FALSE	FALSE
3410	7995	14822	62000	FALSE	FALSE
3411	5040	14822	70000	FALSE	FALSE
3412	6540	14822	145000	FALSE	FALSE
3413	4960	14822	55000	FALSE	FALSE
3414	4960	14822	40000	FALSE	FALSE
3415	5940	14822	63000	FALSE	FALSE
3416	7866	14822	165000	FALSE	FALSE
3417	5082	14822	94900	FALSE	FALSE
3418	5203	14822	84900	FALSE	FALSE
3419	5200	14822	75000	FALSE	FALSE

3420	8384	14822	94000	FALSE	FALSE
3421	4920	14822	75000	FALSE	FALSE
3422	3600	14822	155000	TRUE	TRUE
3423	3600	14822	200000	TRUE	TRUE
3424	10710	14822	690000	TRUE	TRUE
3425	4880	14822	308000	TRUE	TRUE
3426	4800	14822	318000	TRUE	TRUE
3427	4880	14822	206000	TRUE	TRUE
3428	4104	14822	799000	TRUE	TRUE
3429	4800	14822	156900	FALSE	FALSE
3430	6000	14822	167000	FALSE	FALSE
3431	6800	14822	145000	FALSE	FALSE
3432	6000	14822	149000	FALSE	FALSE
3433	5120	14822	128000	FALSE	FALSE
3434	6250	14822	152000	FALSE	TRUE
3435	5000	14822	99000	FALSE	FALSE
3436	21104	14822	85000	FALSE	FALSE
3437	5400	14822	147000	FALSE	FALSE
3438	6022	14822	120000	FALSE	FALSE
3439	10140	14822	310000	FALSE	TRUE
3440	14933	14822	87000	FALSE	FALSE
3441	6050	14822	110700	FALSE	FALSE
3442	8262	14822	107000	FALSE	FALSE
3443	3145	14822	306000	FALSE	TRUE
3444	3937	14822	1100	FALSE	FALSE
3445	3600	14822	35000	FALSE	FALSE
3446	7426	14822	25000	FALSE	FALSE
3447	4800	14822	79900	FALSE	FALSE
3448	5031	14822	54900	FALSE	FALSE
3449	4920	14822	65000	FALSE	FALSE
3450	3600	14822	41000	FALSE	FALSE
3451	3800	14822	33500	FALSE	FALSE
3452	4320	14822	48000	FALSE	FALSE
3453	4200	14822	62000	FALSE	FALSE
3454	3810	14822	88000	FALSE	FALSE
3455	6975	14822	140000	FALSE	FALSE
3456	2550	14822	122500	FALSE	FALSE
3457	3600	14822	75000	FALSE	FALSE
3458	3844	14822	85000	FALSE	FALSE
3459	4445	14822	90000	FALSE	FALSE
3460	3740	14822	90000	FALSE	FALSE
3461	7800	14822	89900	FALSE	FALSE
3462	8105	14822	140000	FALSE	FALSE

3463	9169	14822	126000	FALSE	FALSE
3464	12375	14822	107500	FALSE	FALSE
3465	5047	14822	236000	FALSE	FALSE
3466	5160	14822	139000	FALSE	FALSE
3467	7150	14822	199000	FALSE	FALSE
3468	7392	14822	70000	FALSE	FALSE
3469	4096	14822	105000	FALSE	FALSE
3470	4800	14822	80000	FALSE	FALSE
3471	6600	14822	236900	FALSE	TRUE
3472	4520	14822	139000	FALSE	FALSE
3473	4800	14822	138000	FALSE	FALSE
3474	3750	14822	156000	FALSE	FALSE
3475	5760	14822	93000	FALSE	FALSE
3476	5200	14822	138000	FALSE	FALSE
3477	3600	14822	60000	FALSE	FALSE
3478	4800	14822	140000	FALSE	FALSE
3479	4920	14822	150000	FALSE	FALSE
3480	4800	14822	133000	FALSE	FALSE
3481	4760	14822	127600	FALSE	FALSE
3482	4797	14822	164500	FALSE	FALSE
3483	5750	14822	128000	FALSE	FALSE
3484	4760	14822	153000	FALSE	FALSE
3485	10010	14822	187000	FALSE	FALSE
3486	12893	14822	165000	FALSE	FALSE
3487	4800	14822	140000	FALSE	FALSE
3488	6480	14822	147000	FALSE	FALSE
3489	4800	14822	123000	FALSE	FALSE
3490	7625	14822	144000	FALSE	FALSE
3491	8040	14822	128900	FALSE	FALSE
3492	6000	14822	135000	FALSE	FALSE
3493	2465	14822	78000	FALSE	FALSE
3494	3060	14822	65000	FALSE	FALSE
3495	7560	14822	161500	FALSE	FALSE
3496	4200	14822	87400	FALSE	FALSE
3497	10800	14822	140000	FALSE	FALSE
3498	6596	14822	155000	FALSE	FALSE
3499	5900	14822	172000	FALSE	FALSE
3500	5400	14822	138000	FALSE	FALSE
3501	6708	14822	109900	FALSE	FALSE
3502	4680	14822	155000	FALSE	FALSE
3503	6875	14822	172000	FALSE	FALSE
3504	6348	14822	144500	FALSE	FALSE
3505	7280	14822	168500	FALSE	TRUE

3506	4800	14822	150000	FALSE	FALSE
3507	4880	14822	110000	FALSE	FALSE
3508	4880	14822	136000	FALSE	FALSE
3509	5002	14822	149950	FALSE	FALSE
3510	4400	14822	225000	FALSE	FALSE
3511	4838	14822	216600	FALSE	FALSE
3512	4680	14822	131900	FALSE	FALSE
3513	4636	14822	120000	FALSE	FALSE
3514	3870	14822	175000	FALSE	FALSE
3515	5850	14822	156000	FALSE	FALSE
3516	3240	14822	140000	FALSE	FALSE
3517	5500	14822	110000	FALSE	FALSE
3518	4000	14822	164500	FALSE	FALSE
3519	4200	14822	50000	FALSE	FALSE
3520	6000	14822	45000	FALSE	FALSE
3521	5950	14822	157000	FALSE	FALSE
3522	4800	14822	110000	FALSE	FALSE
3523	3500	14822	15000	FALSE	FALSE
3524	6960	14853	85000	FALSE	FALSE
3525	5400	14853	97900	FALSE	FALSE
3526	4800	14853	70000	FALSE	FALSE
3527	4800	14853	10000	FALSE	FALSE
3528	9420	14853	75000	FALSE	FALSE
3529	4800	14853	79900	FALSE	FALSE
3530	10349	14853	116500	FALSE	FALSE
3531	4800	14853	55000	FALSE	FALSE
3532	11088	14853	69000	FALSE	FALSE
3533	6750	14853	35000	FALSE	FALSE
3534	3600	14853	115000	TRUE	TRUE
3535	4484	14853	198000	TRUE	TRUE
3536	4560	14853	161300	TRUE	TRUE
3537	4560	14853	215000	TRUE	TRUE
3538	4826	14853	195000	TRUE	FALSE
3539	3600	14853	124300	TRUE	FALSE
3540	6000	14853	179900	FALSE	TRUE
3541	5376	14853	115300	FALSE	FALSE
3542	5120	14853	119000	FALSE	FALSE
3543	9585	14853	201500	FALSE	FALSE
3544	9000	14853	160577	FALSE	FALSE
3545	5040	14853	103000	FALSE	FALSE
3546	6384	14853	129900	FALSE	FALSE
3547	8246	14853	155000	FALSE	FALSE
3548	6000	14853	110000	FALSE	FALSE

3549	11850	14853	183800	FALSE	FALSE
3550	7800	14853	96200	FALSE	FALSE
3551	6890	14853	95000	FALSE	FALSE
3552	4800	14853	110000	FALSE	FALSE
3553	9984	14853	144000	FALSE	FALSE
3554	6000	14853	118000	FALSE	FALSE
3555	6500	14853	95000	FALSE	FALSE
3556	4920	14853	125000	FALSE	FALSE
3557	7095	14853	85000	FALSE	FALSE
3558	4800	14853	79000	FALSE	FALSE
3559	10582	14853	220000	FALSE	FALSE
3560	4000	14853	134900	FALSE	FALSE
3561	4440	14853	25039	FALSE	FALSE
3562	4290	14853	10000	FALSE	FALSE
3563	4960	14853	92000	FALSE	FALSE
3564	4840	14853	138400	FALSE	FALSE
3565	3600	14853	35000	FALSE	FALSE
3566	3600	14853	35000	FALSE	FALSE
3567	3182	14853	110600	FALSE	FALSE
3568	4235	14853	80000	FALSE	FALSE
3569	18532	14853	131500	FALSE	FALSE
3570	20651	14853	120000	FALSE	FALSE
3571	12826	14853	250000	FALSE	FALSE
3572	7200	14853	160000	FALSE	FALSE
3573	7800	14853	130000	FALSE	FALSE
3574	7200	14853	105000	FALSE	FALSE
3575	10000	14853	113000	FALSE	FALSE
3576	7272	14853	96000	FALSE	FALSE
3577	9738	14853	72000	FALSE	FALSE
3578	7200	14853	80000	FALSE	FALSE
3579	6587	14853	177000	FALSE	FALSE
3580	5200	14853	104000	FALSE	FALSE
3581	3720	14853	130000	FALSE	FALSE
3582	5760	14853	127500	FALSE	FALSE
3583	8600	14853	250000	FALSE	TRUE
3584	3600	14853	85000	FALSE	FALSE
3585	5250	14853	65000	FALSE	FALSE
3586	4879	14853	140000	FALSE	FALSE
3587	8400	14853	96000	FALSE	FALSE
3588	9250	14853	165000	FALSE	FALSE
3589	4797	14853	121700	FALSE	FALSE
3590	5400	14853	160000	FALSE	TRUE
3591	4920	14853	120000	FALSE	FALSE

3592	13260	14853	124900	FALSE	FALSE
3593	7980	14853	115000	FALSE	FALSE
3594	7260	14853	155000	FALSE	FALSE
3595	8250	14853	175900	FALSE	FALSE
3596	4900	14853	145000	FALSE	TRUE
3597	6906	14853	182500	FALSE	FALSE
3598	10720	14853	170000	FALSE	FALSE
3599	7980	14853	136000	FALSE	FALSE
3600	4826	14853	145000	FALSE	FALSE
3601	7326	14853	90000	FALSE	FALSE
3602	7245	14853	127500	FALSE	FALSE
3604	7260	14853	135000	FALSE	FALSE
3605	9500	14853	205000	FALSE	FALSE
3606	7202	14853	220000	FALSE	FALSE
3607	3500	14853	51000	FALSE	FALSE
3608	4290	14853	82000	FALSE	FALSE
3609	3872	14853	94000	FALSE	FALSE
3610	5456	14853	84000	FALSE	FALSE
3611	4880	14853	134900	FALSE	FALSE
3612	21000	14853	275000	FALSE	FALSE
3613	5480	14853	136100	FALSE	FALSE
3614	4200	14853	116000	FALSE	FALSE
3615	6028	14853	125000	FALSE	FALSE
3616	8225	14853	126092	FALSE	FALSE
3617	4800	14853	97700	FALSE	FALSE
3618	2640	14853	120000	FALSE	FALSE
3619	5160	14853	105000	FALSE	FALSE
3620	5040	14853	127000	FALSE	FALSE
3621	6150	14853	125000	FALSE	FALSE
3622	2521	14853	187900	FALSE	FALSE
3623	3600	14853	75000	FALSE	FALSE
3624	3025	14853	113000	FALSE	FALSE
3625	3300	14853	168000	FALSE	TRUE
3626	3540	14853	96200	FALSE	FALSE
3627	4389	14853	99000	FALSE	FALSE
3628	6500	14853	27917	FALSE	FALSE
3629	9450	14853	20000	FALSE	FALSE
3630	4680	14853	13500	FALSE	FALSE
3631	3600	14853	12000	FALSE	FALSE
3632	6762	14853	173000	FALSE	FALSE
3633	5000	14853	89900	FALSE	FALSE
3634	1590	14853	19000	FALSE	FALSE
3635	4960	14883	72000	FALSE	FALSE

3636	5400	14883	89000	FALSE	FALSE
3637	13111	14883	138000	FALSE	FALSE
3638	30000	14883	119900	FALSE	FALSE
3639	7905	14883	85000	FALSE	FALSE
3640	6300	14883	65000	FALSE	FALSE
3641	4750	14883	42500	FALSE	FALSE
3642	6150	14883	96500	FALSE	FALSE
3643	6000	14883	107500	FALSE	FALSE
3644	9240	14883	43500	FALSE	FALSE
3645	5040	14883	75000	FALSE	FALSE
3646	8910	14883	74500	FALSE	FALSE
3647	7245	14883	70000	FALSE	FALSE
3648	15840	14883	324274	TRUE	FALSE
3649	9000	14883	535000	TRUE	TRUE
3650	4720	14883	200000	TRUE	FALSE
3651	3600	14883	66000	TRUE	FALSE
3652	5985	14883	134500	FALSE	FALSE
3653	6250	14883	135000	FALSE	FALSE
3654	5000	14883	147000	FALSE	FALSE
3655	6360	14883	140000	FALSE	FALSE
3656	5060	14883	148000	FALSE	FALSE
3657	5000	14883	70000	FALSE	FALSE
3658	6240	14883	130000	FALSE	FALSE
3659	6480	14883	136450	FALSE	FALSE
3660	6120	14883	130000	FALSE	FALSE
3661	5280	14883	99900	FALSE	FALSE
3662	6240	14883	123000	FALSE	FALSE
3663	5771	14883	82000	FALSE	FALSE
3664	5610	14883	89000	FALSE	FALSE
3665	3850	14883	51000	FALSE	FALSE
3666	1860	14883	47500	FALSE	FALSE
3667	1490	14883	22500	FALSE	FALSE
3668	3600	14883	6000	FALSE	FALSE
3669	3960	14883	24900	FALSE	FALSE
3670	3600	14883	65000	FALSE	FALSE
3671	3600	14883	115000	FALSE	FALSE
3672	5000	14883	99900	FALSE	FALSE
3673	7950	14883	53000	FALSE	FALSE
3674	21780	14883	124800	FALSE	FALSE
3675	9863	14883	222900	FALSE	FALSE
3676	10673	14883	145000	FALSE	FALSE
3677	10320	14883	121000	FALSE	FALSE
3678	7320	14883	109900	FALSE	FALSE

3679	8856	14883	96000	FALSE	FALSE
3680	7620	14883	79900	FALSE	FALSE
3681	8514	14883	75000	FALSE	FALSE
3682	5200	14883	64000	FALSE	FALSE
3683	6600	14883	249900	FALSE	FALSE
3684	4760	14883	181500	FALSE	FALSE
3685	5160	14883	110000	FALSE	FALSE
3686	4800	14883	93000	FALSE	FALSE
3687	6720	14883	65000	FALSE	FALSE
3688	7056	14883	160000	FALSE	FALSE
3689	4800	14883	73800	FALSE	FALSE
3690	3960	14883	193000	FALSE	FALSE
3691	3780	14883	77400	FALSE	FALSE
3692	5390	14883	241000	FALSE	FALSE
3693	6700	14883	130000	FALSE	FALSE
3694	4920	14883	149900	FALSE	FALSE
3695	6075	14883	118000	FALSE	FALSE
3696	6700	14883	147200	FALSE	FALSE
3697	4300	14883	133300	FALSE	FALSE
3698	8700	14883	190000	FALSE	FALSE
3699	6660	14883	130000	FALSE	FALSE
3700	7455	14883	148000	FALSE	FALSE
3701	8400	14883	120000	FALSE	FALSE
3702	7625	14883	141000	FALSE	FALSE
3703	8442	14883	118000	FALSE	FALSE
3704	12511	14883	147000	FALSE	FALSE
3705	7560	14883	137900	FALSE	FALSE
3706	7320	14883	134500	FALSE	FALSE
3707	6240	14883	121500	FALSE	FALSE
3708	5450	14883	124300	FALSE	FALSE
3709	7503	14883	120000	FALSE	FALSE
3710	11376	14883	137000	FALSE	FALSE
3711	6750	14883	77500	FALSE	FALSE
3712	5499	14883	80000	FALSE	FALSE
3713	8496	14883	127000	FALSE	FALSE
3714	5985	14883	182000	FALSE	FALSE
3715	11326	14883	245000	FALSE	FALSE
3716	3570	14883	124300	FALSE	FALSE
3717	9675	14883	156000	FALSE	FALSE
3718	6240	14883	130000	FALSE	FALSE
3719	4920	14883	140600	FALSE	FALSE
3720	4920	14883	132500	FALSE	FALSE
3721	2880	14883	31000	FALSE	FALSE

3722	5490	14883	132500	FALSE	FALSE
3723	6390	14883	174900	FALSE	FALSE
3724	3600	14883	129900	FALSE	FALSE
3725	3870	14883	189000	FALSE	TRUE
3726	7946	14883	257000	FALSE	TRUE
3727	3600	14883	95000	FALSE	FALSE
3728	3630	14883	187500	FALSE	FALSE
3729	7400	14883	379000	FALSE	TRUE
3730	4500	14883	135750	FALSE	FALSE
3731	5490	14883	123000	FALSE	FALSE
3732	3870	14883	229000	FALSE	FALSE
3733	3660	14883	66500	FALSE	FALSE
3734	5681	14883	100	FALSE	FALSE
3735	4080	14883	44000	FALSE	FALSE
3736	3600	14883	9500	FALSE	FALSE
3737	3840	14883	47000	FALSE	FALSE
3738	5187	14883	119000	FALSE	FALSE
3739	5999	14883	85000	FALSE	FALSE
3740	3720	14883	14000	FALSE	FALSE
3741	5400	14914	60000	FALSE	FALSE
3742	6160	14914	14000	FALSE	FALSE
3743	5400	14914	96000	FALSE	FALSE
3744	5535	14914	75000	FALSE	FALSE
3745	5040	14914	110000	FALSE	FALSE
3746	6000	14914	62500	FALSE	FALSE
3747	5400	14914	84000	FALSE	FALSE
3748	4920	14914	37500	FALSE	FALSE
3749	8760	14914	150800	FALSE	FALSE
3750	7300	14914	82000	FALSE	FALSE
3751	9750	14914	525000	TRUE	FALSE
3752	9900	14914	500000	TRUE	TRUE
3753	6000	14914	360000	TRUE	TRUE
3754	6000	14914	275000	TRUE	TRUE
3755	5120	14914	159000	FALSE	FALSE
3756	4800	14914	160000	FALSE	FALSE
3757	5106	14914	162300	FALSE	TRUE
3758	5000	14914	147000	FALSE	FALSE
3759	8400	14914	302000	FALSE	FALSE
3760	5000	14914	172000	FALSE	FALSE
3761	4800	14914	170000	FALSE	FALSE
3762	9280	14914	185000	FALSE	FALSE
3763	7800	14914	140000	FALSE	FALSE
3764	4800	14914	157000	FALSE	FALSE

3765	5080	14914	123000	FALSE	FALSE
3766	3960	14914	57000	FALSE	FALSE
3767	5100	14914	137500	FALSE	FALSE
3768	4800	14914	82500	FALSE	FALSE
3769	11040	14914	176000	FALSE	FALSE
3770	4920	14914	100000	FALSE	FALSE
3771	4800	14914	110000	FALSE	FALSE
3772	3780	14914	84000	FALSE	FALSE
3773	4640	14914	65000	FALSE	FALSE
3774	4800	14914	113000	FALSE	FALSE
3775	3840	14914	110000	FALSE	FALSE
3776	3547	14914	133600	FALSE	FALSE
3777	3600	14914	65000	FALSE	FALSE
3778	11450	14914	345000	FALSE	FALSE
3779	8400	14914	106000	FALSE	FALSE
3780	11500	14914	132500	FALSE	FALSE
3781	7620	14914	80000	FALSE	FALSE
3782	7200	14914	85000	FALSE	FALSE
3783	7200	14914	76000	FALSE	FALSE
3784	21780	14914	189000	FALSE	FALSE
3785	7440	14914	150000	FALSE	FALSE
3786	7100	14914	112500	FALSE	FALSE
3787	4046	14914	63300	FALSE	FALSE
3788	5060	14914	190000	FALSE	FALSE
3789	3720	14914	68000	FALSE	FALSE
3790	4489	14914	109500	FALSE	FALSE
3791	3810	14914	61400	FALSE	FALSE
3792	4800	14914	143000	FALSE	FALSE
3793	4520	14914	171500	FALSE	FALSE
3794	3570	14914	110000	FALSE	FALSE
3795	4560	14914	160000	FALSE	FALSE
3796	4500	14914	77750	FALSE	FALSE
3797	9585	14914	153200	FALSE	FALSE
3798	7800	14914	185000	FALSE	FALSE
3799	6431	14914	133000	FALSE	FALSE
3800	4050	14914	87000	FALSE	FALSE
3801	4200	14914	175400	FALSE	FALSE
3802	11796	14914	145000	FALSE	FALSE
3803	10880	14914	163000	FALSE	FALSE
3804	4880	14914	106500	FALSE	FALSE
3805	5060	14914	107000	FALSE	FALSE
3806	5520	14914	130000	FALSE	FALSE
3807	5002	14914	130000	FALSE	FALSE

3808	8400	14914	120000	FALSE	FALSE
3809	9750	14914	171000	FALSE	FALSE
3810	6450	14914	155000	FALSE	FALSE
3811	7260	14914	160000	FALSE	FALSE
3812	4800	14914	98500	FALSE	FALSE
3813	3600	14914	99800	FALSE	FALSE
3814	4200	14914	87000	FALSE	FALSE
3815	5560	14914	151000	FALSE	FALSE
3816	4800	14914	310000	FALSE	TRUE
3817	6496	14914	200000	FALSE	FALSE
3818	6300	14914	140000	FALSE	FALSE
3819	4920	14914	255000	FALSE	FALSE
3820	5880	14914	110000	FALSE	FALSE
3821	4290	14914	162500	FALSE	FALSE
3822	5040	14914	196000	FALSE	FALSE
3823	5400	14914	65000	FALSE	FALSE
3824	4092	14914	31576	FALSE	FALSE
3825	6000	14914	66700	FALSE	FALSE
3826	5950	14914	148000	FALSE	FALSE
3827	5400	14944	48400	FALSE	FALSE
3828	4800	14944	45000	FALSE	FALSE
3829	4800	14944	57000	FALSE	FALSE
3830	5040	14944	30000	FALSE	FALSE
3831	6160	14944	62000	FALSE	FALSE
3832	6240	14944	45000	FALSE	FALSE
3833	4680	14944	19000	FALSE	FALSE
3834	4680	14944	25000	FALSE	FALSE
3835	21780	14944	134000	FALSE	FALSE
3836	5080	14944	86500	FALSE	FALSE
3837	21331	14944	159900	FALSE	FALSE
3838	5400	14944	95000	FALSE	FALSE
3839	4800	14944	96900	FALSE	FALSE
3840	5120	14944	85000	FALSE	FALSE
3841	4920	14944	82000	FALSE	FALSE
3842	7644	14944	90000	FALSE	FALSE
3843	5160	14944	81000	FALSE	FALSE
3844	6360	14944	39900	FALSE	FALSE
3845	13200	14944	430000	TRUE	TRUE
3846	5490	14944	395000	TRUE	TRUE
3847	3600	14944	263000	TRUE	TRUE
3848	3600	14944	90000	TRUE	FALSE
3849	6250	14944	159000	FALSE	FALSE
3850	6750	14944	139900	FALSE	FALSE

3851	4800	14944	129000	FALSE	FALSE
3852	5125	14944	115500	FALSE	FALSE
3853	6400	14944	123000	FALSE	FALSE
3854	4800	14944	112500	FALSE	FALSE
3855	12009	14944	350000	FALSE	FALSE
3856	5500	14944	129000	FALSE	FALSE
3857	6500	14944	160000	FALSE	FALSE
3858	7200	14944	92000	FALSE	FALSE
3859	5885	14944	80000	FALSE	FALSE
3860	7686	14944	95000	FALSE	FALSE
3861	6000	14944	30000	FALSE	FALSE
3862	4840	14944	124900	FALSE	FALSE
3863	4800	14944	55000	FALSE	FALSE
3864	3375	14944	9900	FALSE	FALSE
3865	4800	14944	53000	FALSE	FALSE
3866	4270	14944	26000	FALSE	FALSE
3867	4800	14944	128000	FALSE	FALSE
3868	3720	14944	60000	FALSE	FALSE
3869	3795	14944	62000	FALSE	FALSE
3870	3540	14944	90000	FALSE	FALSE
3872	10700	14944	143900	FALSE	FALSE
3873	7200	14944	56000	FALSE	FALSE
3874	12600	14944	125000	FALSE	FALSE
3875	10000	14944	119000	FALSE	FALSE
3876	7717	14944	108000	FALSE	FALSE
3877	7200	14944	106000	FALSE	FALSE
3878	7700	14944	65000	FALSE	FALSE
3879	4680	14944	129000	FALSE	FALSE
3880	7392	14944	186000	FALSE	FALSE
3881	4046	14944	59900	FALSE	FALSE
3882	4680	14944	135000	FALSE	FALSE
3883	6674	14944	127500	FALSE	FALSE
3884	3960	14944	75000	FALSE	FALSE
3885	3750	14944	115500	FALSE	FALSE
3886	5625	14944	130000	FALSE	FALSE
3887	11200	14944	120000	FALSE	FALSE
3888	6000	14944	135000	FALSE	FALSE
3889	5500	14944	155000	FALSE	FALSE
3890	6500	14944	123800	FALSE	FALSE
3891	4800	14944	133000	FALSE	FALSE
3892	5000	14944	169000	FALSE	FALSE
3893	9480	14944	177000	FALSE	FALSE
3894	7320	14944	130000	FALSE	FALSE

3895	7320	14944	142000	FALSE	FALSE
3896	5040	14944	118000	FALSE	FALSE
3897	5500	14944	120000	FALSE	FALSE
3898	7920	14944	137000	FALSE	FALSE
3899	7080	14944	93000	FALSE	FALSE
3900	22320	14944	130000	FALSE	FALSE
3901	3720	14944	63000	FALSE	FALSE
3902	3480	14944	75000	FALSE	FALSE
3903	4760	14944	154000	FALSE	FALSE
3904	5715	14944	159000	FALSE	FALSE
3905	4830	14944	95000	FALSE	FALSE
3906	6250	14944	166000	FALSE	FALSE
3907	3600	14944	90500	FALSE	FALSE
3908	8296	14944	178000	FALSE	FALSE
3909	10680	14944	156000	FALSE	FALSE
3910	4830	14944	112700	FALSE	FALSE
3911	5520	14944	167000	FALSE	FALSE
3912	4375	14944	135000	FALSE	FALSE
3913	4620	14944	110000	FALSE	FALSE
3914	4270	14944	86000	FALSE	FALSE
3915	4880	14944	124900	FALSE	FALSE
3916	3600	14944	136000	FALSE	FALSE
3917	4620	14944	66990	FALSE	FALSE
3918	2880	14944	42500	FALSE	FALSE
3919	11459	14944	130000	FALSE	FALSE
3920	6000	14944	129400	FALSE	FALSE
3921	3003	14944	185000	FALSE	FALSE
3922	8280	14944	219000	FALSE	FALSE
3923	4674	14944	105000	FALSE	FALSE
3924	4880	14944	130000	FALSE	FALSE
3925	2700	14944	44000	FALSE	FALSE
3926	4800	14975	75000	FALSE	FALSE
3927	7200	14975	103500	FALSE	FALSE
3928	5120	14975	55000	FALSE	FALSE
3929	6165	14975	48000	FALSE	FALSE
3930	4960	14975	33000	FALSE	FALSE
3931	6144	14975	75000	FALSE	FALSE
3932	7200	14975	83000	FALSE	FALSE
3933	3960	14975	87700	TRUE	FALSE
3934	3310	14975	405000	TRUE	FALSE
3935	6000	14975	170900	FALSE	FALSE
3936	6400	14975	162000	FALSE	FALSE
3937	5632	14975	155000	FALSE	FALSE

3938	5760	14975	150000	FALSE	TRUE
3939	5504	14975	118900	FALSE	FALSE
3940	4800	14975	124000	FALSE	FALSE
3941	5000	14975	113500	FALSE	FALSE
3942	5355	14975	119750	FALSE	FALSE
3943	4800	14975	115000	FALSE	FALSE
3944	4960	14975	97500	FALSE	FALSE
3945	5000	14975	135000	FALSE	FALSE
3946	5520	14975	135000	FALSE	FALSE
3947	14528	14975	157500	FALSE	FALSE
3948	7524	14975	115000	FALSE	FALSE
3949	5000	14975	110000	FALSE	FALSE
3950	6837	14975	90000	FALSE	FALSE
3951	8040	14975	50000	FALSE	FALSE
3952	3690	14975	20000	FALSE	FALSE
3953	2244	14975	35000	FALSE	FALSE
3954	4800	14975	85000	FALSE	FALSE
3955	4880	14975	72000	FALSE	FALSE
3956	4900	14975	80000	FALSE	FALSE
3957	5850	14975	108000	FALSE	FALSE
3958	3600	14975	79000	FALSE	FALSE
3959	3600	14975	75000	FALSE	FALSE
3960	3959	14975	87000	FALSE	FALSE
3961	8543	14975	79100	FALSE	FALSE
3962	4320	14975	152050	FALSE	FALSE
3963	4760	14975	130000	FALSE	FALSE
3964	4680	14975	96500	FALSE	FALSE
3965	4800	14975	170000	FALSE	FALSE
3966	4522	14975	148300	FALSE	FALSE
3967	4200	14975	179900	FALSE	FALSE
3968	4640	14975	100000	FALSE	FALSE
3969	4920	14975	174000	FALSE	FALSE
3970	5200	14975	140000	FALSE	FALSE
3971	5412	14975	152000	FALSE	FALSE
3972	5400	14975	110000	FALSE	FALSE
3973	6700	14975	168000	FALSE	FALSE
3974	5120	14975	138000	FALSE	FALSE
3975	7750	14975	83000	FALSE	FALSE
3976	7772	14975	120000	FALSE	FALSE
3977	6400	14975	132000	FALSE	FALSE
3978	7480	14975	155000	FALSE	FALSE
3979	8235	14975	156000	FALSE	FALSE
3980	8520	14975	119500	FALSE	FALSE

3981	3750	14975	35000	FALSE	FALSE
3982	9900	14975	155000	FALSE	FALSE
3983	7516	14975	150000	FALSE	FALSE
3984	5400	14975	128515	FALSE	FALSE
3985	3150	14975	159000	FALSE	FALSE
3986	5490	14975	109900	FALSE	FALSE
3987	3600	14975	90000	FALSE	FALSE
3988	3660	14975	89000	FALSE	FALSE
3989	4500	14975	155000	FALSE	TRUE
3990	3450	14975	180000	FALSE	FALSE
3991	4280	14975	67000	FALSE	FALSE
3992	7106	14975	100000	FALSE	FALSE
3993	5400	15006	62500	FALSE	FALSE
3994	5400	15006	23000	FALSE	FALSE
3995	7260	15006	131900	FALSE	FALSE
3996	4800	15006	47750	FALSE	FALSE
3997	5040	15006	113500	FALSE	FALSE
3998	4800	15006	80000	FALSE	FALSE
3999	4720	15006	159000	TRUE	FALSE
4000	4840	15006	300420	TRUE	TRUE
4001	6120	15006	152500	FALSE	FALSE
4002	4800	15006	121500	FALSE	FALSE
4003	4879	15006	143000	FALSE	FALSE
4004	5000	15006	118000	FALSE	FALSE
4005	9280	15006	135000	FALSE	FALSE
4006	5512	15006	132000	FALSE	FALSE
4007	4914	15006	90000	FALSE	FALSE
4008	15982	15006	98500	FALSE	FALSE
4009	4800	15006	128000	FALSE	FALSE
4010	7164	15006	130000	FALSE	FALSE
4011	7245	15006	130000	FALSE	FALSE
4012	4800	15006	65500	FALSE	FALSE
4013	5000	15006	69900	FALSE	FALSE
4014	5355	15006	100000	FALSE	FALSE
4015	3600	15006	105000	FALSE	FALSE
4016	7276	15006	79000	FALSE	FALSE
4017	10018	15006	219000	FALSE	FALSE
4018	3840	15006	94900	FALSE	FALSE
4019	4800	15006	130000	FALSE	FALSE
4020	3810	15006	90000	FALSE	FALSE
4021	12864	15006	140000	FALSE	FALSE
4022	7840	15006	115000	FALSE	FALSE
4023	8576	15006	118200	FALSE	FALSE

4024	7552	15006	141500	FALSE	FALSE
4025	8640	15006	172000	FALSE	FALSE
4026	5280	15006	112500	FALSE	FALSE
4027	7070	15006	120000	FALSE	FALSE
4028	6042	15006	142000	FALSE	FALSE
4029	6650	15006	121000	FALSE	FALSE
4030	1800	15006	58500	FALSE	FALSE
4031	5590	15006	143000	FALSE	FALSE
4032	7200	15006	197000	FALSE	FALSE
4033	7200	15006	131900	FALSE	FALSE
4034	6450	15006	130000	FALSE	FALSE
4035	5625	15006	126000	FALSE	FALSE
4036	7950	15006	325000	FALSE	FALSE
4037	3605	15006	82400	FALSE	FALSE
4038	4464	15006	111500	FALSE	FALSE
4039	4515	15006	279000	FALSE	TRUE
4040	4636	15006	102500	FALSE	FALSE
4041	3600	15006	51000	FALSE	FALSE
4042	4674	15006	72000	FALSE	FALSE
4043	6000	15006	105000	FALSE	FALSE
4044	5160	15034	44500	FALSE	FALSE
4045	9396	15034	130800	FALSE	FALSE
4046	13104	15034	129900	FALSE	FALSE
4047	6450	15034	99000	FALSE	FALSE
4048	5715	15034	31725	FALSE	FALSE
4049	5040	15034	125000	FALSE	FALSE
4050	5000	15034	40000	FALSE	FALSE
4051	5000	15034	30000	FALSE	FALSE
4052	5000	15034	40000	FALSE	FALSE
4053	5000	15034	19500	FALSE	FALSE
4054	11475	15034	99000	FALSE	FALSE
4055	6976	15034	40000	FALSE	FALSE
4056	7164	15034	79900	FALSE	FALSE
4057	5800	15034	668000	TRUE	TRUE
4058	3600	15034	105500	TRUE	FALSE
4059	2500	15034	142000	TRUE	TRUE
4060	3360	15034	205000	TRUE	TRUE
4061	3600	15034	130000	TRUE	TRUE
4062	5376	15034	155000	FALSE	FALSE
4063	5625	15034	97000	FALSE	FALSE
4064	5000	15034	122000	FALSE	FALSE
4065	4800	15034	88000	FALSE	FALSE
4066	5400	15034	55000	FALSE	FALSE

4067	5760	15034	178900	FALSE	FALSE
4068	4840	15034	142900	FALSE	FALSE
4069	5120	15034	137000	FALSE	FALSE
4070	6372	15034	110000	FALSE	FALSE
4071	4800	15034	101000	FALSE	FALSE
4072	6272	15034	130000	FALSE	FALSE
4073	7930	15034	129900	FALSE	TRUE
4074	6300	15034	215000	FALSE	TRUE
4075	2100	15034	70000	FALSE	FALSE
4076	3800	15034	1250	FALSE	FALSE
4077	3600	15034	8000	FALSE	FALSE
4078	7280	15034	65000	FALSE	FALSE
4079	5124	15034	30000	FALSE	FALSE
4080	4600	15034	33500	FALSE	FALSE
4081	3600	15034	32000	FALSE	FALSE
4082	5950	15034	100000	FALSE	FALSE
4083	4250	15034	100000	FALSE	FALSE
4084	5080	15034	86000	FALSE	FALSE
4085	3600	15034	85000	FALSE	FALSE
4086	10692	15034	198900	FALSE	FALSE
4087	10530	15034	168000	FALSE	FALSE
4088	7740	15034	81800	FALSE	FALSE
4089	18502	15034	259900	FALSE	FALSE
4090	7200	15034	106000	FALSE	FALSE
4091	4370	15034	130000	FALSE	FALSE
4092	2747	15034	197500	FALSE	FALSE
4093	4800	15034	139000	FALSE	FALSE
4094	4800	15034	143500	FALSE	FALSE
4095	3750	15034	100000	FALSE	FALSE
4096	4640	15034	66900	FALSE	FALSE
4097	4760	15034	98900	FALSE	FALSE
4098	3720	15034	85000	FALSE	FALSE
4099	4125	15034	139900	FALSE	FALSE
4100	5000	15034	109000	FALSE	FALSE
4101	5040	15034	125000	FALSE	FALSE
4102	6650	15034	147500	FALSE	FALSE
4103	7738	15034	94000	FALSE	FALSE
4104	8500	15034	145000	FALSE	FALSE
4105	8400	15034	135000	FALSE	FALSE
4106	5000	15034	133500	FALSE	FALSE
4107	6120	15034	107500	FALSE	FALSE
4108	7440	15034	115900	FALSE	FALSE
4109	7200	15034	107000	FALSE	FALSE

4110	5160	15034	115000	FALSE	FALSE
4111	7080	15034	240000	FALSE	TRUE
4112	3600	15034	62000	FALSE	FALSE
4113	3600	15034	50000	FALSE	FALSE
4115	4920	15034	105000	FALSE	FALSE
4116	17990	15034	265000	FALSE	FALSE
4117	4800	15034	148000	FALSE	FALSE
4118	7560	15034	138000	FALSE	FALSE
4119	8950	15034	115800	FALSE	FALSE
4120	7440	15034	145000	FALSE	FALSE
4121	7150	15034	199900	FALSE	FALSE
4122	6000	15034	160000	FALSE	FALSE
4123	4800	15034	150000	FALSE	FALSE
4124	4960	15034	190000	FALSE	FALSE
4125	3705	15034	164500	FALSE	FALSE
4126	6480	15034	206000	FALSE	FALSE
4127	2500	15034	99000	FALSE	FALSE
4128	3510	15034	164000	FALSE	FALSE
4129	3600	15034	234200	FALSE	TRUE
4130	5640	15034	131500	FALSE	FALSE
4131	3720	15034	158000	FALSE	FALSE
4132	3000	15034	153000	FALSE	FALSE
4133	3150	15034	143000	FALSE	FALSE
4134	3025	15034	179900	FALSE	TRUE
4135	3600	15034	122000	FALSE	FALSE
4136	5950	15034	143500	FALSE	FALSE
4137	3010	15034	33000	FALSE	FALSE
4138	4800	15065	55000	FALSE	FALSE
4139	8448	15065	96500	FALSE	FALSE
4140	4920	15065	84900	FALSE	FALSE
4141	7360	15065	116600	FALSE	FALSE
4142	5330	15065	113500	FALSE	FALSE
4143	5900	15065	109000	FALSE	FALSE
4144	5000	15065	80000	FALSE	FALSE
4145	6820	15065	74900	FALSE	FALSE
4146	11200	15065	485000	TRUE	FALSE
4147	8400	15065	512000	TRUE	TRUE
4148	2500	15065	215500	TRUE	TRUE
4149	3600	15065	160000	TRUE	FALSE
4150	7200	15065	149000	TRUE	TRUE
4151	5000	15065	136000	FALSE	FALSE
4152	6000	15065	118000	FALSE	FALSE
4153	4800	15065	166000	FALSE	FALSE

4154	6000	15065	141000	FALSE	TRUE
4155	5120	15065	98900	FALSE	FALSE
4156	4800	15065	124000	FALSE	FALSE
4157	6000	15065	109900	FALSE	FALSE
4158	4960	15065	146000	FALSE	FALSE
4159	6750	15065	135000	FALSE	FALSE
4160	7590	15065	115000	FALSE	FALSE
4161	4800	15065	100000	FALSE	FALSE
4162	5760	15065	75000	FALSE	FALSE
4163	7380	15065	72350	FALSE	FALSE
4164	5000	15065	106500	FALSE	FALSE
4165	7500	15065	32000	FALSE	FALSE
4166	3600	15065	40000	FALSE	FALSE
4167	4914	15065	54500	FALSE	FALSE
4168	4720	15065	67000	FALSE	FALSE
4169	5000	15065	101000	FALSE	FALSE
4170	5040	15065	105000	FALSE	FALSE
4171	3750	15065	55500	FALSE	FALSE
4172	4200	15065	85000	FALSE	FALSE
4173	3630	15065	64900	FALSE	FALSE
4174	7820	15065	21405	FALSE	FALSE
4175	8065	15065	105000	FALSE	FALSE
4176	7680	15065	182000	FALSE	FALSE
4177	5400	15065	152000	FALSE	FALSE
4178	5355	15065	43100	FALSE	FALSE
4179	5040	15065	67000	FALSE	FALSE
4180	4320	15065	157000	FALSE	TRUE
4181	3680	15065	116900	FALSE	TRUE
4182	3600	15065	95000	FALSE	FALSE
4183	5080	15065	70000	FALSE	FALSE
4184	7150	15065	82000	FALSE	FALSE
4185	4800	15065	84500	FALSE	FALSE
4186	5400	15065	78000	FALSE	FALSE
4187	4760	15065	161200	FALSE	FALSE
4188	6650	15065	155000	FALSE	FALSE
4189	8040	15065	65000	FALSE	FALSE
4190	9272	15065	240000	FALSE	TRUE
4191	8710	15065	179900	FALSE	TRUE
4192	8030	15065	152000	FALSE	TRUE
4193	7260	15065	181000	FALSE	FALSE
4194	7536	15065	180000	FALSE	FALSE
4195	6201	15065	133000	FALSE	FALSE
4196	6600	15065	148400	FALSE	FALSE

4197	12420	15065	140000	FALSE	FALSE
4198	6060	15065	125000	FALSE	FALSE
4199	4944	15065	135000	FALSE	FALSE
4200	4884	15065	112500	FALSE	FALSE
4201	6644	15065	135000	FALSE	FALSE
4202	7370	15065	112000	FALSE	FALSE
4203	5500	15065	121200	FALSE	FALSE
4204	6360	15065	137500	FALSE	FALSE
4205	7420	15065	117000	FALSE	FALSE
4206	3600	15065	74000	FALSE	FALSE
4207	5130	15065	121000	FALSE	FALSE
4208	8580	15065	147000	FALSE	FALSE
4209	5246	15065	115000	FALSE	FALSE
4210	5220	15065	155000	FALSE	FALSE
4211	5400	15065	207000	FALSE	FALSE
4212	7560	15065	160000	FALSE	FALSE
4213	11100	15065	190000	FALSE	FALSE
4214	8400	15065	192000	FALSE	FALSE
4215	7800	15065	170000	FALSE	FALSE
4216	6161	15065	103000	FALSE	FALSE
4217	4800	15065	113900	FALSE	FALSE
4218	6708	15065	105000	FALSE	FALSE
4219	6000	15065	184000	FALSE	TRUE
4220	8856	15065	125000	FALSE	FALSE
4221	7623	15065	132000	FALSE	FALSE
4222	4500	15065	122000	FALSE	FALSE
4223	4815	15065	141200	FALSE	FALSE
4224	6900	15065	128250	FALSE	FALSE
4225	3600	15065	210000	FALSE	TRUE
4226	4800	15065	153100	FALSE	FALSE
4227	3276	15065	176000	FALSE	FALSE
4228	4080	15065	174800	FALSE	TRUE
4229	6250	15065	118000	FALSE	FALSE
4230	4900	15065	225000	FALSE	FALSE
4231	1710	15065	136900	FALSE	FALSE
4232	3660	15065	92500	FALSE	FALSE
4233	3600	15065	194500	FALSE	FALSE
4234	2920	15065	111500	FALSE	FALSE
4235	4314	15065	109900	FALSE	FALSE
4236	5000	15065	67000	FALSE	FALSE
4237	5500	15095	80000	FALSE	FALSE
4238	6080	15095	135900	FALSE	TRUE
4239	5880	15095	90000	FALSE	FALSE

4240	6600	15095	99900	FALSE	FALSE
4241	4800	15095	76000	FALSE	FALSE
4242	5880	15095	77000	FALSE	FALSE
4243	4800	15095	32000	FALSE	FALSE
4244	8906	15095	86000	FALSE	FALSE
4245	6150	15095	75000	FALSE	FALSE
4246	7232	15095	100000	FALSE	FALSE
4247	5200	15095	80000	FALSE	FALSE
4248	5160	15095	92000	FALSE	FALSE
4249	9600	15095	68000	FALSE	FALSE
4250	4620	15095	75000	FALSE	FALSE
4251	5080	15095	77000	FALSE	FALSE
4252	4800	15095	118500	FALSE	FALSE
4253	6000	15095	98000	FALSE	FALSE
4254	6000	15095	75000	FALSE	FALSE
4255	3600	15095	173000	TRUE	FALSE
4256	3600	15095	201000	TRUE	TRUE
4257	8400	15095	72000	TRUE	FALSE
4258	4720	15095	165000	TRUE	FALSE
4259	3600	15095	210000	TRUE	TRUE
4260	2220	15095	145000	TRUE	TRUE
4261	4800	15095	230000	TRUE	FALSE
4262	9600	15095	455000	TRUE	FALSE
4263	2550	15095	200000	TRUE	FALSE
4264	4800	15095	227000	TRUE	TRUE
4266	12001	15095	142500	FALSE	FALSE
4267	4720	15095	152000	FALSE	FALSE
4268	5120	15095	147500	FALSE	FALSE
4269	9000	15095	124000	FALSE	FALSE
4270	4800	15095	114800	FALSE	FALSE
4271	5060	15095	107000	FALSE	FALSE
4272	8494	15095	143500	FALSE	FALSE
4273	8400	15095	169000	FALSE	TRUE
4274	6250	15095	160000	FALSE	FALSE
4275	6720	15095	127000	FALSE	FALSE
4276	5715	15095	136000	FALSE	FALSE
4277	6500	15095	130800	FALSE	FALSE
4278	21780	15095	166000	FALSE	TRUE
4279	5762	15095	124900	FALSE	FALSE
4280	7426	15095	105000	FALSE	FALSE
4281	2870	15095	12500	FALSE	FALSE
4282	3775	15095	19000	FALSE	FALSE
4283	8360	15095	100500	FALSE	FALSE

4284	4400	15095	55500	FALSE	FALSE
4285	3600	15095	24000	FALSE	FALSE
4286	3750	15095	11500	FALSE	FALSE
4287	6120	15095	125000	FALSE	FALSE
4288	3780	15095	82500	FALSE	FALSE
4289	3600	15095	81500	FALSE	FALSE
4290	3540	15095	65000	FALSE	FALSE
4291	9353	15095	44000	FALSE	FALSE
4292	4982	15095	194000	FALSE	FALSE
4293	6210	15095	156582	FALSE	FALSE
4294	7440	15095	155600	FALSE	FALSE
4295	4860	15095	131500	FALSE	FALSE
4296	6000	15095	105000	FALSE	FALSE
4297	7434	15095	59000	FALSE	FALSE
4298	9030	15095	380000	FALSE	FALSE
4299	5160	15095	149000	FALSE	FALSE
4300	7335	15095	106200	FALSE	FALSE
4301	4945	15095	160000	FALSE	FALSE
4302	5450	15095	135000	FALSE	FALSE
4303	5687	15095	119000	FALSE	FALSE
4304	6786	15095	109500	FALSE	FALSE
4305	5760	15095	65000	FALSE	FALSE
4306	5400	15095	235000	FALSE	FALSE
4307	4800	15095	140000	FALSE	FALSE
4308	5000	15095	92000	FALSE	FALSE
4309	9620	15095	182000	FALSE	FALSE
4310	6720	15095	150000	FALSE	FALSE
4311	4960	15095	136500	FALSE	FALSE
4312	6120	15095	185000	FALSE	FALSE
4313	4836	15095	168000	FALSE	FALSE
4314	5625	15095	119000	FALSE	FALSE
4315	7150	15095	188000	FALSE	FALSE
4316	7877	15095	136000	FALSE	FALSE
4317	6650	15095	140000	FALSE	FALSE
4318	6550	15095	154000	FALSE	FALSE
4319	8113	15095	114900	FALSE	FALSE
4320	7250	15095	122000	FALSE	FALSE
4321	5520	15095	125000	FALSE	FALSE
4322	8400	15095	155000	FALSE	FALSE
4323	6270	15095	94000	FALSE	FALSE
4324	5720	15095	86700	FALSE	FALSE
4325	7192	15095	122000	FALSE	FALSE
4326	7260	15095	92000	FALSE	FALSE

4327	6120	15095	95000	FALSE	FALSE
4328	7680	15095	101000	FALSE	FALSE
4329	7980	15095	122500	FALSE	FALSE
4330	3500	15095	48000	FALSE	FALSE
4331	4674	15095	119000	FALSE	FALSE
4332	7200	15095	150000	FALSE	FALSE
4333	8400	15095	146900	FALSE	FALSE
4334	7260	15095	123500	FALSE	FALSE
4335	7590	15095	137500	FALSE	FALSE
4336	4800	15095	75000	FALSE	FALSE
4337	5004	15095	148200	FALSE	FALSE
4338	6048	15095	122700	FALSE	FALSE
4339	3660	15095	85700	FALSE	FALSE
4340	6313	15095	173500	FALSE	FALSE
4341	4200	15095	110000	FALSE	FALSE
4342	5000	15095	135650	FALSE	TRUE
4343	3870	15095	200000	FALSE	FALSE
4344	3600	15095	179000	FALSE	FALSE
4345	9600	15095	180000	FALSE	FALSE
4346	4800	15095	142000	FALSE	FALSE
4347	3600	15095	33678	FALSE	FALSE
4348	7568	15126	85500	FALSE	FALSE
4349	3710	15126	25000	FALSE	FALSE
4350	5400	15126	43000	FALSE	FALSE
4351	6240	15126	96800	FALSE	FALSE
4352	7560	15126	69300	FALSE	FALSE
4353	5460	15126	82500	FALSE	FALSE
4354	6000	15126	70000	FALSE	FALSE
4355	4800	15126	96000	FALSE	FALSE
4356	8448	15126	98000	FALSE	FALSE
4357	6000	15126	69500	FALSE	FALSE
4358	5240	15126	80000	FALSE	FALSE
4359	6650	15126	277000	TRUE	TRUE
4360	3600	15126	169000	TRUE	TRUE
4361	3600	15126	145000	TRUE	FALSE
4362	8400	15126	645000	TRUE	TRUE
4363	9750	15126	613750	TRUE	FALSE
4364	9900	15126	505225	TRUE	TRUE
4365	4880	15126	424000	TRUE	TRUE
4366	3848	15126	265000	TRUE	TRUE
4367	3840	15126	237500	TRUE	TRUE
4368	10800	15126	140000	FALSE	FALSE
4369	8256	15126	148500	FALSE	FALSE

4370	4995	15126	154400	FALSE	FALSE
4371	5400	15126	161900	FALSE	FALSE
4372	6500	15126	153000	FALSE	FALSE
4373	5120	15126	150000	FALSE	FALSE
4374	6000	15126	89000	FALSE	FALSE
4375	5265	15126	115000	FALSE	FALSE
4376	5400	15126	96000	FALSE	FALSE
4377	10530	15126	240500	FALSE	FALSE
4378	6816	15126	135000	FALSE	FALSE
4379	5520	15126	77000	FALSE	FALSE
4380	8625	15126	62000	FALSE	FALSE
4381	11371	15126	163800	FALSE	FALSE
4382	10875	15126	138000	FALSE	FALSE
4383	4800	15126	80000	FALSE	FALSE
4384	7250	15126	101000	FALSE	FALSE
4385	4800	15126	103500	FALSE	FALSE
4386	5994	15126	123000	FALSE	FALSE
4387	9455	15126	148500	FALSE	FALSE
4388	9024	15126	110000	FALSE	FALSE
4389	9825	15126	92000	FALSE	FALSE
4390	6050	15126	137000	FALSE	FALSE
4391	8400	15126	109900	FALSE	FALSE
4392	7380	15126	75500	FALSE	FALSE
4393	6000	15126	131500	FALSE	FALSE
4394	4940	15126	39000	FALSE	FALSE
4395	3210	15126	40000	FALSE	FALSE
4396	3750	15126	6500	FALSE	FALSE
4397	3450	15126	151000	FALSE	FALSE
4398	3990	15126	25656	FALSE	FALSE
4399	4920	15126	155000	FALSE	FALSE
4400	7168	15126	46000	FALSE	FALSE
4401	4773	15126	101000	FALSE	FALSE
4402	7500	15126	32000	FALSE	FALSE
4403	5250	15126	126000	FALSE	FALSE
4404	4500	15126	79000	FALSE	FALSE
4405	5130	15126	97000	FALSE	FALSE
4406	3750	15126	73500	FALSE	FALSE
4407	10080	15126	155000	FALSE	FALSE
4408	7200	15126	150000	FALSE	FALSE
4409	13651	15126	144500	FALSE	FALSE
4410	15150	15126	130000	FALSE	FALSE
4411	7797	15126	109000	FALSE	FALSE
4412	7200	15126	90000	FALSE	FALSE

4413	4950	15126	86000	FALSE	FALSE
4414	7208	15126	101500	FALSE	FALSE
4415	7200	15126	93000	FALSE	FALSE
4416	12500	15126	123500	FALSE	FALSE
4417	10260	15126	105600	FALSE	FALSE
4418	6318	15126	250000	FALSE	FALSE
4419	4800	15126	138500	FALSE	FALSE
4420	5330	15126	114000	FALSE	FALSE
4421	5040	15126	100000	FALSE	FALSE
4422	2704	15126	89990	FALSE	FALSE
4423	4760	15126	58375	FALSE	FALSE
4424	4800	15126	82500	FALSE	FALSE
4425	3330	15126	50000	FALSE	FALSE
4426	4510	15126	146500	FALSE	FALSE
4427	4760	15126	89000	FALSE	FALSE
4428	10200	15126	182000	FALSE	FALSE
4429	5200	15126	122400	FALSE	FALSE
4430	4375	15126	92000	FALSE	FALSE
4431	3300	15126	17700	FALSE	FALSE
4432	6683	15126	242500	FALSE	FALSE
4433	5043	15126	140000	FALSE	FALSE
4434	4880	15126	124900	FALSE	FALSE
4435	5000	15126	126500	FALSE	FALSE
4436	8976	15126	70000	FALSE	FALSE
4437	5400	15126	97500	FALSE	FALSE
4438	5480	15126	112000	FALSE	FALSE
4439	10240	15126	235350	FALSE	FALSE
4440	8400	15126	167000	FALSE	FALSE
4441	5000	15126	147000	FALSE	FALSE
4442	7450	15126	149000	FALSE	FALSE
4443	6144	15126	195000	FALSE	FALSE
4444	8250	15126	144000	FALSE	FALSE
4445	7980	15126	98000	FALSE	FALSE
4446	7623	15126	144900	FALSE	FALSE
4447	5500	15126	147000	FALSE	FALSE
4448	5280	15126	120000	FALSE	FALSE
4449	8472	15126	148000	FALSE	FALSE
4450	8100	15126	90000	FALSE	FALSE
4451	6000	15126	115000	FALSE	FALSE
4452	9625	15126	127500	FALSE	FALSE
4453	7440	15126	142000	FALSE	FALSE
4454	8160	15126	108000	FALSE	FALSE
4455	6120	15126	110900	FALSE	FALSE

4456	7200	15126	143400	FALSE	FALSE
4457	6150	15126	113000	FALSE	FALSE
4458	5684	15126	150000	FALSE	TRUE
4459	9520	15126	96000	FALSE	FALSE
4460	9324	15126	140000	FALSE	FALSE
4461	5805	15126	146000	FALSE	FALSE
4462	4640	15126	144400	FALSE	FALSE
4463	3600	15126	95000	FALSE	FALSE
4464	15717	15126	180000	FALSE	FALSE
4465	9576	15126	150000	FALSE	FALSE
4466	7540	15126	189500	FALSE	FALSE
4467	5900	15126	137500	FALSE	FALSE
4468	7290	15126	125900	FALSE	FALSE
4469	9840	15126	137500	FALSE	FALSE
4470	5760	15126	124000	FALSE	FALSE
4471	7200	15126	115000	FALSE	FALSE
4472	5400	15126	95000	FALSE	FALSE
4473	4800	15126	104500	FALSE	FALSE
4474	4800	15126	104500	FALSE	FALSE
4475	4800	15126	110000	FALSE	FALSE
4476	4800	15126	60000	FALSE	FALSE
4477	4200	15126	120000	FALSE	FALSE
4478	2625	15126	225000	FALSE	FALSE
4479	1300	15126	68500	FALSE	FALSE
4480	3600	15126	105000	FALSE	FALSE
4481	4560	15126	140000	FALSE	FALSE
4482	3600	15126	145000	FALSE	FALSE
4483	4920	15126	194000	FALSE	TRUE
4484	5760	15126	130000	FALSE	FALSE
4485	5400	15126	210000	FALSE	FALSE
4486	3200	15126	101000	FALSE	FALSE
4487	2880	15126	43000	FALSE	FALSE
4488	4920	15126	232900	FALSE	FALSE
4489	3750	15126	28000	FALSE	FALSE
4490	10000	15156	113197	FALSE	FALSE
4491	6400	15156	62900	FALSE	FALSE
4492	6250	15156	69500	FALSE	FALSE
4493	4800	15156	55000	FALSE	FALSE
4494	8624	15156	89000	FALSE	FALSE
4495	11592	15156	88000	FALSE	FALSE
4496	5760	15156	40000	FALSE	FALSE
4497	6000	15156	70000	FALSE	FALSE
4498	4400	15156	99900	FALSE	FALSE

4499	5000	15156	43490	FALSE	FALSE
4500	5400	15156	37000	FALSE	FALSE
4501	4800	15156	74000	FALSE	FALSE
4502	8109	15156	75000	FALSE	FALSE
4503	11613	15156	112000	FALSE	FALSE
4504	4800	15156	85000	FALSE	FALSE
4505	6600	15156	362000	TRUE	FALSE
4506	5400	15156	292000	TRUE	TRUE
4507	4200	15156	194000	TRUE	FALSE
4508	10710	15156	650000	TRUE	TRUE
4509	8250	15156	555000	TRUE	TRUE
4510	8400	15156	368500	TRUE	FALSE
4511	4880	15156	395000	TRUE	TRUE
4512	2100	15156	125000	TRUE	FALSE
4513	7320	15156	147000	TRUE	FALSE
4514	14196	15156	211000	FALSE	FALSE
4515	5000	15156	112000	FALSE	FALSE
4516	5376	15156	115000	FALSE	FALSE
4517	7500	15156	157500	FALSE	TRUE
4518	5000	15156	115000	FALSE	FALSE
4519	4800	15156	107500	FALSE	FALSE
4520	6500	15156	167000	FALSE	FALSE
4521	8100	15156	125000	FALSE	FALSE
4522	7670	15156	126000	FALSE	FALSE
4523	5580	15156	100500	FALSE	FALSE
4524	7800	15156	80000	FALSE	FALSE
4525	10526	15156	137500	FALSE	FALSE
4526	7502	15156	83500	FALSE	FALSE
4527	5080	15156	68000	FALSE	FALSE
4528	6120	15156	87550	FALSE	FALSE
4529	6000	15156	102800	FALSE	FALSE
4530	6500	15156	82900	FALSE	FALSE
4531	6250	15156	109900	FALSE	FALSE
4532	12000	15156	35500	FALSE	FALSE
4533	3150	15156	31500	FALSE	FALSE
4534	3810	15156	142500	FALSE	TRUE
4535	5980	15156	64800	FALSE	FALSE
4536	4800	15156	79000	FALSE	FALSE
4537	4588	15156	28000	FALSE	FALSE
4538	3750	15156	15000	FALSE	FALSE
4539	4200	15156	39900	FALSE	FALSE
4540	3810	15156	89000	FALSE	FALSE
4541	4840	15156	100000	FALSE	FALSE

4542	4900	15156	83000	FALSE	FALSE
4543	7500	15156	93000	FALSE	FALSE
4544	3600	15156	76000	FALSE	FALSE
4545	2400	15156	81500	FALSE	FALSE
4546	3750	15156	75000	FALSE	FALSE
4547	11932	15156	83000	FALSE	FALSE
4548	10125	15156	129500	FALSE	FALSE
4549	20000	15156	73000	FALSE	FALSE
4550	6375	15156	113500	FALSE	FALSE
4551	6075	15156	140000	FALSE	FALSE
4552	4800	15156	95000	FALSE	FALSE
4553	6654	15156	107000	FALSE	FALSE
4554	5166	15156	137000	FALSE	FALSE
4555	5330	15156	86000	FALSE	FALSE
4556	5040	15156	92100	FALSE	FALSE
4557	5040	15156	123000	FALSE	FALSE
4558	6000	15156	185000	FALSE	FALSE
4559	4600	15156	200000	FALSE	FALSE
4560	3840	15156	71400	FALSE	FALSE
4561	4520	15156	163500	FALSE	FALSE
4562	4800	15156	105000	FALSE	FALSE
4563	4602	15156	191000	FALSE	FALSE
4564	4800	15156	126000	FALSE	FALSE
4565	4520	15156	31000	FALSE	FALSE
4566	4520	15156	21500	FALSE	FALSE
4567	6206	15156	208100	FALSE	FALSE
4568	6480	15156	100000	FALSE	FALSE
4569	7440	15156	92500	FALSE	FALSE
4570	4800	15156	77500	FALSE	FALSE
4571	6600	15156	230000	FALSE	FALSE
4572	5625	15156	171000	FALSE	FALSE
4573	13158	15156	168500	FALSE	FALSE
4574	4920	15156	148000	FALSE	FALSE
4575	5000	15156	128000	FALSE	FALSE
4576	10680	15156	165000	FALSE	FALSE
4577	5080	15156	137900	FALSE	FALSE
4578	5040	15156	148000	FALSE	FALSE
4579	8400	15156	165000	FALSE	FALSE
4580	6500	15156	163000	FALSE	FALSE
4581	7020	15156	151700	FALSE	FALSE
4582	7029	15156	141500	FALSE	FALSE
4583	5750	15156	120000	FALSE	FALSE
4584	4900	15156	125000	FALSE	FALSE

4585	10360	15156	123000	FALSE	FALSE
4586	7200	15156	143900	FALSE	FALSE
4587	5555	15156	94000	FALSE	FALSE
4588	6000	15156	84000	FALSE	FALSE
4589	7200	15156	85000	FALSE	FALSE
4590	1750	15156	50000	FALSE	FALSE
4591	3000	15156	26500	FALSE	FALSE
4592	3600	15156	85000	FALSE	FALSE
4593	4725	15156	154000	FALSE	FALSE
4594	5805	15156	110000	FALSE	FALSE
4595	4900	15156	120000	FALSE	FALSE
4596	7290	15156	129900	FALSE	FALSE
4597	5382	15156	158000	FALSE	FALSE
4598	8479	15156	131800	FALSE	FALSE
4599	7290	15156	125000	FALSE	FALSE
4600	4800	15156	128000	FALSE	FALSE
4601	4800	15156	222000	FALSE	TRUE
4602	5040	15156	148900	FALSE	FALSE
4603	5040	15156	155000	FALSE	FALSE
4604	4800	15156	190000	FALSE	FALSE
4605	4025	15156	168500	FALSE	FALSE
4606	3080	15156	141000	FALSE	FALSE
4607	7400	15156	96000	FALSE	FALSE
4608	8040	15156	229900	FALSE	FALSE
4609	7000	15156	146000	FALSE	FALSE
4610	3600	15156	202000	FALSE	FALSE
4611	4422	15156	153900	FALSE	FALSE
4612	5490	15156	72000	FALSE	FALSE
4613	6000	15156	158000	FALSE	FALSE
4614	5023	15156	120000	FALSE	FALSE
4615	6480	15187	67500	FALSE	FALSE
4616	4960	15187	15000	FALSE	FALSE
4617	8100	15187	85000	FALSE	FALSE
4618	4560	15187	54900	FALSE	FALSE
4619	6000	15187	99900	FALSE	FALSE
4620	5460	15187	118500	FALSE	FALSE
4621	4840	15187	59000	FALSE	FALSE
4622	6240	15187	114900	FALSE	FALSE
4623	5900	15187	79900	FALSE	FALSE
4624	5390	15187	52000	FALSE	FALSE
4625	6344	15187	382000	TRUE	FALSE
4626	5185	15187	156000	TRUE	FALSE
4627	3600	15187	110000	TRUE	FALSE

4628	4800	15187	172500	TRUE	TRUE
4629	3850	15187	415000	TRUE	FALSE
4630	6600	15187	425000	TRUE	FALSE
4631	11760	15187	599000	TRUE	TRUE
4632	10529	15187	475000	TRUE	TRUE
4633	4840	15187	295000	TRUE	TRUE
4634	4800	15187	239300	TRUE	TRUE
4635	3109	15187	187000	TRUE	FALSE
4637	3200	15187	220000	TRUE	TRUE
4638	5800	15187	608000	TRUE	TRUE
4639	5100	15187	18000	FALSE	FALSE
4640	6625	15187	154800	FALSE	FALSE
4641	8640	15187	127900	FALSE	FALSE
4642	6000	15187	175000	FALSE	FALSE
4643	6292	15187	137000	FALSE	FALSE
4644	4800	15187	151400	FALSE	FALSE
4645	6750	15187	120000	FALSE	FALSE
4646	4950	15187	117500	FALSE	FALSE
4647	5080	15187	148500	FALSE	FALSE
4648	6000	15187	85000	FALSE	FALSE
4649	9956	15187	187900	FALSE	FALSE
4650	6490	15187	120000	FALSE	FALSE
4651	21780	15187	210000	FALSE	FALSE
4652	10950	15187	160000	FALSE	FALSE
4653	21780	15187	166500	FALSE	FALSE
4654	7008	15187	118000	FALSE	FALSE
4655	7700	15187	116000	FALSE	FALSE
4656	8150	15187	116000	FALSE	FALSE
4657	5125	15187	99000	FALSE	FALSE
4658	5395	15187	16500	FALSE	FALSE
4659	3600	15187	45000	FALSE	FALSE
4660	3600	15187	14000	FALSE	FALSE
4661	3480	15187	43000	FALSE	FALSE
4662	2100	15187	20000	FALSE	FALSE
4663	4888	15187	175000	FALSE	FALSE
4664	4800	15187	95000	FALSE	FALSE
4665	5160	15187	70000	FALSE	FALSE
4666	4760	15187	125000	FALSE	FALSE
4667	4352	15187	33000	FALSE	FALSE
4668	6625	15187	78000	FALSE	FALSE
4669	1680	15187	42000	FALSE	FALSE
4670	4800	15187	113000	FALSE	FALSE
4671	5400	15187	90000	FALSE	FALSE

4672	5000	15187	76063	FALSE	FALSE
4673	3960	15187	79500	FALSE	FALSE
4674	5780	15187	90000	FALSE	FALSE
4675	9000	15187	134900	FALSE	FALSE
4676	6000	15187	62000	FALSE	FALSE
4677	7500	15187	64000	FALSE	FALSE
4678	15211	15187	140000	FALSE	FALSE
4679	6240	15187	110000	FALSE	FALSE
4680	7630	15187	113000	FALSE	FALSE
4681	7200	15187	117900	FALSE	FALSE
4682	7200	15187	81000	FALSE	FALSE
4683	4680	15187	275000	FALSE	FALSE
4684	6372	15187	194500	FALSE	FALSE
4685	5244	15187	209000	FALSE	FALSE
4686	5724	15187	162000	FALSE	FALSE
4687	6800	15187	181900	FALSE	FALSE
4688	4760	15187	215000	FALSE	TRUE
4689	5150	15187	129000	FALSE	FALSE
4690	5080	15187	143500	FALSE	FALSE
4691	4352	15187	130000	FALSE	FALSE
4692	4480	15187	79900	FALSE	FALSE
4693	4879	15187	96500	FALSE	FALSE
4694	5332	15187	150000	FALSE	FALSE
4695	4800	15187	119000	FALSE	FALSE
4696	5280	15187	86500	FALSE	FALSE
4697	5883	15187	183000	FALSE	FALSE
4698	13630	15187	125000	FALSE	FALSE
4699	6150	15187	158200	FALSE	FALSE
4700	4920	15187	135000	FALSE	FALSE
4701	4920	15187	130000	FALSE	FALSE
4702	8100	15187	132000	FALSE	FALSE
4703	6490	15187	129900	FALSE	FALSE
4704	6480	15187	94750	FALSE	FALSE
4705	4800	15187	182000	FALSE	FALSE
4706	6250	15187	189000	FALSE	FALSE
4707	7980	15187	125600	FALSE	FALSE
4708	7336	15187	174000	FALSE	FALSE
4709	7540	15187	130000	FALSE	FALSE
4710	6968	15187	135000	FALSE	FALSE
4711	7475	15187	144000	FALSE	FALSE
4712	7800	15187	107000	FALSE	FALSE
4713	6150	15187	138500	FALSE	FALSE
4714	6120	15187	98000	FALSE	FALSE

4715	7936	15187	132000	FALSE	FALSE
4716	6890	15187	87000	FALSE	FALSE
4717	6240	15187	138800	FALSE	FALSE
4718	4840	15187	105000	FALSE	FALSE
4719	5000	15187	158200	FALSE	TRUE
4720	5760	15187	63500	FALSE	FALSE
4721	5720	15187	150000	FALSE	FALSE
4722	7200	15187	143500	FALSE	FALSE
4723	4560	15187	137500	FALSE	FALSE
4724	6579	15187	75000	FALSE	FALSE
4725	9000	15187	215000	FALSE	FALSE
4726	7540	15187	165000	FALSE	FALSE
4727	9000	15187	148900	FALSE	FALSE
4728	7200	15187	180000	FALSE	FALSE
4729	6943	15187	130000	FALSE	FALSE
4730	14356	15187	140000	FALSE	FALSE
4731	4800	15187	74000	FALSE	FALSE
4732	4830	15187	115000	FALSE	FALSE
4733	6075	15187	139000	FALSE	FALSE
4734	5160	15187	162500	FALSE	FALSE
4735	3600	15187	147900	FALSE	TRUE
4736	5580	15187	170000	FALSE	FALSE
4737	6100	15187	130000	FALSE	FALSE
4738	3870	15187	145000	FALSE	TRUE
4739	4200	15187	110000	FALSE	FALSE
4740	3600	15187	235000	FALSE	FALSE
4741	3600	15187	207500	FALSE	FALSE
4742	3660	15187	46000	FALSE	FALSE
4743	8370	15187	144500	FALSE	FALSE
4744	5200	15187	93000	FALSE	FALSE
4745	4900	15187	99000	FALSE	FALSE
4746	6300	15187	111000	FALSE	FALSE
4747	2300	15187	116000	FALSE	FALSE
4748	5400	15187	125900	FALSE	FALSE
4749	4350	15187	126000	FALSE	FALSE
4750	4620	15187	69900	FALSE	FALSE
4751	8909	15187	85200	FALSE	FALSE
4752	7500	15187	143500	FALSE	FALSE
4753	6885	15218	49000	FALSE	FALSE
4754	5400	15218	165000	FALSE	FALSE
4755	13184	15218	95000	FALSE	FALSE
4756	10800	15218	93500	FALSE	FALSE
4757	4914	15218	59900	FALSE	FALSE

4758	5160	15218	26000	FALSE	FALSE
4759	15000	15218	124900	FALSE	FALSE
4760	7000	15218	20600	FALSE	FALSE
4761	8100	15218	87600	FALSE	FALSE
4762	7980	15218	91000	FALSE	FALSE
4763	5850	15218	470000	TRUE	TRUE
4764	8700	15218	460000	TRUE	FALSE
4765	2400	15218	148900	TRUE	TRUE
4766	8469	15218	190000	TRUE	FALSE
4767	3600	15218	236500	TRUE	TRUE
4768	9000	15218	496000	TRUE	TRUE
4769	3373	15218	180000	TRUE	FALSE
4770	3810	15218	154900	TRUE	FALSE
4771	6384	15218	280000	TRUE	TRUE
4772	5535	15218	274000	TRUE	TRUE
4773	4823	15218	94500	FALSE	FALSE
4774	1800	15218	11000	FALSE	FALSE
4775	5760	15218	62500	FALSE	FALSE
4776	3809	15218	80000	FALSE	FALSE
4777	5000	15218	139000	FALSE	FALSE
4778	5400	15218	90500	FALSE	FALSE
4779	5625	15218	144000	FALSE	FALSE
4780	5760	15218	160000	FALSE	FALSE
4781	5896	15218	150000	FALSE	FALSE
4782	5000	15218	139620	FALSE	FALSE
4783	4960	15218	140000	FALSE	FALSE
4784	5160	15218	149500	FALSE	FALSE
4785	6250	15218	102500	FALSE	FALSE
4786	4950	15218	85000	FALSE	FALSE
4787	19365	15218	349900	FALSE	FALSE
4788	8494	15218	123500	FALSE	FALSE
4789	5120	15218	155000	FALSE	FALSE
4790	4800	15218	149900	FALSE	FALSE
4791	7080	15218	85000	FALSE	FALSE
4792	5240	15218	137000	FALSE	FALSE
4793	4800	15218	119000	FALSE	FALSE
4794	16250	15218	115000	FALSE	FALSE
4795	6812	15218	82500	FALSE	FALSE
4796	6750	15218	68000	FALSE	FALSE
4797	10350	15218	85000	FALSE	FALSE
4798	5000	15218	85360	FALSE	FALSE
4799	3600	15218	33800	FALSE	FALSE
4800	5085	15218	194500	FALSE	TRUE

4801	6300	15218	237500	FALSE	FALSE
4802	4872	15218	82500	FALSE	FALSE
4803	4200	15218	19000	FALSE	FALSE
4804	3750	15218	26000	FALSE	FALSE
4805	4800	15218	94000	FALSE	FALSE
4806	4200	15218	85000	FALSE	FALSE
4807	2639	15218	29500	FALSE	FALSE
4808	6600	15218	103000	FALSE	FALSE
4809	3750	15218	95900	FALSE	FALSE
4810	4200	15218	45000	FALSE	FALSE
4811	4200	15218	81000	FALSE	FALSE
4813	7877	15218	146000	FALSE	FALSE
4814	11350	15218	131500	FALSE	FALSE
4815	10860	15218	136000	FALSE	FALSE
4816	3840	15218	82000	FALSE	FALSE
4817	5160	15218	134000	FALSE	FALSE
4818	6500	15218	205000	FALSE	FALSE
4819	5400	15218	80000	FALSE	FALSE
4820	5760	15218	86000	FALSE	FALSE
4821	9450	15218	250000	FALSE	FALSE
4822	5760	15218	118000	FALSE	FALSE
4823	4800	15218	120000	FALSE	FALSE
4824	6000	15218	80000	FALSE	FALSE
4825	5000	15218	60000	FALSE	FALSE
4826	6000	15218	115000	FALSE	FALSE
4827	5280	15218	68000	FALSE	FALSE
4828	6000	15218	114000	FALSE	FALSE
4829	4800	15218	122000	FALSE	FALSE
4830	3750	15218	78000	FALSE	FALSE
4831	6030	15218	143000	FALSE	FALSE
4832	5796	15218	109000	FALSE	FALSE
4833	6298	15218	102000	FALSE	FALSE
4834	4900	15218	140000	FALSE	FALSE
4835	6000	15218	154000	FALSE	TRUE
4836	6600	15218	200000	FALSE	FALSE
4837	10530	15218	191500	FALSE	FALSE
4838	8100	15218	188500	FALSE	FALSE
4839	7200	15218	148000	FALSE	FALSE
4840	8025	15218	130500	FALSE	FALSE
4841	5490	15218	128500	FALSE	FALSE
4842	5160	15218	115000	FALSE	FALSE
4843	6765	15218	127000	FALSE	FALSE
4844	5000	15218	82000	FALSE	FALSE

4845	5000	15218	82000	FALSE	FALSE
4846	5400	15218	83000	FALSE	FALSE
4847	15424	15218	245000	FALSE	FALSE
4848	3600	15218	50800	FALSE	FALSE
4849	5670	15218	151000	FALSE	FALSE
4850	4800	15218	138000	FALSE	FALSE
4851	4850	15218	109500	FALSE	FALSE
4852	4800	15218	135900	FALSE	FALSE
4853	7740	15218	144000	FALSE	FALSE
4854	4800	15218	68500	FALSE	FALSE
4855	6109	15218	60000	FALSE	FALSE
4856	8031	15218	140000	FALSE	FALSE
4857	5828	15218	161000	FALSE	FALSE
4858	4945	15218	134000	FALSE	FALSE
4859	5130	15218	215500	FALSE	FALSE
4860	3600	15218	100000	FALSE	FALSE
4861	3705	15218	164900	FALSE	FALSE
4862	5160	15218	122500	FALSE	FALSE
4863	5160	15218	245000	FALSE	FALSE
4864	6300	15218	128000	FALSE	FALSE
4865	5600	15218	139400	FALSE	FALSE
4866	5000	15218	170000	FALSE	FALSE
4867	7503	15218	164900	FALSE	FALSE
4868	4880	15218	104900	FALSE	FALSE
4869	3648	15218	118000	FALSE	FALSE
4870	2730	15218	133900	FALSE	FALSE
4871	3870	15218	216500	FALSE	FALSE
4872	3025	15218	265000	FALSE	TRUE
4873	4290	15218	174000	FALSE	FALSE
4874	4740	15218	163500	FALSE	FALSE
4875	2880	15218	41000	FALSE	FALSE
4876	3600	15218	78000	FALSE	FALSE
4877	5246	15218	135000	FALSE	FALSE
4878	5987	15218	85000	FALSE	FALSE
4879	5225	15218	75000	FALSE	FALSE
4880	13230	15248	125000	FALSE	FALSE
4881	5480	15248	95000	FALSE	FALSE
4882	11700	15248	115000	FALSE	FALSE
4883	7800	15248	139000	FALSE	FALSE
4884	5408	15248	78000	FALSE	FALSE
4885	5080	15248	37900	FALSE	FALSE
4886	8760	15248	62000	FALSE	FALSE
4887	8400	15248	93000	FALSE	FALSE

4888	5330	15248	92500	FALSE	FALSE
4889	6000	15248	65500	FALSE	FALSE
4890	4840	15248	57500	FALSE	FALSE
4891	6480	15248	55000	FALSE	FALSE
4892	7000	15248	50000	FALSE	FALSE
4893	7980	15248	72900	FALSE	FALSE
4894	3800	15248	210000	TRUE	FALSE
4895	9600	15248	985000	TRUE	TRUE
4896	4840	15248	330000	TRUE	TRUE
4897	6450	15248	390000	TRUE	TRUE
4898	5460	15248	311000	TRUE	TRUE
4899	2670	15248	133500	TRUE	TRUE
4900	7000	15248	78000	FALSE	FALSE
4901	4305	15248	50000	FALSE	FALSE
4902	21780	15248	170000	FALSE	FALSE
4903	6240	15248	167500	FALSE	FALSE
4904	6720	15248	152000	FALSE	FALSE
4905	5120	15248	112000	FALSE	FALSE
4906	5600	15248	119000	FALSE	FALSE
4907	5760	15248	133000	FALSE	FALSE
4908	5240	15248	116500	FALSE	FALSE
4909	6000	15248	100000	FALSE	FALSE
4910	4950	15248	102400	FALSE	FALSE
4911	4800	15248	170500	FALSE	FALSE
4912	5120	15248	105000	FALSE	FALSE
4913	5040	15248	179000	FALSE	FALSE
4914	14933	15248	290000	FALSE	FALSE
4915	11200	15248	143000	FALSE	FALSE
4916	7080	15248	88000	FALSE	FALSE
4917	8450	15248	105000	FALSE	FALSE
4918	8400	15248	108000	FALSE	FALSE
4919	5616	15248	119000	FALSE	FALSE
4920	6490	15248	99500	FALSE	FALSE
4921	4800	15248	73000	FALSE	FALSE
4922	5080	15248	75000	FALSE	FALSE
4923	5520	15248	55000	FALSE	FALSE
4924	5125	15248	81500	FALSE	FALSE
4925	3145	15248	255900	FALSE	FALSE
4926	4125	15248	18000	FALSE	FALSE
4927	3750	15248	17000	FALSE	FALSE
4928	5208	15248	80000	FALSE	FALSE
4929	5160	15248	75000	FALSE	FALSE
4930	5520	15248	84700	FALSE	FALSE

4931	6160	15248	16000	FALSE	FALSE
4932	3810	15248	76000	FALSE	FALSE
4933	3750	15248	66000	FALSE	FALSE
4934	3750	15248	90000	FALSE	FALSE
4935	3750	15248	95000	FALSE	FALSE
4936	3750	15248	67000	FALSE	FALSE
4937	6148	15248	73000	FALSE	FALSE
4938	7719	15248	145000	FALSE	FALSE
4939	6100	15248	100000	FALSE	FALSE
4940	6120	15248	110000	FALSE	FALSE
4941	7860	15248	119000	FALSE	FALSE
4942	5400	15248	198000	FALSE	FALSE
4943	4200	15248	110000	FALSE	FALSE
4944	5040	15248	72500	FALSE	FALSE
4945	7380	15248	136950	FALSE	FALSE
4946	3404	15248	91000	FALSE	FALSE
4947	5000	15248	128500	FALSE	FALSE
4948	5480	15248	125000	FALSE	FALSE
4949	4800	15248	86000	FALSE	FALSE
4950	7150	15248	76800	FALSE	FALSE
4951	4960	15248	131500	FALSE	FALSE
4952	6069	15248	84500	FALSE	FALSE
4953	6700	15248	156900	FALSE	FALSE
4954	4800	15248	103000	FALSE	FALSE
4955	6000	15248	170000	FALSE	FALSE
4956	8083	15248	162900	FALSE	FALSE
4957	7200	15248	101000	FALSE	FALSE
4958	7200	15248	130000	FALSE	FALSE
4959	7800	15248	142000	FALSE	FALSE
4960	6150	15248	120000	FALSE	FALSE
4961	4900	15248	49900	FALSE	FALSE
4962	6600	15248	138000	FALSE	FALSE
4963	9000	15248	205000	FALSE	FALSE
4964	5040	15248	197600	FALSE	FALSE
4965	6948	15248	117500	FALSE	FALSE
4966	7260	15248	135000	FALSE	FALSE
4967	6450	15248	132000	FALSE	FALSE
4968	6840	15248	165000	FALSE	FALSE
4969	6608	15248	125000	FALSE	FALSE
4970	8040	15248	110700	FALSE	FALSE
4971	9120	15248	124000	FALSE	FALSE
4972	5400	15248	97000	FALSE	FALSE
4973	4800	15248	206000	FALSE	TRUE

4974	5600	15248	140000	FALSE	FALSE
4975	3660	15248	128500	FALSE	FALSE
4976	5476	15248	210000	FALSE	FALSE
4977	4200	15248	40000	FALSE	FALSE
4978	3600	15248	177900	FALSE	FALSE
4979	5560	15248	109000	FALSE	FALSE
4980	4590	15248	129000	FALSE	FALSE
4981	4905	15248	300000	FALSE	FALSE
4982	2100	15248	183000	FALSE	FALSE
4983	3600	15248	150000	FALSE	FALSE
4984	3600	15248	199000	FALSE	TRUE
4985	4800	15248	139000	FALSE	FALSE
4986	3600	15248	136500	FALSE	FALSE
4987	10738	15248	48000	FALSE	FALSE
4988	3600	15248	2000	FALSE	FALSE
4989	4080	15248	59000	FALSE	FALSE
4990	4290	15248	21000	FALSE	FALSE
4991	4800	15279	75000	FALSE	FALSE
4992	8910	15279	67500	FALSE	FALSE
4993	5160	15279	118000	FALSE	FALSE
4994	8527	15279	82500	FALSE	FALSE
4995	10133	15279	245000	FALSE	FALSE
4996	7200	15279	69500	FALSE	FALSE
4997	7200	15279	56000	FALSE	FALSE
4998	5750	15279	103900	FALSE	FALSE
4999	7040	15279	62000	FALSE	FALSE
5000	3600	15279	106000	TRUE	TRUE
5001	5850	15279	515000	TRUE	TRUE
5002	3600	15279	206000	TRUE	TRUE
5003	2700	15279	196000	TRUE	TRUE
5004	3600	15279	220000	TRUE	TRUE
5005	6000	15279	138000	FALSE	FALSE
5006	7200	15279	119900	FALSE	FALSE
5007	5240	15279	109000	FALSE	FALSE
5008	4800	15279	56000	FALSE	FALSE
5009	10400	15279	185000	FALSE	FALSE
5010	6120	15279	152500	FALSE	FALSE
5011	6000	15279	119400	FALSE	FALSE
5012	9750	15279	171000	FALSE	FALSE
5013	12000	15279	148500	FALSE	FALSE
5014	6250	15279	157000	FALSE	FALSE
5015	8056	15279	112000	FALSE	FALSE
5016	12864	15279	125000	FALSE	FALSE

5017	6000	15279	100000	FALSE	FALSE
5018	3840	15279	83000	FALSE	FALSE
5019	3000	15279	82000	FALSE	FALSE
5021	4995	15279	92000	FALSE	FALSE
5022	5400	15279	69000	FALSE	FALSE
5023	5336	15279	77500	FALSE	FALSE
5024	4920	15279	89000	FALSE	FALSE
5025	4773	15279	105000	FALSE	FALSE
5026	3840	15279	73000	FALSE	FALSE
5027	4720	15279	95000	FALSE	FALSE
5028	3600	15279	70000	FALSE	FALSE
5029	3663	15279	199900	FALSE	FALSE
5031	6500	15279	65000	FALSE	FALSE
5032	9159	15279	204500	FALSE	FALSE
5033	5400	15279	170000	FALSE	FALSE
5034	5160	15279	144000	FALSE	FALSE
5035	7200	15279	115000	FALSE	FALSE
5036	6500	15279	129000	FALSE	FALSE
5037	4800	15279	99500	FALSE	FALSE
5038	4810	15279	129000	FALSE	FALSE
5039	3955	15279	88000	FALSE	FALSE
5040	5520	15279	192000	FALSE	FALSE
5041	5928	15279	147500	FALSE	FALSE
5042	6272	15279	120000	FALSE	FALSE
5043	4165	15279	122900	FALSE	FALSE
5044	4800	15279	95000	FALSE	FALSE
5045	5382	15279	120000	FALSE	FALSE
5046	11389	15279	185000	FALSE	FALSE
5047	5280	15279	146000	FALSE	FALSE
5048	7854	15279	147000	FALSE	FALSE
5049	8976	15279	110000	FALSE	FALSE
5050	4960	15279	107000	FALSE	FALSE
5051	8040	15279	175900	FALSE	FALSE
5052	5320	15279	143000	FALSE	FALSE
5053	5400	15279	118000	FALSE	FALSE
5054	6360	15279	146300	FALSE	FALSE
5055	5580	15279	102900	FALSE	FALSE
5056	7215	15279	158500	FALSE	FALSE
5057	6700	15279	125900	FALSE	FALSE
5058	8925	15279	151500	FALSE	FALSE
5059	7830	15279	109300	FALSE	FALSE
5060	7371	15279	162000	FALSE	FALSE
5061	7316	15279	120000	FALSE	FALSE

5062	10658	15279	160000	FALSE	FALSE
5063	7344	15279	85000	FALSE	FALSE
5064	6860	15279	80000	FALSE	FALSE
5065	6160	15279	84000	FALSE	FALSE
5066	6750	15279	67500	FALSE	FALSE
5067	2419	15279	50000	FALSE	FALSE
5068	2928	15279	33000	FALSE	FALSE
5069	4200	15279	71000	FALSE	FALSE
5070	9750	15279	192500	FALSE	FALSE
5071	5040	15279	133000	FALSE	FALSE
5072	7872	15279	151000	FALSE	FALSE
5073	5192	15279	95000	FALSE	FALSE
5074	4720	15279	106500	FALSE	FALSE
5075	4800	15279	84900	FALSE	FALSE
5076	4760	15279	108500	FALSE	FALSE
5077	5400	15279	85000	FALSE	FALSE
5078	7920	15279	180000	FALSE	FALSE
5079	3600	15279	79900	FALSE	FALSE
5080	7920	15279	175000	FALSE	FALSE
5081	18727	15279	191000	FALSE	FALSE
5082	6840	15279	166000	FALSE	FALSE
5083	11520	15279	105000	FALSE	FALSE
5084	4680	15279	117500	FALSE	FALSE
5085	3510	15279	85000	FALSE	FALSE
5086	7425	15279	108000	FALSE	FALSE
5087	3600	15279	80000	FALSE	FALSE
5088	4920	15279	202000	FALSE	FALSE
5089	5400	15279	113000	FALSE	FALSE
5090	5400	15279	79400	FALSE	FALSE
5091	5400	15279	184500	FALSE	FALSE
5092	4720	15279	52500	FALSE	FALSE
5095	4770	15309	74000	FALSE	FALSE
5096	6000	15309	75000	FALSE	FALSE
5097	4800	15309	85000	FALSE	FALSE
5098	4800	15309	72000	FALSE	FALSE
5099	4800	15309	59000	FALSE	FALSE
5100	10500	15309	96000	FALSE	FALSE
5101	5320	15309	115000	FALSE	FALSE
5102	5360	15309	74900	FALSE	FALSE
5103	6150	15309	90000	FALSE	FALSE
5104	4800	15309	59000	FALSE	FALSE
5105	7245	15309	68000	FALSE	FALSE
5106	7828	15309	119900	FALSE	FALSE

5107	4720	15309	225000	TRUE	TRUE
5108	4340	15309	134000	TRUE	TRUE
5109	16498	15309	1075000	TRUE	FALSE
5110	15000	15309	1000000	TRUE	TRUE
5111	10950	15309	720000	TRUE	TRUE
5112	4800	15309	362000	TRUE	TRUE
5113	4880	15309	355500	TRUE	TRUE
5114	5400	15309	340000	TRUE	TRUE
5115	4800	15309	259000	TRUE	TRUE
5116	4572	15309	190000	TRUE	TRUE
5117	3480	15309	107000	TRUE	FALSE
5118	9600	15309	724600	TRUE	TRUE
5119	3600	15309	58400	FALSE	FALSE
5120	6840	15309	165000	FALSE	FALSE
5121	21780	15309	213200	FALSE	TRUE
5122	8120	15309	65000	FALSE	FALSE
5123	7517	15309	84700	FALSE	FALSE
5124	17705	15309	326129	FALSE	FALSE
5125	7200	15309	144900	FALSE	FALSE
5126	6250	15309	225000	FALSE	TRUE
5127	6960	15309	92500	FALSE	FALSE
5128	12000	15309	148000	FALSE	FALSE
5129	7320	15309	207000	FALSE	TRUE
5130	5512	15309	92900	FALSE	FALSE
5131	7800	15309	150000	FALSE	FALSE
5132	6890	15309	87500	FALSE	FALSE
5133	6250	15309	119000	FALSE	FALSE
5134	6080	15309	100000	FALSE	FALSE
5135	6000	15309	115100	FALSE	FALSE
5136	7920	15309	140800	FALSE	FALSE
5137	5700	15309	47500	FALSE	FALSE
5138	4686	15309	32000	FALSE	FALSE
5139	3500	15309	38000	FALSE	FALSE
5140	4375	15309	26000	FALSE	FALSE
5141	5250	15309	100000	FALSE	FALSE
5142	5453	15309	38000	FALSE	FALSE
5143	4800	15309	77000	FALSE	FALSE
5144	4840	15309	65000	FALSE	FALSE
5145	4500	15309	56300	FALSE	FALSE
5146	4960	15309	150000	FALSE	FALSE
5147	3600	15309	50000	FALSE	FALSE
5148	3808	15309	95000	FALSE	FALSE
5149	15253	15309	254291	FALSE	FALSE

5150	9000	15309	180000	FALSE	FALSE
5151	7620	15309	87000	FALSE	FALSE
5152	6360	15309	106500	FALSE	FALSE
5153	4480	15309	118000	FALSE	FALSE
5154	3720	15309	105000	FALSE	FALSE
5155	4800	15309	69000	FALSE	FALSE
5156	5355	15309	119000	FALSE	FALSE
5157	5400	15309	95000	FALSE	FALSE
5158	4181	15309	50000	FALSE	FALSE
5159	4800	15309	110000	FALSE	FALSE
5160	4641	15309	95000	FALSE	FALSE
5161	4760	15309	100000	FALSE	FALSE
5162	9200	15309	79000	FALSE	FALSE
5163	3600	15309	117000	FALSE	FALSE
5164	4884	15309	133000	FALSE	FALSE
5165	4725	15309	107000	FALSE	FALSE
5166	7920	15309	38000	FALSE	FALSE
5167	12940	15309	167000	FALSE	FALSE
5168	9610	15309	156000	FALSE	FALSE
5169	8040	15309	148500	FALSE	FALSE
5170	6000	15309	101000	FALSE	FALSE
5171	6240	15309	86000	FALSE	FALSE
5172	5000	15309	125500	FALSE	FALSE
5173	6250	15309	120000	FALSE	FALSE
5174	7865	15309	120000	FALSE	FALSE
5175	5959	15309	105000	FALSE	FALSE
5176	5060	15309	99900	FALSE	FALSE
5177	5500	15309	61500	FALSE	FALSE
5178	5625	15309	60000	FALSE	FALSE
5179	4653	15309	42000	FALSE	FALSE
5180	4680	15309	171000	FALSE	FALSE
5181	5805	15309	85000	FALSE	FALSE
5182	4428	15309	147500	FALSE	FALSE
5183	4800	15309	95000	FALSE	FALSE
5184	4800	15309	120000	FALSE	FALSE
5185	7128	15309	175300	FALSE	FALSE
5186	8100	15309	110000	FALSE	FALSE
5187	4788	15309	90000	FALSE	FALSE
5188	4880	15309	88000	FALSE	FALSE
5189	4998	15309	165000	FALSE	FALSE
5190	4636	15309	99000	FALSE	FALSE
5191	7200	15309	233350	FALSE	TRUE
5192	3510	15309	164200	FALSE	FALSE

5193	7800	15309	24900	FALSE	FALSE
5194	5400	15309	45500	FALSE	FALSE
5195	5880	15340	35000	FALSE	FALSE
5196	3270	15340	34507	FALSE	FALSE
5197	5250	15340	112900	FALSE	FALSE
5198	4800	15340	67000	FALSE	FALSE
5199	21571	15340	194800	FALSE	FALSE
5200	6480	15340	88950	FALSE	FALSE
5201	7860	15340	99000	FALSE	FALSE
5202	4800	15340	63000	FALSE	FALSE
5203	4920	15340	68000	FALSE	FALSE
5204	5400	15340	89000	FALSE	FALSE
5205	4800	15340	62000	FALSE	FALSE
5206	8820	15340	102500	FALSE	FALSE
5207	5800	15340	610000	TRUE	TRUE
5208	5400	15340	150000	TRUE	FALSE
5209	2869	15340	149900	TRUE	TRUE
5210	7200	15340	580000	TRUE	TRUE
5211	4800	15340	280000	TRUE	TRUE
5212	5040	15340	142500	FALSE	FALSE
5213	7000	15340	113000	FALSE	FALSE
5214	6000	15340	82000	FALSE	FALSE
5215	4800	15340	70000	FALSE	FALSE
5216	5014	15340	62000	FALSE	FALSE
5217	8400	15340	131000	FALSE	FALSE
5218	6624	15340	115000	FALSE	FALSE
5219	10500	15340	30000	FALSE	FALSE
5220	6000	15340	42000	FALSE	FALSE
5221	3120	15340	29000	FALSE	FALSE
5222	4797	15340	119000	FALSE	FALSE
5223	4551	15340	89000	FALSE	FALSE
5224	4800	15340	89900	FALSE	FALSE
5225	5160	15340	105000	FALSE	FALSE
5226	4280	15340	31500	FALSE	FALSE
5227	3270	15340	50000	FALSE	FALSE
5228	5760	15340	92000	FALSE	FALSE
5229	3570	15340	92000	FALSE	FALSE
5230	23700	15340	60000	FALSE	FALSE
5231	9592	15340	128000	FALSE	FALSE
5232	5460	15340	87000	FALSE	FALSE
5233	3840	15340	45000	FALSE	FALSE
5234	3720	15340	87500	FALSE	FALSE
5235	6230	15340	132000	FALSE	FALSE

5236	5760	15340	155000	FALSE	FALSE
5237	3660	15340	110000	FALSE	FALSE
5238	4520	15340	150000	FALSE	FALSE
5239	7192	15340	87000	FALSE	FALSE
5240	8060	15340	170000	FALSE	FALSE
5241	4900	15340	110000	FALSE	FALSE
5242	6500	15340	120000	FALSE	FALSE
5243	4880	15340	116800	FALSE	FALSE
5244	7208	15340	100000	FALSE	FALSE
5245	4960	15340	144900	FALSE	FALSE
5246	8100	15340	145000	FALSE	FALSE
5247	6240	15340	87500	FALSE	FALSE
5248	5060	15340	83000	FALSE	FALSE
5249	4953	15340	112000	FALSE	FALSE
5250	4875	15340	96000	FALSE	FALSE
5251	4800	15340	148000	FALSE	FALSE
5252	4753	15340	103500	FALSE	FALSE
5253	4680	15340	125000	FALSE	FALSE
5254	6600	15340	164900	FALSE	FALSE
5255	4800	15340	99900	FALSE	FALSE
5256	5880	15340	158000	FALSE	FALSE
5257	9450	15340	179000	FALSE	TRUE
5258	8280	15340	137000	FALSE	FALSE
5259	4880	15340	75000	FALSE	FALSE
5260	5490	15340	80000	FALSE	FALSE
5261	4440	15340	182400	FALSE	FALSE
5262	3600	15340	168400	FALSE	FALSE
5263	5265	15340	252000	FALSE	FALSE
5264	7200	15340	220000	FALSE	FALSE
5265	3296	15340	127500	FALSE	FALSE
5266	5198	15340	164500	FALSE	FALSE
5267	3660	15340	82000	FALSE	FALSE
5268	6000	15340	150000	FALSE	TRUE
5269	5289	15340	110000	FALSE	FALSE
5270	5542	15371	73027	FALSE	FALSE
5271	9280	15371	92000	FALSE	FALSE
5272	6930	15371	74000	FALSE	FALSE
5273	5460	15371	105000	FALSE	FALSE
5274	8064	15371	130000	FALSE	FALSE
5275	6890	15371	380000	TRUE	TRUE
5276	2133	15371	170000	TRUE	FALSE
5277	4800	15371	300000	TRUE	FALSE
5278	3840	15371	128000	TRUE	FALSE

5279	9600	15371	545000	TRUE	FALSE
5280	4800	15371	224900	TRUE	FALSE
5281	3000	15371	62000	FALSE	FALSE
5282	4800	15371	122000	FALSE	FALSE
5283	15500	15371	144000	FALSE	FALSE
5284	5000	15371	121800	FALSE	FALSE
5285	8241	15371	85000	FALSE	FALSE
5286	7800	15371	131000	FALSE	FALSE
5287	5750	15371	118900	FALSE	FALSE
5288	7015	15371	96000	FALSE	FALSE
5290	6480	15371	95500	FALSE	FALSE
5291	8400	15371	67000	FALSE	FALSE
5292	6400	15371	95000	FALSE	FALSE
5293	4500	15371	79000	FALSE	FALSE
5294	5850	15371	112600	FALSE	FALSE
5295	4200	15371	60000	FALSE	FALSE
5296	11321	15371	139000	FALSE	FALSE
5297	11800	15371	115000	FALSE	FALSE
5298	10800	15371	82000	FALSE	FALSE
5299	5120	15371	122300	FALSE	FALSE
5300	6000	15371	217000	FALSE	FALSE
5301	7248	15371	230000	FALSE	FALSE
5302	4800	15371	64000	FALSE	FALSE
5303	5400	15371	130500	FALSE	FALSE
5304	9000	15371	95000	FALSE	FALSE
5305	8189	15371	110000	FALSE	FALSE
5306	7000	15371	152000	FALSE	FALSE
5307	5400	15371	120000	FALSE	FALSE
5308	5520	15371	185000	FALSE	FALSE
5309	8100	15371	117500	FALSE	FALSE
5310	7680	15371	190000	FALSE	FALSE
5311	8083	15371	183500	FALSE	TRUE
5312	4840	15371	149900	FALSE	FALSE
5313	7250	15371	187500	FALSE	FALSE
5314	7140	15371	137500	FALSE	FALSE
5315	6400	15371	105000	FALSE	FALSE
5316	4000	15371	142000	FALSE	FALSE
5317	5106	15371	115500	FALSE	FALSE
5318	3660	15371	37000	FALSE	FALSE
5319	4200	15371	45000	FALSE	FALSE
5320	5850	15371	174000	FALSE	FALSE
5321	3390	15371	150000	FALSE	FALSE
5322	3600	15371	59000	FALSE	FALSE

5323	1875	15371	143500	FALSE	FALSE
5324	7500	15371	42000	FALSE	FALSE
5325	5000	15400	89900	FALSE	FALSE
5326	5640	15400	62000	FALSE	FALSE
5327	4440	15400	59000	FALSE	FALSE
5328	4360	15400	83000	FALSE	FALSE
5329	4800	15400	107000	FALSE	FALSE
5330	4800	15400	97500	FALSE	FALSE
5331	4880	15400	220000	TRUE	FALSE
5332	5520	15400	142000	TRUE	FALSE
5333	5400	15400	340000	TRUE	TRUE
5334	6000	15400	254500	TRUE	TRUE
5335	7800	15400	190000	FALSE	FALSE
5336	4800	15400	136000	FALSE	FALSE
5337	7200	15400	90000	FALSE	FALSE
5338	6250	15400	95000	FALSE	FALSE
5339	5474	15400	90000	FALSE	FALSE
5340	5120	15400	150700	FALSE	FALSE
5341	5240	15400	112500	FALSE	FALSE
5342	6250	15400	118000	FALSE	FALSE
5343	8750	15400	299900	FALSE	FALSE
5344	17408	15400	291000	FALSE	FALSE
5345	5400	15400	132900	FALSE	FALSE
5346	11280	15400	257500	FALSE	TRUE
5347	11475	15400	187900	FALSE	FALSE
5348	13216	15400	156000	FALSE	FALSE
5349	5000	15400	106000	FALSE	FALSE
5350	5940	15400	123000	FALSE	FALSE
5351	8874	15400	115000	FALSE	FALSE
5352	3600	15400	93000	FALSE	FALSE
5353	2000	15400	158000	FALSE	FALSE
5354	3300	15400	120000	FALSE	FALSE
5355	4800	15400	83000	FALSE	FALSE
5356	5040	15400	138000	FALSE	FALSE
5357	4888	15400	145000	FALSE	FALSE
5358	7560	15400	75000	FALSE	FALSE
5359	3600	15400	65000	FALSE	FALSE
5360	1740	15400	87650	FALSE	FALSE
5361	3960	15400	28000	FALSE	FALSE
5363	10607	15400	242000	FALSE	FALSE
5364	8192	15400	109900	FALSE	FALSE
5365	7200	15400	115000	FALSE	FALSE
5366	3600	15400	125000	FALSE	FALSE

5367	9450	15400	154000	FALSE	FALSE
5368	4920	15400	106900	FALSE	FALSE
5369	6630	15400	109000	FALSE	FALSE
5370	4524	15400	104900	FALSE	FALSE
5371	4800	15400	72500	FALSE	FALSE
5372	5080	15400	169000	FALSE	FALSE
5373	4600	15400	137000	FALSE	FALSE
5374	4760	15400	76500	FALSE	FALSE
5375	5400	15400	42000	FALSE	FALSE
5376	4520	15400	79900	FALSE	FALSE
5377	4800	15400	130000	FALSE	FALSE
5378	5300	15400	88000	FALSE	FALSE
5379	6500	15400	119500	FALSE	FALSE
5380	6000	15400	153000	FALSE	FALSE
5381	5480	15400	125000	FALSE	FALSE
5382	7500	15400	90000	FALSE	FALSE
5383	5040	15400	165000	FALSE	FALSE
5384	4920	15400	158000	FALSE	FALSE
5385	4900	15400	94000	FALSE	FALSE
5386	8100	15400	88000	FALSE	FALSE
5387	11440	15400	192400	FALSE	FALSE
5388	5400	15400	126500	FALSE	FALSE
5389	8253	15400	155900	FALSE	FALSE
5390	7500	15400	125000	FALSE	FALSE
5391	10000	15400	137500	FALSE	FALSE
5392	7875	15400	110000	FALSE	FALSE
5393	6240	15400	62500	FALSE	FALSE
5394	9412	15400	85500	FALSE	FALSE
5395	5040	15400	122000	FALSE	FALSE
5396	6480	15400	96500	FALSE	FALSE
5397	4290	15400	83000	FALSE	FALSE
5398	3300	15400	54500	FALSE	FALSE
5399	2820	15400	90000	FALSE	FALSE
5400	4720	15400	80000	FALSE	FALSE
5401	6000	15400	144000	FALSE	FALSE
5402	8772	15400	117000	FALSE	FALSE
5403	6150	15400	83000	FALSE	FALSE
5404	5400	15400	80000	FALSE	FALSE
5405	6165	15400	149900	FALSE	FALSE
5406	5640	15400	136000	FALSE	FALSE
5407	3540	15400	137000	FALSE	TRUE
5408	8640	15400	122000	FALSE	FALSE
5409	10480	15400	147500	FALSE	FALSE

5410	6943	15400	109000	FALSE	FALSE
5411	6000	15400	115000	FALSE	FALSE
5412	7560	15400	95000	FALSE	FALSE
5413	6837	15400	127500	FALSE	FALSE
5414	4500	15400	120000	FALSE	FALSE
5415	5060	15400	114500	FALSE	FALSE
5416	2650	15400	282000	FALSE	TRUE
5417	3690	15400	132500	FALSE	FALSE
5418	4880	15400	129000	FALSE	FALSE
5419	6050	15400	190000	FALSE	FALSE
5420	4560	15400	139900	FALSE	FALSE
5421	7254	15400	360000	FALSE	TRUE
5422	3600	15400	250000	FALSE	TRUE
5423	11220	15400	167000	FALSE	FALSE
5424	5806	15400	77000	FALSE	FALSE
5425	4000	15400	28500	FALSE	FALSE
5426	5187	15400	99900	FALSE	FALSE
5427	4800	15400	58000	FALSE	FALSE
5428	4680	15431	61000	FALSE	FALSE
5429	7560	15431	74700	FALSE	FALSE
5430	5320	15431	71000	FALSE	FALSE
5431	4800	15431	42900	FALSE	FALSE
5432	12320	15431	81000	FALSE	FALSE
5433	8040	15431	91000	FALSE	FALSE
5434	5040	15431	69000	FALSE	FALSE
5435	6000	15431	72000	FALSE	FALSE
5436	9000	15431	720000	TRUE	TRUE
5437	5400	15431	170000	TRUE	FALSE
5438	4800	15431	248000	TRUE	TRUE
5439	2400	15431	135000	TRUE	TRUE
5440	4800	15431	195000	TRUE	TRUE
5441	12700	15431	580000	TRUE	TRUE
5442	2220	15431	207000	TRUE	TRUE
5443	4920	15431	169000	TRUE	FALSE
5444	3960	15431	130000	FALSE	FALSE
5445	3390	15431	11500	FALSE	FALSE
5446	4800	15431	152000	FALSE	FALSE
5447	5632	15431	117900	FALSE	FALSE
5448	5120	15431	83000	FALSE	FALSE
5449	5376	15431	83000	FALSE	FALSE
5450	6552	15431	100000	FALSE	FALSE
5451	7200	15431	97900	FALSE	FALSE
5452	5240	15431	77500	FALSE	FALSE

5453	5060	15431	106000	FALSE	FALSE
5454	5360	15431	127500	FALSE	FALSE
5455	5000	15431	125000	FALSE	FALSE
5456	21780	15431	198000	FALSE	FALSE
5457	11025	15431	135000	FALSE	FALSE
5458	7200	15431	148900	FALSE	FALSE
5459	5280	15431	122000	FALSE	FALSE
5460	18760	15431	149000	FALSE	FALSE
5461	9141	15431	155000	FALSE	FALSE
5462	6360	15431	80000	FALSE	FALSE
5463	5200	15431	80000	FALSE	FALSE
5464	6407	15431	115000	FALSE	FALSE
5465	7200	15431	108000	FALSE	FALSE
5466	7130	15431	57000	FALSE	FALSE
5467	5100	15431	33300	FALSE	FALSE
5468	5324	15431	84900	FALSE	FALSE
5469	5160	15431	85000	FALSE	FALSE
5470	4800	15431	70000	FALSE	FALSE
5471	4988	15431	89000	FALSE	FALSE
5472	5000	15431	30000	FALSE	FALSE
5473	5160	15431	149000	FALSE	FALSE
5474	5586	15431	113000	FALSE	FALSE
5476	5160	15431	68000	FALSE	FALSE
5477	3750	15431	90000	FALSE	FALSE
5478	5715	15431	94000	FALSE	FALSE
5479	3540	15431	84900	FALSE	FALSE
5480	7200	15431	154000	FALSE	FALSE
5481	10000	15431	138000	FALSE	FALSE
5482	9000	15431	111000	FALSE	FALSE
5483	5280	15431	150000	FALSE	FALSE
5484	4800	15431	178500	FALSE	FALSE
5485	6000	15431	131000	FALSE	FALSE
5486	9461	15431	73000	FALSE	FALSE
5487	3840	15431	65000	FALSE	FALSE
5488	5400	15431	117500	FALSE	FALSE
5489	7595	15431	160000	FALSE	FALSE
5490	7200	15431	34000	FALSE	FALSE
5491	4560	15431	135000	FALSE	FALSE
5492	4872	15431	108000	FALSE	FALSE
5493	4320	15431	87500	FALSE	FALSE
5494	5280	15431	128000	FALSE	FALSE
5495	4800	15431	150000	FALSE	FALSE
5496	5080	15431	177500	FALSE	FALSE

5497	4200	15431	137500	FALSE	FALSE
5498	4620	15431	147900	FALSE	FALSE
5499	4760	15431	67500	FALSE	FALSE
5500	5400	15431	128000	FALSE	FALSE
5501	3680	15431	92000	FALSE	FALSE
5502	6374	15431	118000	FALSE	FALSE
5503	5400	15431	245000	FALSE	FALSE
5504	4920	15431	175000	FALSE	FALSE
5505	4800	15431	118500	FALSE	FALSE
5507	5906	15431	95000	FALSE	FALSE
5508	6579	15431	129500	FALSE	FALSE
5509	15300	15431	284000	FALSE	FALSE
5510	8400	15431	185000	FALSE	FALSE
5511	4884	15431	145000	FALSE	FALSE
5512	6528	15431	135000	FALSE	FALSE
5513	9106	15431	146500	FALSE	TRUE
5514	7223	15431	149000	FALSE	FALSE
5515	7200	15431	175000	FALSE	FALSE
5516	4920	15431	115000	FALSE	FALSE
5517	9864	15431	169900	FALSE	FALSE
5518	4876	15431	160000	FALSE	FALSE
5519	8100	15431	134900	FALSE	FALSE
5520	6450	15431	127500	FALSE	FALSE
5521	8174	15431	98000	FALSE	FALSE
5522	6240	15431	82000	FALSE	FALSE
5523	4640	15431	196900	FALSE	FALSE
5524	3720	15431	80000	FALSE	FALSE
5525	4230	15431	94000	FALSE	FALSE
5526	3720	15431	80000	FALSE	FALSE
5527	3600	15431	80000	FALSE	FALSE
5528	3000	15431	65000	FALSE	FALSE
5529	21780	15431	214000	FALSE	FALSE
5530	5289	15431	52500	FALSE	FALSE
5531	7500	15431	159000	FALSE	FALSE
5532	4800	15431	110000	FALSE	FALSE
5533	4704	15431	120000	FALSE	FALSE
5534	6450	15431	150000	FALSE	FALSE
5535	4641	15431	119900	FALSE	FALSE
5536	14400	15431	230000	FALSE	FALSE
5537	6450	15431	149900	FALSE	FALSE
5538	6000	15431	109300	FALSE	FALSE
5539	4920	15431	35000	FALSE	FALSE
5540	7680	15431	129500	FALSE	FALSE

5541	7560	15431	120000	FALSE	FALSE
5542	930	15431	17000	FALSE	FALSE
5543	10212	15431	119000	FALSE	FALSE
5544	6000	15431	119900	FALSE	FALSE
5545	4838	15431	117400	FALSE	FALSE
5546	3270	15431	315000	FALSE	TRUE
5547	5828	15431	142500	FALSE	FALSE
5548	3780	15431	108000	FALSE	FALSE
5549	3600	15431	220000	FALSE	TRUE
5550	3510	15431	202000	FALSE	FALSE
5551	3600	15431	160900	FALSE	FALSE
5552	5900	15431	95000	FALSE	FALSE
5553	4500	15431	167000	FALSE	FALSE
5554	11250	15431	172500	FALSE	FALSE
5555	4560	15431	165000	FALSE	FALSE
5556	3660	15431	117000	FALSE	FALSE
5557	3420	15431	121000	FALSE	FALSE
5558	2870	15431	134900	FALSE	FALSE
5560	4920	15431	99000	FALSE	FALSE
5561	4853	15431	133000	FALSE	FALSE
5562	10452	15461	64000	FALSE	FALSE
5563	4242	15461	37725	FALSE	FALSE
5564	7867	15461	119000	FALSE	FALSE
5565	5040	15461	83000	FALSE	FALSE
5566	5460	15461	95000	FALSE	FALSE
5567	5520	15461	125000	FALSE	FALSE
5568	10720	15461	63000	FALSE	FALSE
5569	8192	15461	80000	FALSE	FALSE
5570	7680	15461	104000	FALSE	FALSE
5571	2750	15461	87000	TRUE	FALSE
5572	4200	15461	229500	TRUE	TRUE
5573	10050	15461	640000	TRUE	FALSE
5574	7500	15461	525000	TRUE	TRUE
5575	4305	15461	252000	TRUE	TRUE
5576	3600	15461	182000	TRUE	TRUE
5577	4800	15461	145000	TRUE	TRUE
5578	3600	15461	102000	TRUE	FALSE
5579	1200	15461	110000	TRUE	FALSE
5580	7200	15461	500000	TRUE	FALSE
5581	5800	15461	520000	TRUE	TRUE
5582	6150	15461	63000	FALSE	FALSE
5583	5355	15461	152000	FALSE	FALSE
5584	5750	15461	160000	FALSE	FALSE

5585	6000	15461	177500	FALSE	TRUE
5586	4800	15461	77500	FALSE	FALSE
5587	4800	15461	88000	FALSE	FALSE
5588	4838	15461	101500	FALSE	FALSE
5589	5760	15461	120000	FALSE	FALSE
5590	4687	15461	112000	FALSE	FALSE
5591	5000	15461	135500	FALSE	FALSE
5592	6250	15461	131500	FALSE	FALSE
5593	9000	15461	136500	FALSE	FALSE
5594	5588	15461	90000	FALSE	FALSE
5595	5000	15461	95000	FALSE	FALSE
5596	5650	15461	171500	FALSE	FALSE
5597	4800	15461	151000	FALSE	FALSE
5598	6192	15461	144000	FALSE	FALSE
5599	5280	15461	110000	FALSE	FALSE
5600	11830	15461	79000	FALSE	FALSE
5601	4800	15461	84500	FALSE	FALSE
5602	8060	15461	86000	FALSE	FALSE
5603	4800	15461	122900	FALSE	FALSE
5604	8220	15461	110000	FALSE	FALSE
5605	6000	15461	92000	FALSE	FALSE
5606	6250	15461	110500	FALSE	FALSE
5607	5520	15461	80000	FALSE	FALSE
5608	6000	15461	17302	FALSE	FALSE
5609	4440	15461	131000	FALSE	FALSE
5610	7920	15461	51800	FALSE	FALSE
5611	5054	15461	180000	FALSE	FALSE
5612	5520	15461	66000	FALSE	FALSE
5613	5160	15461	125000	FALSE	FALSE
5614	5125	15461	77000	FALSE	FALSE
5615	4920	15461	57000	FALSE	FALSE
5616	4800	15461	78000	FALSE	FALSE
5617	6000	15461	149000	FALSE	FALSE
5618	4200	15461	35000	FALSE	FALSE
5619	3750	15461	105000	FALSE	FALSE
5620	3582	15461	92000	FALSE	FALSE
5621	3600	15461	70000	FALSE	FALSE
5622	9118	15461	214900	FALSE	FALSE
5623	10763	15461	188200	FALSE	FALSE
5624	9000	15461	115000	FALSE	FALSE
5625	11250	15461	137900	FALSE	FALSE
5626	7787	15461	122900	FALSE	FALSE
5627	13875	15461	117000	FALSE	FALSE

5628	18120	15461	65000	FALSE	FALSE
5629	6150	15461	149500	FALSE	FALSE
5630	5200	15461	125000	FALSE	FALSE
5631	5400	15461	132000	FALSE	FALSE
5632	4920	15461	64500	FALSE	FALSE
5633	5450	15461	84800	FALSE	FALSE
5634	5580	15461	104000	FALSE	FALSE
5635	4998	15461	214000	FALSE	FALSE
5636	5418	15461	135000	FALSE	FALSE
5637	6413	15461	156000	FALSE	FALSE
5638	3600	15461	100000	FALSE	FALSE
5639	4680	15461	213000	FALSE	FALSE
5640	4879	15461	155000	FALSE	FALSE
5641	4560	15461	165000	FALSE	FALSE
5642	4738	15461	136400	FALSE	FALSE
5643	4816	15461	134500	FALSE	FALSE
5644	6720	15461	129000	FALSE	FALSE
5645	4800	15461	75000	FALSE	FALSE
5646	4760	15461	86425	FALSE	FALSE
5647	6592	15461	75000	FALSE	FALSE
5648	7400	15461	188500	FALSE	FALSE
5649	5000	15461	137500	FALSE	FALSE
5650	6930	15461	240000	FALSE	FALSE
5651	4800	15461	160000	FALSE	FALSE
5652	5240	15461	142900	FALSE	FALSE
5653	8040	15461	132000	FALSE	FALSE
5654	4800	15461	115000	FALSE	FALSE
5655	4800	15461	119900	FALSE	FALSE
5656	5080	15461	130000	FALSE	FALSE
5657	5960	15461	98000	FALSE	FALSE
5658	7590	15461	215000	FALSE	FALSE
5659	5040	15461	102500	FALSE	FALSE
5660	4879	15461	156600	FALSE	FALSE
5661	9750	15461	159300	FALSE	TRUE
5662	4800	15461	110000	FALSE	FALSE
5663	5445	15461	103500	FALSE	FALSE
5664	6750	15461	142000	FALSE	FALSE
5665	6664	15461	147000	FALSE	FALSE
5666	8064	15461	113000	FALSE	FALSE
5667	6500	15461	120100	FALSE	FALSE
5668	8440	15461	112000	FALSE	FALSE
5669	6240	15461	101600	FALSE	FALSE
5670	5576	15461	95000	FALSE	FALSE

5671	8100	15461	175000	FALSE	FALSE
5672	4000	15461	90000	FALSE	FALSE
5673	8280	15461	140000	FALSE	FALSE
5674	4760	15461	126000	FALSE	FALSE
5675	6069	15461	109900	FALSE	FALSE
5676	5016	15461	100000	FALSE	FALSE
5677	4800	15461	180000	FALSE	FALSE
5678	5500	15461	108500	FALSE	FALSE
5679	6000	15461	160000	FALSE	FALSE
5680	10640	15461	164000	FALSE	FALSE
5681	20001	15461	203000	FALSE	FALSE
5682	9840	15461	146000	FALSE	FALSE
5683	4884	15461	122000	FALSE	FALSE
5684	8480	15461	124400	FALSE	FALSE
5685	6608	15461	117500	FALSE	FALSE
5686	5202	15461	106700	FALSE	FALSE
5687	4760	15461	145000	FALSE	FALSE
5688	3600	15461	83300	FALSE	FALSE
5689	4800	15461	269900	FALSE	FALSE
5690	7560	15461	148000	FALSE	FALSE
5691	5520	15461	112000	FALSE	FALSE
5692	3150	15461	190000	FALSE	FALSE
5693	4880	15461	70100	FALSE	FALSE
5694	4880	15461	129000	FALSE	FALSE
5695	4200	15461	205000	FALSE	TRUE
5696	4905	15461	360000	FALSE	TRUE
5697	4800	15461	215000	FALSE	FALSE
5698	3600	15461	168000	FALSE	FALSE
5699	4880	15461	80000	FALSE	FALSE
5700	4680	15461	154000	FALSE	FALSE
5701	3660	15461	80000	FALSE	FALSE
5702	7000	15461	142000	FALSE	FALSE
5703	4720	15461	140100	FALSE	FALSE
5704	5880	15461	111000	FALSE	FALSE
5705	6634	15461	134900	FALSE	FALSE
5706	4800	15461	94400	FALSE	FALSE
5707	4720	15461	122900	FALSE	FALSE
5708	2050	15461	218000	FALSE	FALSE
5709	4940	15461	175500	FALSE	FALSE
5710	4500	15461	198000	FALSE	FALSE
5711	3660	15461	99000	FALSE	FALSE
5712	4050	15461	170000	FALSE	FALSE
5713	3540	15461	94000	FALSE	FALSE

5714	10574	15461	121000	FALSE	FALSE
5715	6397	15461	81200	FALSE	FALSE
5716	5000	15461	26000	FALSE	FALSE
5717	3600	15461	40750	FALSE	FALSE
5718	3840	15461	15000	FALSE	FALSE
5719	3840	15461	23500	FALSE	FALSE
5720	7100	15492	85900	FALSE	FALSE
5721	10584	15492	115000	FALSE	FALSE
5722	4920	15492	107000	FALSE	FALSE
5723	5160	15492	65000	FALSE	FALSE
5724	10368	15492	119900	FALSE	FALSE
5725	5280	15492	54000	FALSE	FALSE
5726	4800	15492	97000	FALSE	FALSE
5727	4840	15492	240500	TRUE	TRUE
5728	2500	15492	279000	TRUE	TRUE
5729	3600	15492	169900	TRUE	TRUE
5730	3600	15492	120000	TRUE	FALSE
5731	7200	15492	477000	TRUE	TRUE
5732	7200	15492	370000	TRUE	FALSE
5733	4880	15492	390000	TRUE	TRUE
5734	4000	15492	257000	TRUE	TRUE
5735	2400	15492	185000	TRUE	FALSE
5736	3600	15492	169900	TRUE	TRUE
5737	2640	15492	199900	TRUE	TRUE
5738	2670	15492	134000	TRUE	FALSE
5739	3600	15492	153900	TRUE	FALSE
5740	3600	15492	209900	TRUE	TRUE
5741	7020	15492	520000	TRUE	FALSE
5742	10425	15492	144250	FALSE	FALSE
5743	6240	15492	152000	FALSE	FALSE
5744	5000	15492	139000	FALSE	FALSE
5745	5120	15492	128000	FALSE	FALSE
5746	6307	15492	116500	FALSE	FALSE
5747	4800	15492	102000	FALSE	FALSE
5748	4838	15492	107500	FALSE	FALSE
5749	6000	15492	148000	FALSE	FALSE
5750	5400	15492	164000	FALSE	FALSE
5751	4800	15492	142000	FALSE	FALSE
5752	6800	15492	73000	FALSE	FALSE
5753	20624	15492	322500	FALSE	FALSE
5754	4800	15492	167500	FALSE	FALSE
5755	4800	15492	96500	FALSE	FALSE
5756	7800	15492	183500	FALSE	FALSE

5757	21780	15492	141900	FALSE	FALSE
5758	5120	15492	78000	FALSE	FALSE
5759	7920	15492	90000	FALSE	FALSE
5760	5000	15492	131800	FALSE	FALSE
5761	6000	15492	82000	FALSE	FALSE
5762	6360	15492	114500	FALSE	FALSE
5763	6000	15492	85500	FALSE	FALSE
5764	5400	15492	95000	FALSE	FALSE
5765	7875	15492	86766	FALSE	FALSE
5766	8214	15492	100000	FALSE	FALSE
5767	5120	15492	89500	FALSE	FALSE
5768	4995	15492	57900	FALSE	FALSE
5769	5440	15492	45000	FALSE	FALSE
5770	3330	15492	30000	FALSE	FALSE
5771	12000	15492	51000	FALSE	FALSE
5773	5000	15492	84500	FALSE	FALSE
5774	3795	15492	69000	FALSE	FALSE
5775	2520	15492	50000	FALSE	FALSE
5776	4140	15492	73000	FALSE	FALSE
5777	2678	15492	23500	FALSE	FALSE
5778	18224	15492	100000	FALSE	FALSE
5779	13000	15492	220000	FALSE	FALSE
5780	14400	15492	144000	FALSE	FALSE
5781	9180	15492	119000	FALSE	FALSE
5782	10000	15492	125000	FALSE	FALSE
5783	7200	15492	107000	FALSE	FALSE
5784	6655	15492	109900	FALSE	FALSE
5785	6148	15492	76900	FALSE	FALSE
5786	7398	15492	69000	FALSE	FALSE
5787	7208	15492	94900	FALSE	FALSE
5788	7200	15492	85000	FALSE	FALSE
5789	12900	15492	121500	FALSE	FALSE
5790	5974	15492	137000	FALSE	FALSE
5791	6110	15492	143000	FALSE	FALSE
5792	5400	15492	150000	FALSE	FALSE
5793	5934	15492	177900	FALSE	FALSE
5794	6250	15492	199000	FALSE	FALSE
5795	5400	15492	55600	FALSE	FALSE
5796	4800	15492	80000	FALSE	FALSE
5797	6600	15492	142000	FALSE	FALSE
5798	4510	15492	160000	FALSE	FALSE
5799	6600	15492	77500	FALSE	FALSE
5800	4640	15492	16500	FALSE	FALSE

5801	4142	15492	121000	FALSE	FALSE
5802	6600	15492	127500	FALSE	FALSE
5803	3780	15492	109000	FALSE	FALSE
5804	8400	15492	99900	FALSE	FALSE
5805	4800	15492	113000	FALSE	FALSE
5806	4410	15492	138000	FALSE	FALSE
5807	6180	15492	162000	FALSE	FALSE
5808	4046	15492	37500	FALSE	FALSE
5809	5768	15492	99500	FALSE	FALSE
5810	5800	15492	155000	FALSE	FALSE
5811	8568	15492	145000	FALSE	FALSE
5812	10680	15492	145000	FALSE	FALSE
5813	7980	15492	87900	FALSE	FALSE
5814	5625	15492	134000	FALSE	FALSE
5815	6750	15492	125000	FALSE	FALSE
5816	4920	15492	119000	FALSE	FALSE
5817	7980	15492	77500	FALSE	FALSE
5818	7480	15492	200000	FALSE	FALSE
5819	4799	15492	275000	FALSE	FALSE
5820	4920	15492	121500	FALSE	FALSE
5821	4960	15492	145000	FALSE	FALSE
5822	6784	15492	150000	FALSE	FALSE
5823	4920	15492	139000	FALSE	FALSE
5824	5400	15492	150000	FALSE	FALSE
5825	4960	15492	170700	FALSE	FALSE
5826	8450	15492	195900	FALSE	FALSE
5827	8816	15492	156000	FALSE	FALSE
5828	6840	15492	139900	FALSE	FALSE
5829	8640	15492	121000	FALSE	FALSE
5830	5500	15492	120000	FALSE	FALSE
5831	9850	15492	61000	FALSE	FALSE
5832	6545	15492	152500	FALSE	FALSE
5833	7590	15492	150000	FALSE	FALSE
5834	6800	15492	128000	FALSE	FALSE
5835	6188	15492	120000	FALSE	FALSE
5836	5750	15492	166000	FALSE	FALSE
5837	5500	15492	68000	FALSE	FALSE
5838	9144	15492	222000	FALSE	FALSE
5840	4120	15492	58774	FALSE	FALSE
5841	5192	15492	147500	FALSE	FALSE
5842	4840	15492	156500	FALSE	FALSE
5843	9000	15492	160000	FALSE	FALSE
5844	5544	15492	140000	FALSE	FALSE

5845	6450	15492	144900	FALSE	FALSE
5846	4840	15492	94900	FALSE	FALSE
5847	11250	15492	186500	FALSE	FALSE
5848	8449	15492	205000	FALSE	FALSE
5849	5400	15492	88000	FALSE	FALSE
5850	3000	15492	128000	FALSE	FALSE
5851	5085	15492	150000	FALSE	FALSE
5852	8910	15492	145000	FALSE	FALSE
5853	6222	15492	136000	FALSE	FALSE
5854	4800	15492	115000	FALSE	FALSE
5855	11758	15492	90000	FALSE	FALSE
5856	4800	15492	132500	FALSE	FALSE
5857	7800	15492	168000	FALSE	FALSE
5858	9600	15492	180000	FALSE	FALSE
5859	9000	15492	162000	FALSE	FALSE
5860	7626	15492	174000	FALSE	FALSE
5861	9375	15492	160000	FALSE	FALSE
5862	12350	15492	140000	FALSE	FALSE
5863	6000	15492	112900	FALSE	FALSE
5864	6100	15492	96000	FALSE	FALSE
5866	4560	15492	156500	FALSE	FALSE
5867	4720	15492	111500	FALSE	FALSE
5868	5760	15492	122000	FALSE	FALSE
5869	6400	15492	160500	FALSE	FALSE
5870	5240	15492	155000	FALSE	FALSE
5871	5332	15492	128500	FALSE	FALSE
5872	5060	15492	134900	FALSE	FALSE
5873	3750	15492	114000	FALSE	FALSE
5874	5180	15492	55000	FALSE	FALSE
5875	3660	15492	30100	FALSE	FALSE
5876	3134	15492	200000	FALSE	FALSE
5877	5600	15492	97500	FALSE	FALSE
5878	3750	15492	169900	FALSE	FALSE
5879	3600	15492	148000	FALSE	FALSE
5881	4758	15492	79900	FALSE	FALSE
5882	3660	15492	87500	FALSE	FALSE
5883	3600	15492	155000	FALSE	FALSE
5884	4000	15492	105000	FALSE	FALSE
5885	2430	15492	195000	FALSE	FALSE
5886	2702	15492	204900	FALSE	FALSE
5887	3600	15492	152000	FALSE	FALSE
5888	3660	15492	27500	FALSE	FALSE
5889	4720	15492	41000	FALSE	FALSE

5890	6440	15492	60000	FALSE	FALSE
5891	6700	15492	199000	FALSE	FALSE
5892	6157	15492	75000	FALSE	FALSE
5893	5040	15522	87500	FALSE	FALSE
5894	5632	15522	35000	FALSE	FALSE
5895	21780	15522	220000	FALSE	FALSE
5896	6477	15522	81000	FALSE	FALSE
5897	22000	15522	146483	FALSE	FALSE
5898	12555	15522	69800	FALSE	FALSE
5899	5125	15522	93500	FALSE	FALSE
5900	5830	15522	105000	FALSE	FALSE
5901	5000	15522	67900	FALSE	FALSE
5902	4800	15522	50000	FALSE	FALSE
5903	3600	15522	210000	TRUE	TRUE
5904	6960	15522	635000	TRUE	TRUE
5905	4800	15522	353000	TRUE	TRUE
5906	4080	15522	265000	TRUE	FALSE
5907	5490	15522	315000	TRUE	FALSE
5908	4800	15522	240000	TRUE	FALSE
5909	4800	15522	329800	TRUE	TRUE
5910	4800	15522	239000	TRUE	TRUE
5911	3600	15522	239000	TRUE	TRUE
5912	3600	15522	169000	TRUE	TRUE
5913	8475	15522	240000	FALSE	FALSE
5914	7320	15522	156000	FALSE	FALSE
5915	6000	15522	74000	FALSE	FALSE
5916	5760	15522	136500	FALSE	FALSE
5917	5000	15522	143000	FALSE	FALSE
5918	4800	15522	82000	FALSE	FALSE
5919	5000	15522	129900	FALSE	FALSE
5920	5120	15522	106500	FALSE	FALSE
5921	6000	15522	76000	FALSE	FALSE
5922	21780	15522	110000	FALSE	FALSE
5923	4800	15522	92000	FALSE	FALSE
5924	20000	15522	183000	FALSE	FALSE
5925	5080	15522	100000	FALSE	FALSE
5926	17200	15522	154000	FALSE	FALSE
5927	13696	15522	149500	FALSE	FALSE
5928	6095	15522	116000	FALSE	FALSE
5929	5280	15522	112000	FALSE	FALSE
5930	5871	15522	118000	FALSE	FALSE
5931	8250	15522	108000	FALSE	FALSE
5932	6000	15522	80000	FALSE	FALSE

5933	4800	15522	72000	FALSE	FALSE
5934	6000	15522	116500	FALSE	FALSE
5935	6000	15522	124500	FALSE	FALSE
5937	5520	15522	69900	FALSE	FALSE
5938	8570	15522	97000	FALSE	FALSE
5939	5200	15522	69000	FALSE	FALSE
5940	5000	15522	25000	FALSE	FALSE
5941	3600	15522	195000	FALSE	TRUE
5942	4440	15522	25000	FALSE	FALSE
5943	4800	15522	265000	FALSE	FALSE
5944	7500	15522	215000	FALSE	TRUE
5945	5100	15522	104000	FALSE	FALSE
5946	6247	15522	84000	FALSE	FALSE
5947	5040	15522	130000	FALSE	FALSE
5948	5100	15522	95000	FALSE	FALSE
5949	6528	15522	40000	FALSE	FALSE
5950	4800	15522	78000	FALSE	FALSE
5951	3450	15522	65000	FALSE	FALSE
5952	3582	15522	84500	FALSE	FALSE
5953	4200	15522	69900	FALSE	FALSE
5954	5250	15522	53000	FALSE	FALSE
5955	12000	15522	265000	FALSE	FALSE
5956	9710	15522	207750	FALSE	FALSE
5957	9184	15522	115000	FALSE	FALSE
5958	9375	15522	126900	FALSE	FALSE
5959	8880	15522	90000	FALSE	FALSE
5960	8344	15522	83000	FALSE	FALSE
5961	10004	15522	133500	FALSE	FALSE
5962	7200	15522	150000	FALSE	FALSE
5963	4949	15522	214900	FALSE	FALSE
5964	5400	15522	137000	FALSE	FALSE
5965	4641	15522	50200	FALSE	FALSE
5966	5400	15522	62500	FALSE	FALSE
5967	3960	15522	79000	FALSE	FALSE
5968	4937	15522	175400	FALSE	FALSE
5969	4691	15522	97000	FALSE	FALSE
5970	5760	15522	74900	FALSE	FALSE
5971	5936	15522	172500	FALSE	FALSE
5972	4640	15522	88900	FALSE	FALSE
5973	4284	15522	207000	FALSE	TRUE
5974	4800	15522	135000	FALSE	FALSE
5975	5200	15522	102000	FALSE	FALSE
5976	5720	15522	86000	FALSE	FALSE

5977	3600	15522	19100	FALSE	FALSE
5978	4800	15522	175000	FALSE	FALSE
5979	5400	15522	172000	FALSE	FALSE
5980	4760	15522	162500	FALSE	FALSE
5981	4171	15522	69900	FALSE	FALSE
5982	4800	15522	107000	FALSE	FALSE
5983	6020	15522	176900	FALSE	FALSE
5984	4480	15522	84000	FALSE	FALSE
5985	5450	15522	165500	FALSE	FALSE
5986	4046	15522	39000	FALSE	FALSE
5987	6615	15522	104900	FALSE	FALSE
5988	6360	15522	153000	FALSE	FALSE
5989	5360	15522	119900	FALSE	FALSE
5990	4080	15522	124900	FALSE	FALSE
5991	4880	15522	136500	FALSE	FALSE
5992	5400	15522	70700	FALSE	FALSE
5993	4800	15522	163000	FALSE	FALSE
5994	3600	15522	93200	FALSE	FALSE
5995	10547	15522	162800	FALSE	TRUE
5996	6000	15522	170900	FALSE	FALSE
5997	11390	15522	175000	FALSE	FALSE
5998	5280	15522	72000	FALSE	FALSE
5999	4838	15522	144500	FALSE	FALSE
6000	9546	15522	148000	FALSE	FALSE
6001	8300	15522	90000	FALSE	FALSE
6002	7800	15522	200000	FALSE	FALSE
6003	8250	15522	133500	FALSE	FALSE
6004	6650	15522	149900	FALSE	FALSE
6005	5700	15522	156800	FALSE	FALSE
6006	6600	15522	149000	FALSE	FALSE
6007	6840	15522	115000	FALSE	FALSE
6008	6720	15522	133500	FALSE	FALSE
6009	6240	15522	103500	FALSE	FALSE
6010	7440	15522	145000	FALSE	FALSE
6011	7371	15522	115200	FALSE	FALSE
6012	8555	15522	100000	FALSE	FALSE
6013	8442	15522	87700	FALSE	FALSE
6014	6700	15522	90600	FALSE	FALSE
6015	3600	15522	44000	FALSE	FALSE
6016	3750	15522	32000	FALSE	FALSE
6017	4375	15522	80000	FALSE	FALSE
6018	4250	15522	72000	FALSE	FALSE
6019	5408	15522	162000	FALSE	FALSE

6020	11970	15522	210000	FALSE	FALSE
6021	6450	15522	139900	FALSE	FALSE
6022	5040	15522	164900	FALSE	FALSE
6023	5289	15522	118450	FALSE	FALSE
6024	5805	15522	136000	FALSE	FALSE
6025	6450	15522	95000	FALSE	FALSE
6026	4800	15522	83800	FALSE	FALSE
6027	4956	15522	98000	FALSE	FALSE
6028	6550	15522	89900	FALSE	FALSE
6029	6161	15522	154000	FALSE	FALSE
6030	6519	15522	135000	FALSE	FALSE
6031	4708	15522	110000	FALSE	FALSE
6032	5000	15522	165000	FALSE	FALSE
6033	3600	15522	110000	FALSE	FALSE
6034	20640	15522	208750	FALSE	FALSE
6035	6450	15522	128000	FALSE	FALSE
6036	6600	15522	105000	FALSE	FALSE
6037	3000	15522	34900	FALSE	FALSE
6038	4800	15522	149000	FALSE	FALSE
6040	2250	15522	450000	FALSE	FALSE
6041	3630	15522	161000	FALSE	FALSE
6042	3300	15522	45000	FALSE	FALSE
6043	3600	15522	184500	FALSE	FALSE
6044	3920	15522	112000	FALSE	FALSE
6045	4350	15522	133500	FALSE	FALSE
6046	7200	15522	251500	FALSE	TRUE
6047	5400	15522	249900	FALSE	TRUE
6048	3570	15522	215000	FALSE	FALSE
6049	10130	15522	201500	FALSE	FALSE
6050	4836	15522	116000	FALSE	FALSE
6051	4838	15522	119900	FALSE	FALSE
6052	3150	15522	155000	FALSE	FALSE
6053	6588	15522	128000	FALSE	FALSE
6054	5358	15522	154000	FALSE	FALSE
6055	4590	15522	232000	FALSE	FALSE
6056	3600	15522	38000	FALSE	FALSE
6057	2400	15522	30000	FALSE	FALSE
6058	10800	15522	25000	FALSE	FALSE
6059	5289	15553	70000	FALSE	FALSE
6060	4200	15553	80000	FALSE	FALSE
6061	8100	15553	74000	FALSE	FALSE
6062	12998	15553	118500	FALSE	FALSE
6063	5200	15553	81000	FALSE	FALSE

6064	5280	15553	106000	FALSE	FALSE
6065	7500	15553	123100	FALSE	FALSE
6066	7920	15553	30000	FALSE	FALSE
6067	13696	15553	100000	FALSE	FALSE
6068	5782	15553	97000	FALSE	FALSE
6069	7740	15553	81000	FALSE	FALSE
6070	4800	15553	40000	FALSE	FALSE
6071	4800	15553	40000	FALSE	FALSE
6072	6000	15553	50000	FALSE	FALSE
6073	12895	15553	108000	FALSE	FALSE
6074	5400	15553	158000	TRUE	FALSE
6075	2775	15553	105000	TRUE	FALSE
6076	2217	15553	112000	TRUE	TRUE
6077	4140	15553	222000	TRUE	TRUE
6078	4800	15553	217000	TRUE	TRUE
6079	3600	15553	126000	TRUE	FALSE
6080	3600	15553	64100	TRUE	FALSE
6081	9000	15553	540000	TRUE	FALSE
6082	4800	15553	295900	TRUE	TRUE
6083	7200	15553	475000	TRUE	TRUE
6084	6100	15553	384000	TRUE	TRUE
6085	4880	15553	369000	TRUE	TRUE
6086	2400	15553	193000	TRUE	TRUE
6087	12708	15553	45000	FALSE	FALSE
6088	6750	15553	152000	FALSE	FALSE
6089	2352	15553	92000	FALSE	FALSE
6090	5000	15553	161500	FALSE	FALSE
6091	7750	15553	110000	FALSE	FALSE
6092	4800	15553	127500	FALSE	FALSE
6094	5000	15553	138500	FALSE	FALSE
6095	4800	15553	88500	FALSE	FALSE
6096	5120	15553	97000	FALSE	FALSE
6097	4800	15553	107000	FALSE	FALSE
6098	5000	15553	131500	FALSE	FALSE
6099	5060	15553	98500	FALSE	FALSE
6100	7200	15553	196000	FALSE	FALSE
6101	4800	15553	154000	FALSE	FALSE
6102	5000	15553	123700	FALSE	FALSE
6103	10230	15553	120000	FALSE	FALSE
6104	7200	15553	110000	FALSE	FALSE
6105	7800	15553	152000	FALSE	FALSE
6106	12000	15553	144000	FALSE	FALSE
6107	5805	15553	99500	FALSE	FALSE

6108	5850	15553	92000	FALSE	FALSE
6109	4800	15553	87000	FALSE	FALSE
6110	5040	15553	89900	FALSE	FALSE
6111	8400	15553	95000	FALSE	FALSE
6112	7380	15553	95000	FALSE	FALSE
6113	7192	15553	118000	FALSE	FALSE
6114	7200	15553	107000	FALSE	FALSE
6115	6250	15553	136500	FALSE	FALSE
6116	6480	15553	141000	FALSE	TRUE
6117	4687	15553	84500	FALSE	FALSE
6118	6301	15553	255000	FALSE	FALSE
6119	3900	15553	76900	FALSE	FALSE
6120	5000	15553	60000	FALSE	FALSE
6121	3690	15553	159000	FALSE	FALSE
6122	5160	15553	105000	FALSE	FALSE
6123	4960	15553	30000	FALSE	FALSE
6124	4800	15553	94000	FALSE	FALSE
6125	6160	15553	65000	FALSE	FALSE
6126	6105	15553	112000	FALSE	FALSE
6127	4515	15553	66000	FALSE	FALSE
6128	5320	15553	54000	FALSE	FALSE
6129	10000	15553	109900	FALSE	FALSE
6130	8750	15553	106000	FALSE	FALSE
6131	5002	15553	161500	FALSE	FALSE
6132	4800	15553	131900	FALSE	FALSE
6133	7443	15553	113500	FALSE	FALSE
6134	5280	15553	82000	FALSE	FALSE
6135	5950	15553	146000	FALSE	FALSE
6136	5207	15553	191555	FALSE	FALSE
6137	3840	15553	35000	FALSE	FALSE
6138	4522	15553	55000	FALSE	FALSE
6139	4800	15553	65900	FALSE	FALSE
6140	5715	15553	152000	FALSE	FALSE
6141	4720	15553	152000	FALSE	FALSE
6142	3680	15553	167000	FALSE	FALSE
6143	6250	15553	142900	FALSE	FALSE
6144	3600	15553	28000	FALSE	FALSE
6145	7150	15553	67500	FALSE	FALSE
6146	4600	15553	92000	FALSE	FALSE
6147	3900	15553	63000	FALSE	FALSE
6148	5400	15553	155000	FALSE	FALSE
6149	5400	15553	149000	FALSE	FALSE
6150	3844	15553	75000	FALSE	FALSE

6151	21000	15553	86000	FALSE	FALSE
6152	5400	15553	90000	FALSE	FALSE
6153	7200	15553	169900	FALSE	FALSE
6155	12035	15553	150000	FALSE	FALSE
6156	7320	15553	117500	FALSE	FALSE
6157	7965	15553	135000	FALSE	FALSE
6158	8100	15553	137000	FALSE	FALSE
6159	7600	15553	129900	FALSE	FALSE
6160	7383	15553	128000	FALSE	FALSE
6161	5250	15553	125000	FALSE	FALSE
6162	5760	15553	70000	FALSE	FALSE
6163	4140	15553	65000	FALSE	FALSE
6164	7700	15553	215000	FALSE	FALSE
6165	7200	15553	148000	FALSE	FALSE
6166	4800	15553	110000	FALSE	FALSE
6167	6390	15553	118300	FALSE	FALSE
6168	8050	15553	120000	FALSE	FALSE
6169	5375	15553	90500	FALSE	FALSE
6170	5940	15553	126850	FALSE	FALSE
6171	7500	15553	160000	FALSE	FALSE
6172	7150	15553	143000	FALSE	FALSE
6173	4800	15553	107500	FALSE	FALSE
6174	8532	15553	88500	FALSE	FALSE
6175	12120	15553	212500	FALSE	FALSE
6176	9450	15553	215000	FALSE	FALSE
6177	8520	15553	196750	FALSE	FALSE
6179	8300	15553	107000	FALSE	FALSE
6180	6450	15553	156000	FALSE	FALSE
6181	6840	15553	110000	FALSE	FALSE
6182	6840	15553	149900	FALSE	FALSE
6183	7080	15553	126500	FALSE	FALSE
6184	6000	15553	145000	FALSE	FALSE
6185	6240	15553	130000	FALSE	FALSE
6186	7770	15553	114500	FALSE	FALSE
6187	5350	15553	97000	FALSE	FALSE
6188	6500	15553	141000	FALSE	FALSE
6189	6020	15553	139900	FALSE	FALSE
6190	3720	15553	98475	FALSE	FALSE
6191	4500	15553	106000	FALSE	FALSE
6192	9540	15553	154000	FALSE	FALSE
6193	6280	15553	169900	FALSE	FALSE
6194	4800	15553	132500	FALSE	FALSE
6195	3600	15553	100000	FALSE	FALSE

6196	4515	15553	250000	FALSE	FALSE
6197	3360	15553	47500	FALSE	FALSE
6198	5616	15553	180000	FALSE	TRUE
6199	4250	15553	166000	FALSE	TRUE
6200	3540	15553	165000	FALSE	FALSE
6201	4800	15553	149500	FALSE	FALSE
6202	5760	15553	160000	FALSE	FALSE
6204	5000	15553	20000	FALSE	FALSE
6205	3150	15553	28800	FALSE	FALSE
6206	5125	15584	64500	FALSE	FALSE
6207	6250	15584	74900	FALSE	FALSE
6208	4920	15584	93900	FALSE	FALSE
6209	5040	15584	85000	FALSE	FALSE
6210	5880	15584	93000	FALSE	FALSE
6211	6600	15584	69000	FALSE	FALSE
6212	6000	15584	415000	TRUE	FALSE
6213	3690	15584	125000	TRUE	TRUE
6214	4838	15584	155000	TRUE	FALSE
6215	6400	15584	100000	TRUE	FALSE
6216	5080	15584	175000	TRUE	TRUE
6217	9600	15584	867000	TRUE	FALSE
6218	11400	15584	975000	TRUE	TRUE
6219	10800	15584	545000	TRUE	TRUE
6220	7200	15584	530000	TRUE	TRUE
6221	4880	15584	355000	TRUE	TRUE
6222	3810	15584	208000	TRUE	FALSE
6223	3600	15584	261500	TRUE	TRUE
6224	3600	15584	158500	TRUE	FALSE
6225	6000	15584	260000	TRUE	TRUE
6226	2600	15584	15000	FALSE	FALSE
6227	3600	15584	282500	FALSE	TRUE
6228	5360	15584	147425	FALSE	FALSE
6229	4800	15584	131000	FALSE	FALSE
6230	7000	15584	80000	FALSE	FALSE
6231	6000	15584	81200	FALSE	FALSE
6232	12118	15584	276000	FALSE	FALSE
6233	9350	15584	145000	FALSE	FALSE
6234	6250	15584	183000	FALSE	TRUE
6235	6708	15584	104900	FALSE	FALSE
6236	4800	15584	85000	FALSE	FALSE
6237	6250	15584	136000	FALSE	FALSE
6238	6890	15584	81500	FALSE	FALSE
6239	10125	15584	144000	FALSE	FALSE

6240	5550	15584	120000	FALSE	FALSE
6241	5248	15584	69000	FALSE	FALSE
6242	5160	15584	55225	FALSE	FALSE
6243	5160	15584	72400	FALSE	FALSE
6244	3750	15584	26000	FALSE	FALSE
6245	5280	15584	60000	FALSE	FALSE
6246	4773	15584	105000	FALSE	FALSE
6247	4720	15584	95000	FALSE	FALSE
6248	4335	15584	121000	FALSE	FALSE
6249	3750	15584	85000	FALSE	FALSE
6250	17984	15584	138500	FALSE	FALSE
6251	11600	15584	159000	FALSE	FALSE
6252	8954	15584	129900	FALSE	FALSE
6253	10309	15584	141000	FALSE	FALSE
6254	9000	15584	101000	FALSE	FALSE
6255	9094	15584	94900	FALSE	FALSE
6256	8400	15584	107000	FALSE	FALSE
6257	7208	15584	97700	FALSE	FALSE
6258	7320	15584	99000	FALSE	FALSE
6259	9800	15584	135000	FALSE	FALSE
6261	5500	15584	149000	FALSE	FALSE
6262	4320	15584	164000	FALSE	FALSE
6263	6375	15584	205000	FALSE	FALSE
6264	5225	15584	188000	FALSE	FALSE
6265	5040	15584	125000	FALSE	FALSE
6266	5200	15584	150000	FALSE	FALSE
6267	3840	15584	85000	FALSE	FALSE
6268	5895	15584	160000	FALSE	TRUE
6269	3880	15584	93000	FALSE	FALSE
6270	3570	15584	100000	FALSE	FALSE
6271	5400	15584	90000	FALSE	FALSE
6272	4340	15584	99900	FALSE	FALSE
6273	4200	15584	182500	FALSE	FALSE
6274	5625	15584	110000	FALSE	FALSE
6275	8450	15584	98500	FALSE	FALSE
6276	8976	15584	185000	FALSE	FALSE
6277	5412	15584	95000	FALSE	FALSE
6278	6000	15584	135500	FALSE	FALSE
6279	7250	15584	118500	FALSE	FALSE
6280	5700	15584	270000	FALSE	FALSE
6281	10188	15584	130000	FALSE	FALSE
6282	6200	15584	142000	FALSE	FALSE
6283	4998	15584	114900	FALSE	FALSE

6284	13600	15584	152900	FALSE	FALSE
6285	4960	15584	106000	FALSE	FALSE
6286	6120	15584	136900	FALSE	FALSE
6287	9940	15584	141500	FALSE	FALSE
6288	6681	15584	171000	FALSE	FALSE
6289	8700	15584	133500	FALSE	FALSE
6290	10008	15584	155000	FALSE	FALSE
6291	7316	15584	115000	FALSE	FALSE
6292	6384	15584	147600	FALSE	FALSE
6293	7200	15584	125000	FALSE	FALSE
6294	7320	15584	84000	FALSE	FALSE
6295	5550	15584	114500	FALSE	FALSE
6296	6273	15584	129400	FALSE	FALSE
6297	5292	15584	59900	FALSE	FALSE
6298	4838	15584	110000	FALSE	FALSE
6299	4920	15584	90000	FALSE	FALSE
6300	3600	15584	89000	FALSE	FALSE
6301	7980	15584	90000	FALSE	FALSE
6302	5760	15584	66500	FALSE	FALSE
6303	8280	15584	75000	FALSE	FALSE
6304	16764	15584	200000	FALSE	TRUE
6305	1375	15584	14000	FALSE	FALSE
6306	4200	15584	58000	FALSE	FALSE
6307	2550	15584	73800	FALSE	FALSE
6308	2500	15584	29000	FALSE	FALSE
6309	4760	15584	130000	FALSE	FALSE
6310	21000	15584	169900	FALSE	FALSE
6311	4884	15584	117000	FALSE	FALSE
6312	6000	15584	115500	FALSE	FALSE
6313	4800	15584	135900	FALSE	FALSE
6314	6150	15584	116500	FALSE	FALSE
6315	21133	15584	242000	FALSE	FALSE
6316	11528	15584	175000	FALSE	FALSE
6317	6000	15584	127000	FALSE	FALSE
6318	8400	15584	115000	FALSE	FALSE
6319	9337	15584	144500	FALSE	FALSE
6320	7380	15584	139900	FALSE	FALSE
6321	5360	15584	106000	FALSE	FALSE
6322	4600	15584	50000	FALSE	FALSE
6323	8460	15584	132900	FALSE	FALSE
6324	6674	15584	139900	FALSE	FALSE
6325	4800	15584	140000	FALSE	FALSE
6326	7650	15584	175000	FALSE	FALSE

6327	4880	15584	85100	FALSE	FALSE
6328	6600	15584	157000	FALSE	FALSE
6329	4880	15584	144000	FALSE	FALSE
6330	4880	15584	129500	FALSE	FALSE
6331	3904	15584	106000	FALSE	FALSE
6332	8040	15584	116500	FALSE	FALSE
6333	3528	15584	105000	FALSE	FALSE
6334	3293	15584	135000	FALSE	FALSE
6335	4110	15584	168000	FALSE	FALSE
6336	4020	15584	155000	FALSE	FALSE
6337	7800	15614	127500	FALSE	FALSE
6338	9000	15614	48000	FALSE	FALSE
6339	5080	15614	55225	FALSE	FALSE
6340	5460	15614	54500	FALSE	FALSE
6341	4960	15614	13124	FALSE	FALSE
6342	16984	15614	69200	FALSE	FALSE
6343	6400	15614	39500	FALSE	FALSE
6344	8700	15614	84000	FALSE	FALSE
6345	5400	15614	86000	FALSE	FALSE
6346	4920	15614	67900	FALSE	FALSE
6347	4800	15614	54000	FALSE	FALSE
6348	5000	15614	72000	FALSE	FALSE
6349	7200	15614	80000	FALSE	FALSE
6350	7280	15614	45000	FALSE	FALSE
6351	3840	15614	200000	TRUE	TRUE
6352	4800	15614	161750	TRUE	TRUE
6353	3600	15614	149000	TRUE	TRUE
6354	4560	15614	139000	TRUE	FALSE
6355	20790	15614	955000	TRUE	TRUE
6356	4125	15614	259000	TRUE	TRUE
6357	3690	15614	133500	TRUE	FALSE
6358	6250	15614	132000	FALSE	FALSE
6359	5000	15614	134900	FALSE	FALSE
6360	4950	15614	149500	FALSE	FALSE
6361	5400	15614	87000	FALSE	FALSE
6362	5080	15614	93000	FALSE	FALSE
6363	5000	15614	115000	FALSE	FALSE
6364	4800	15614	92500	FALSE	FALSE
6365	5080	15614	94000	FALSE	FALSE
6366	5400	15614	105000	FALSE	FALSE
6367	5684	15614	89000	FALSE	FALSE
6368	5280	15614	69900	FALSE	FALSE
6369	5060	15614	99000	FALSE	FALSE

6370	5900	15614	170000	FALSE	FALSE
6371	5610	15614	115000	FALSE	FALSE
6372	8640	15614	137500	FALSE	FALSE
6373	21780	15614	186500	FALSE	FALSE
6374	5978	15614	115000	FALSE	FALSE
6375	5120	15614	121750	FALSE	FALSE
6376	5080	15614	111000	FALSE	FALSE
6377	9664	15614	82400	FALSE	FALSE
6378	5040	15614	93000	FALSE	FALSE
6379	8400	15614	80000	FALSE	FALSE
6380	7315	15614	99000	FALSE	FALSE
6381	4800	15614	102000	FALSE	FALSE
6382	7000	15614	92000	FALSE	FALSE
6383	5346	15614	95000	FALSE	FALSE
6384	4064	15614	263000	FALSE	TRUE
6385	5950	15614	72000	FALSE	FALSE
6386	5125	15614	94800	FALSE	FALSE
6387	5250	15614	87000	FALSE	FALSE
6388	5376	15614	102800	FALSE	FALSE
6389	4800	15614	35000	FALSE	FALSE
6390	4800	15614	60000	FALSE	FALSE
6391	4600	15614	47000	FALSE	FALSE
6392	4760	15614	103000	FALSE	FALSE
6393	4148	15614	95000	FALSE	FALSE
6394	4752	15614	85000	FALSE	FALSE
6395	6300	15614	94000	FALSE	FALSE
6396	7490	15614	127000	FALSE	FALSE
6397	6160	15614	173700	FALSE	FALSE
6398	5760	15614	165850	FALSE	FALSE
6399	5400	15614	142500	FALSE	FALSE
6400	5760	15614	171500	FALSE	TRUE
6401	7050	15614	123500	FALSE	FALSE
6402	4715	15614	96500	FALSE	FALSE
6403	5040	15614	64900	FALSE	FALSE
6404	5088	15614	199000	FALSE	FALSE
6405	4800	15614	125000	FALSE	FALSE
6406	7708	15614	140000	FALSE	FALSE
6407	3600	15614	116500	FALSE	FALSE
6408	5117	15614	200000	FALSE	TRUE
6409	5000	15614	182000	FALSE	FALSE
6410	4480	15614	100000	FALSE	FALSE
6411	4200	15614	159900	FALSE	FALSE
6412	4522	15614	150000	FALSE	FALSE

6413	4800	15614	88000	FALSE	FALSE
6414	3600	15614	68000	FALSE	FALSE
6415	6210	15614	235000	FALSE	FALSE
6416	9672	15614	169000	FALSE	FALSE
6417	6500	15614	123000	FALSE	FALSE
6418	4880	15614	129000	FALSE	FALSE
6419	4920	15614	125000	FALSE	FALSE
6420	7946	15614	140000	FALSE	FALSE
6421	6048	15614	142600	FALSE	FALSE
6422	4836	15614	139500	FALSE	FALSE
6423	7524	15614	135000	FALSE	FALSE
6424	7040	15614	126000	FALSE	FALSE
6425	6700	15614	152000	FALSE	FALSE
6426	5628	15614	109900	FALSE	FALSE
6427	7744	15614	149900	FALSE	FALSE
6428	6500	15614	157500	FALSE	FALSE
6429	6758	15614	114900	FALSE	FALSE
6430	8352	15614	125000	FALSE	FALSE
6431	6700	15614	122900	FALSE	FALSE
6432	8520	15614	71800	FALSE	FALSE
6433	6800	15614	121500	FALSE	FALSE
6434	6600	15614	82000	FALSE	FALSE
6435	6270	15614	114700	FALSE	FALSE
6436	7200	15614	200000	FALSE	FALSE
6437	5400	15614	142000	FALSE	FALSE
6438	5336	15614	92500	FALSE	FALSE
6439	8970	15614	242000	FALSE	FALSE
6440	7320	15614	162500	FALSE	FALSE
6441	9100	15614	217500	FALSE	FALSE
6442	12127	15614	180000	FALSE	FALSE
6443	7200	15614	184000	FALSE	FALSE
6444	20086	15614	150000	FALSE	FALSE
6445	8220	15614	141900	FALSE	FALSE
6446	5588	15614	125000	FALSE	FALSE
6447	7260	15614	104000	FALSE	FALSE
6448	7124	15614	126500	FALSE	FALSE
6449	7290	15614	97500	FALSE	FALSE
6450	5166	15614	107500	FALSE	FALSE
6451	7564	15614	121000	FALSE	FALSE
6452	3440	15614	166000	FALSE	TRUE
6453	4290	15614	194000	FALSE	TRUE
6454	6804	15614	240000	FALSE	FALSE
6455	4800	15614	200000	FALSE	TRUE

6456	5600	15614	75000	FALSE	FALSE
6457	4800	15614	153500	FALSE	FALSE
6458	5760	15614	86000	FALSE	FALSE
6459	4148	15614	92000	FALSE	FALSE
6460	7320	15614	105500	FALSE	FALSE
6461	5720	15614	149900	FALSE	FALSE
6462	4290	15614	187800	FALSE	FALSE
6463	5265	15614	184000	FALSE	FALSE
6464	3750	15614	107500	FALSE	FALSE
6465	6160	15614	101000	FALSE	FALSE
6466	3540	15614	112000	FALSE	FALSE
6467	5950	15614	113000	FALSE	FALSE
6468	5289	15645	50000	FALSE	FALSE
6469	3600	15645	32000	FALSE	FALSE
6470	5280	15645	85000	FALSE	FALSE
6471	4840	15645	36000	FALSE	FALSE
6472	4920	15645	45900	FALSE	FALSE
6473	6656	15645	76900	FALSE	FALSE
6474	5330	15645	116500	FALSE	FALSE
6475	6350	15645	55500	FALSE	FALSE
6476	3600	15645	129000	TRUE	TRUE
6477	7200	15645	629000	TRUE	TRUE
6478	3144	15645	300000	TRUE	FALSE
6479	4683	15645	182000	TRUE	TRUE
6481	2500	15645	285000	TRUE	TRUE
6482	6600	15645	352000	TRUE	FALSE
6483	3390	15645	37500	FALSE	FALSE
6484	4050	15645	61000	FALSE	FALSE
6485	14280	15645	270000	FALSE	FALSE
6486	5400	15645	163000	FALSE	FALSE
6487	9720	15645	170000	FALSE	FALSE
6488	6150	15645	161500	FALSE	FALSE
6489	8580	15645	119000	FALSE	FALSE
6490	5040	15645	147000	FALSE	FALSE
6491	7200	15645	155000	FALSE	FALSE
6492	5040	15645	130000	FALSE	FALSE
6493	7616	15645	172400	FALSE	FALSE
6494	4920	15645	96900	FALSE	FALSE
6495	6580	15645	125900	FALSE	FALSE
6496	7200	15645	135000	FALSE	FALSE
6497	4800	15645	60000	FALSE	FALSE
6498	4800	15645	125000	FALSE	FALSE
6499	6300	15645	96000	FALSE	FALSE

6500	6016	15645	140000	FALSE	FALSE
6501	6120	15645	106500	FALSE	FALSE
6502	7808	15645	82900	FALSE	FALSE
6503	5360	15645	92000	FALSE	FALSE
6504	6000	15645	124000	FALSE	FALSE
6505	1850	15645	99000	FALSE	FALSE
6506	3455	15645	32500	FALSE	FALSE
6507	7550	15645	80000	FALSE	FALSE
6508	5928	15645	17757	FALSE	FALSE
6509	6171	15645	44200	FALSE	FALSE
6510	4800	15645	109900	FALSE	FALSE
6511	5040	15645	37500	FALSE	FALSE
6512	6160	15645	21000	FALSE	FALSE
6513	3960	15645	95000	FALSE	FALSE
6514	3540	15645	90000	FALSE	FALSE
6515	7200	15645	115000	FALSE	FALSE
6516	14772	15645	187000	FALSE	FALSE
6517	10381	15645	119500	FALSE	FALSE
6518	7200	15645	112000	FALSE	FALSE
6519	6000	15645	114000	FALSE	FALSE
6520	7200	15645	107000	FALSE	FALSE
6521	6095	15645	67000	FALSE	FALSE
6522	6250	15645	188500	FALSE	FALSE
6523	6380	15645	253000	FALSE	TRUE
6524	6500	15645	112500	FALSE	FALSE
6525	6000	15645	167500	FALSE	TRUE
6526	6000	15645	145000	FALSE	FALSE
6527	3600	15645	125000	FALSE	FALSE
6528	6786	15645	99900	FALSE	FALSE
6529	4875	15645	30000	FALSE	FALSE
6530	4800	15645	180000	FALSE	FALSE
6531	4960	15645	94000	FALSE	FALSE
6532	7200	15645	189000	FALSE	FALSE
6533	8190	15645	147500	FALSE	FALSE
6534	6968	15645	124000	FALSE	FALSE
6535	7000	15645	118500	FALSE	FALSE
6536	8978	15645	127500	FALSE	FALSE
6537	7370	15645	88000	FALSE	FALSE
6538	8064	15645	47500	FALSE	FALSE
6539	6912	15645	115000	FALSE	FALSE
6540	5500	15645	86400	FALSE	FALSE
6541	7020	15645	132000	FALSE	FALSE
6542	6000	15645	131000	FALSE	FALSE

6543	16632	15645	258000	FALSE	FALSE
6544	7858	15645	150000	FALSE	FALSE
6545	4856	15645	137000	FALSE	FALSE
6546	4200	15645	102000	FALSE	FALSE
6547	10660	15645	176000	FALSE	FALSE
6548	14250	15645	190000	FALSE	FALSE
6549	20663	15645	210000	FALSE	FALSE
6550	8260	15645	129900	FALSE	FALSE
6551	8175	15645	133000	FALSE	FALSE
6552	8140	15645	160000	FALSE	FALSE
6553	9374	15645	119000	FALSE	FALSE
6554	8100	15645	157000	FALSE	FALSE
6555	4680	15645	177275	FALSE	TRUE
6556	3600	15645	169900	FALSE	FALSE
6557	4838	15645	164900	FALSE	FALSE
6558	3600	15645	25000	FALSE	FALSE
6559	5670	15645	112000	FALSE	FALSE
6560	4392	15645	135000	FALSE	FALSE
6561	7750	15645	301000	FALSE	FALSE
6562	3870	15645	146000	FALSE	FALSE
6563	3600	15645	192000	FALSE	FALSE
6564	4680	15645	235000	FALSE	FALSE
6565	4880	15645	129900	FALSE	FALSE
6566	7296	15645	199500	FALSE	FALSE
6567	3870	15645	184000	FALSE	FALSE
6568	3600	15645	175500	FALSE	FALSE
6569	2860	15645	154900	FALSE	FALSE
6571	5760	15645	165000	FALSE	FALSE
6572	3600	15675	58000	FALSE	FALSE
6573	6837	15675	38000	FALSE	FALSE
6574	3745	15675	51250	FALSE	FALSE
6575	6160	15675	42000	FALSE	FALSE
6576	4956	15675	50000	FALSE	FALSE
6577	6160	15675	87000	FALSE	FALSE
6578	7255	15675	90500	FALSE	FALSE
6579	7200	15675	502500	TRUE	TRUE
6580	1680	15675	159900	TRUE	TRUE
6581	2730	15675	131000	TRUE	TRUE
6582	3600	15675	127000	TRUE	TRUE
6583	4880	15675	330000	TRUE	FALSE
6584	5183	15675	230000	TRUE	FALSE
6585	2880	15675	219900	TRUE	TRUE
6586	4500	15675	210000	TRUE	TRUE

6587	3600	15675	245000	TRUE	TRUE
6588	3600	15675	365000	TRUE	TRUE
6589	7500	15675	505000	TRUE	TRUE
6590	9920	15675	180000	FALSE	FALSE
6591	6000	15675	89900	FALSE	FALSE
6592	5000	15675	116900	FALSE	FALSE
6593	6000	15675	112000	FALSE	FALSE
6594	4800	15675	105000	FALSE	FALSE
6595	5000	15675	119900	FALSE	FALSE
6596	4800	15675	62000	FALSE	FALSE
6597	5280	15675	65975	FALSE	FALSE
6598	5120	15675	82000	FALSE	FALSE
6599	11978	15675	195000	FALSE	FALSE
6600	6250	15675	182000	FALSE	FALSE
6601	21780	15675	133000	FALSE	FALSE
6602	8400	15675	77300	FALSE	FALSE
6603	8000	15675	111000	FALSE	FALSE
6604	4800	15675	101000	FALSE	FALSE
6605	6160	15675	116500	FALSE	FALSE
6606	6000	15675	78900	FALSE	FALSE
6607	7600	15675	80000	FALSE	FALSE
6608	6240	15675	81000	FALSE	FALSE
6609	7748	15675	175000	FALSE	FALSE
6610	1850	15675	9400	FALSE	FALSE
6611	4720	15675	41000	FALSE	FALSE
6612	5125	15675	98000	FALSE	FALSE
6613	5760	15675	100375	FALSE	FALSE
6614	3600	15675	77600	FALSE	FALSE
6615	3600	15675	87000	FALSE	FALSE
6616	2929	15675	42500	FALSE	FALSE
6617	6490	15675	62900	FALSE	FALSE
6618	7270	15675	120000	FALSE	FALSE
6619	5200	15675	137500	FALSE	FALSE
6620	4800	15675	93500	FALSE	FALSE
6621	3025	15675	60000	FALSE	FALSE
6622	4560	15675	64900	FALSE	FALSE
6623	5040	15675	86850	FALSE	FALSE
6624	5520	15675	140500	FALSE	FALSE
6625	5040	15675	127000	FALSE	FALSE
6626	9100	15675	140000	FALSE	FALSE
6627	6200	15675	157000	FALSE	FALSE
6628	7200	15675	132700	FALSE	FALSE
6629	5712	15675	146000	FALSE	FALSE

6630	8400	15675	206000	FALSE	FALSE
6631	4800	15675	78000	FALSE	FALSE
6632	7320	15675	147500	FALSE	FALSE
6633	7874	15675	96400	FALSE	FALSE
6634	4880	15675	117900	FALSE	FALSE
6635	7950	15675	108000	FALSE	FALSE
6636	3500	15675	65000	FALSE	FALSE
6637	3000	15675	62000	FALSE	FALSE
6638	2650	15675	190000	FALSE	FALSE
6639	5130	15675	126175	FALSE	FALSE
6640	6500	15675	165000	FALSE	FALSE
6641	6250	15675	145000	FALSE	FALSE
6642	9568	15675	180500	FALSE	TRUE
6643	4800	15675	116500	FALSE	FALSE
6644	10665	15675	172000	FALSE	FALSE
6645	3850	15675	90000	FALSE	FALSE
6646	8990	15675	128000	FALSE	FALSE
6647	7150	15675	190000	FALSE	FALSE
6648	4928	15675	151000	FALSE	FALSE
6649	5400	15675	141000	FALSE	FALSE
6650	4636	15675	106000	FALSE	FALSE
6651	6072	15675	147000	FALSE	FALSE
6652	4020	15675	150000	FALSE	FALSE
6653	3270	15675	396500	FALSE	TRUE
6654	3480	15675	135000	FALSE	FALSE
6655	6000	15675	305000	FALSE	TRUE
6656	4800	15675	92500	FALSE	FALSE
6657	4440	15675	96500	FALSE	FALSE
6658	4800	15675	260000	FALSE	FALSE
6659	3600	15675	70800	FALSE	FALSE
6660	5900	15675	140000	FALSE	FALSE
6661	5160	15675	150000	FALSE	FALSE
6662	4720	15675	32000	FALSE	FALSE
6663	9975	15675	26200	FALSE	FALSE
6664	5500	15706	154000	FALSE	FALSE
6665	4800	15706	10750	FALSE	FALSE
6666	5330	15706	95000	FALSE	FALSE
6667	8868	15706	118000	FALSE	FALSE
6668	10235	15706	115000	FALSE	FALSE
6669	6000	15706	60000	FALSE	FALSE
6670	7245	15706	65000	FALSE	FALSE
6671	7200	15706	48000	FALSE	FALSE
6672	7440	15706	462500	TRUE	FALSE

6673	3600	15706	198000	TRUE	FALSE
6674	3600	15706	115000	TRUE	TRUE
6675	16999	15706	347500	FALSE	FALSE
6676	5000	15706	188831	FALSE	FALSE
6677	5500	15706	147500	FALSE	FALSE
6678	5166	15706	108000	FALSE	FALSE
6679	5253	15706	90000	FALSE	FALSE
6680	5280	15706	100000	FALSE	FALSE
6681	7497	15706	78300	FALSE	FALSE
6682	5720	15706	75000	FALSE	FALSE
6683	5251	15706	83000	FALSE	FALSE
6684	6000	15706	144900	FALSE	FALSE
6685	5760	15706	127500	FALSE	FALSE
6686	2340	15706	24900	FALSE	FALSE
6687	4720	15706	99000	FALSE	FALSE
6688	4888	15706	173500	FALSE	FALSE
6689	3600	15706	60000	FALSE	FALSE
6690	9679	15706	249400	FALSE	FALSE
6691	10478	15706	123000	FALSE	FALSE
6692	9545	15706	108900	FALSE	FALSE
6693	8517	15706	135000	FALSE	FALSE
6694	5160	15706	90900	FALSE	FALSE
6695	6588	15706	120750	FALSE	FALSE
6696	8100	15706	370000	FALSE	TRUE
6697	4320	15706	168000	FALSE	FALSE
6698	4046	15706	79000	FALSE	FALSE
6699	5400	15706	110000	FALSE	FALSE
6700	14985	15706	106900	FALSE	FALSE
6701	6000	15706	218700	FALSE	FALSE
6702	2850	15706	110000	FALSE	FALSE
6703	3600	15706	91000	FALSE	FALSE
6704	4680	15706	105000	FALSE	FALSE
6705	11484	15706	90000	FALSE	FALSE
6706	5124	15706	55000	FALSE	FALSE
6707	16170	15706	66000	FALSE	FALSE
6708	21000	15706	149900	FALSE	FALSE
6709	8215	15706	114000	FALSE	FALSE
6710	5400	15706	94000	FALSE	FALSE
6711	4838	15706	143000	FALSE	FALSE
6712	8520	15706	197400	FALSE	FALSE
6713	8192	15706	153400	FALSE	FALSE
6714	6850	15706	90000	FALSE	FALSE
6715	6890	15706	133500	FALSE	FALSE

6716	7680	15706	80000	FALSE	FALSE
6717	8978	15706	117000	FALSE	FALSE
6718	5600	15706	118000	FALSE	FALSE
6719	5520	15706	104420	FALSE	FALSE
6720	5000	15706	50000	FALSE	FALSE
6721	11264	15706	168000	FALSE	FALSE
6722	4800	15706	168250	FALSE	FALSE
6723	4290	15706	12500	FALSE	FALSE
6724	3706	15706	77000	FALSE	FALSE
6725	4050	15706	28000	FALSE	FALSE
6727	6480	15706	152000	FALSE	FALSE
6728	5560	15706	185100	FALSE	FALSE
6729	5400	15706	99700	FALSE	FALSE
6730	8280	15706	137000	FALSE	FALSE
6731	5310	15706	96000	FALSE	FALSE
6732	4216	15706	75000	FALSE	FALSE
6733	4200	15706	47500	FALSE	FALSE
6734	5400	15706	190000	FALSE	FALSE
6735	3660	15706	51000	FALSE	FALSE
6736	3600	15706	220000	FALSE	FALSE
6737	4836	15706	130000	FALSE	FALSE
6738	5250	15706	149000	FALSE	FALSE
6739	4880	15706	107500	FALSE	FALSE
6740	6144	15706	119000	FALSE	FALSE
6741	7650	15706	230000	FALSE	FALSE
6742	3150	15706	141900	FALSE	FALSE
6743	3750	15706	106000	FALSE	FALSE
6744	1530	15706	95500	FALSE	FALSE
6745	3660	15706	95000	FALSE	FALSE
6746	3150	15706	25000	FALSE	FALSE
6747	4800	15737	57500	FALSE	FALSE
6748	5984	15737	44000	FALSE	FALSE
6749	5000	15737	55000	FALSE	FALSE
6750	6600	15737	62700	FALSE	FALSE
6751	5400	15737	94000	FALSE	FALSE
6752	4800	15737	121500	FALSE	FALSE
6753	5880	15737	99900	FALSE	FALSE
6754	5460	15737	81500	FALSE	FALSE
6755	4800	15737	104700	FALSE	FALSE
6756	7560	15737	15000	FALSE	FALSE
6757	9504	15737	105000	FALSE	FALSE
6758	4920	15737	254000	TRUE	TRUE
6759	6000	15737	204000	TRUE	FALSE

6760	2880	15737	175000	TRUE	FALSE
6761	7200	15737	495000	TRUE	FALSE
6762	10010	15737	535000	TRUE	TRUE
6763	4800	15737	32000	FALSE	FALSE
6764	24192	15737	175000	FALSE	FALSE
6765	5160	15737	137000	FALSE	FALSE
6766	5106	15737	118000	FALSE	FALSE
6767	5240	15737	123000	FALSE	FALSE
6768	5120	15737	99500	FALSE	FALSE
6769	5000	15737	72500	FALSE	FALSE
6770	4200	15737	131000	FALSE	FALSE
6771	8055	15737	136900	FALSE	FALSE
6772	11700	15737	138500	FALSE	FALSE
6773	8384	15737	80500	FALSE	FALSE
6774	9129	15737	109000	FALSE	FALSE
6775	4800	15737	112000	FALSE	FALSE
6776	5400	15737	105500	FALSE	FALSE
6777	7380	15737	89750	FALSE	FALSE
6778	6955	15737	93000	FALSE	FALSE
6779	7000	15737	59000	FALSE	FALSE
6780	5120	15737	118000	FALSE	FALSE
6781	6250	15737	99800	FALSE	FALSE
6782	34400	15737	126000	FALSE	FALSE
6783	8450	15737	125000	FALSE	FALSE
6784	4920	15737	35000	FALSE	FALSE
6785	3750	15737	15000	FALSE	FALSE
6786	5670	15737	24000	FALSE	FALSE
6788	5250	15737	99000	FALSE	FALSE
6789	4879	15737	94800	FALSE	FALSE
6790	4165	15737	65000	FALSE	FALSE
6791	3200	15737	70000	FALSE	FALSE
6792	7520	15737	175000	FALSE	FALSE
6793	2170	15737	25000	FALSE	FALSE
6794	2100	15737	20000	FALSE	FALSE
6795	5582	15737	82900	FALSE	FALSE
6796	3180	15737	85000	FALSE	FALSE
6797	11129	15737	181000	FALSE	FALSE
6798	7344	15737	128000	FALSE	FALSE
6799	5520	15737	104000	FALSE	FALSE
6800	4800	15737	109000	FALSE	FALSE
6801	6000	15737	185900	FALSE	FALSE
6802	4640	15737	96400	FALSE	FALSE
6803	3780	15737	65900	FALSE	FALSE

6804	4403	15737	83000	FALSE	FALSE
6806	5640	15737	164500	FALSE	FALSE
6807	4797	15737	122900	FALSE	FALSE
6808	5593	15737	118000	FALSE	FALSE
6809	4800	15737	90000	FALSE	FALSE
6810	6000	15737	101500	FALSE	FALSE
6811	19889	15737	260000	FALSE	FALSE
6812	4375	15737	84000	FALSE	FALSE
6813	6240	15737	150000	FALSE	FALSE
6814	7320	15737	122250	FALSE	FALSE
6815	5580	15737	135000	FALSE	FALSE
6816	5280	15737	60000	FALSE	FALSE
6817	6700	15737	97046	FALSE	FALSE
6818	3750	15737	92000	FALSE	FALSE
6819	2500	15737	115000	FALSE	FALSE
6820	3600	15737	20500	FALSE	FALSE
6821	6150	15737	130000	FALSE	FALSE
6822	22275	15737	151000	FALSE	FALSE
6823	5852	15737	136000	FALSE	FALSE
6824	8320	15737	172500	FALSE	FALSE
6825	7740	15737	149500	FALSE	FALSE
6826	4455	15737	127900	FALSE	FALSE
6827	4800	15737	111750	FALSE	FALSE
6828	3600	15737	82500	FALSE	FALSE
6829	5166	15737	98000	FALSE	FALSE
6830	3600	15737	135000	FALSE	FALSE
6831	4515	15737	135000	FALSE	FALSE
6832	4170	15737	174519	FALSE	FALSE
6834	4060	15737	185000	FALSE	TRUE
6835	7506	15737	138000	FALSE	FALSE
6836	4290	15737	188900	FALSE	FALSE
6837	3660	15737	20000	FALSE	FALSE
6838	5040	15737	195000	FALSE	FALSE
6839	5348	15737	114000	FALSE	FALSE
6840	4500	15737	30000	FALSE	FALSE
6841	3600	15737	17000	FALSE	FALSE
6842	9350	15765	110500	FALSE	FALSE
6843	4960	15765	63950	FALSE	FALSE
6844	6063	15765	8000	FALSE	FALSE
6845	5312	15765	69900	FALSE	FALSE
6846	5658	15765	12500	FALSE	FALSE
6847	7200	15765	80600	FALSE	FALSE
6848	6000	15765	54000	FALSE	FALSE

6849	5040	15765	79900	FALSE	FALSE
6850	5000	15765	87000	FALSE	FALSE
6851	5445	15765	85000	FALSE	FALSE
6852	6500	15765	89000	FALSE	FALSE
6853	6000	15765	35000	FALSE	FALSE
6854	6000	15765	58000	FALSE	FALSE
6855	5610	15765	58000	FALSE	FALSE
6856	5428	15765	166000	TRUE	FALSE
6857	3000	15765	80000	TRUE	FALSE
6858	4800	15765	347500	TRUE	FALSE
6859	5445	15765	260000	TRUE	TRUE
6860	3600	15765	110000	TRUE	FALSE
6861	3600	15765	126500	TRUE	TRUE
6862	3600	15765	72000	TRUE	FALSE
6863	5800	15765	619315	TRUE	FALSE
6864	6000	15765	480000	TRUE	TRUE
6865	2580	15765	325000	TRUE	FALSE
6866	4800	15765	272000	TRUE	FALSE
6867	4800	15765	388000	TRUE	TRUE
6868	3210	15765	217500	TRUE	FALSE
6869	4920	15765	165000	TRUE	FALSE
6870	6396	15765	60000	FALSE	FALSE
6871	3750	15765	17000	FALSE	FALSE
6872	6336	15765	77000	FALSE	FALSE
6873	6400	15765	146000	FALSE	FALSE
6874	5360	15765	155000	FALSE	FALSE
6875	5000	15765	129000	FALSE	FALSE
6876	5200	15765	89900	FALSE	FALSE
6877	4800	15765	120000	FALSE	FALSE
6878	15756	15765	192000	FALSE	FALSE
6879	5888	15765	125500	FALSE	FALSE
6880	7920	15765	122000	FALSE	FALSE
6881	20135	15765	310000	FALSE	FALSE
6882	10950	15765	115900	FALSE	FALSE
6883	8400	15765	75000	FALSE	FALSE
6884	7320	15765	157500	FALSE	FALSE
6885	13524	15765	105000	FALSE	FALSE
6886	6120	15765	149000	FALSE	FALSE
6887	8400	15765	90000	FALSE	FALSE
6888	5760	15765	97000	FALSE	FALSE
6889	6250	15765	109000	FALSE	FALSE
6890	6750	15765	114900	FALSE	FALSE
6891	6000	15765	68500	FALSE	FALSE

6892	9600	15765	117000	FALSE	FALSE
6893	4070	15765	22500	FALSE	FALSE
6894	4200	15765	154900	FALSE	FALSE
6895	4400	15765	30000	FALSE	FALSE
6896	7956	15765	8000	FALSE	FALSE
6897	3320	15765	20000	FALSE	FALSE
6898	5289	15765	53900	FALSE	FALSE
6899	7420	15765	133500	FALSE	FALSE
6900	8064	15765	94000	FALSE	FALSE
6901	4800	15765	72000	FALSE	FALSE
6902	3600	15765	80000	FALSE	FALSE
6903	3500	15765	64000	FALSE	FALSE
6904	9000	15765	204100	FALSE	FALSE
6905	8878	15765	105000	FALSE	FALSE
6906	12464	15765	106500	FALSE	FALSE
6907	7200	15765	115000	FALSE	FALSE
6908	8910	15765	76000	FALSE	FALSE
6909	7920	15765	77000	FALSE	FALSE
6910	5760	15765	65000	FALSE	FALSE
6911	3680	15765	138000	FALSE	FALSE
6912	4872	15765	115000	FALSE	FALSE
6913	5160	15765	141000	FALSE	FALSE
6914	4920	15765	88800	FALSE	FALSE
6915	8050	15765	300000	FALSE	FALSE
6916	7150	15765	204500	FALSE	FALSE
6917	4600	15765	73500	FALSE	FALSE
6918	4879	15765	80000	FALSE	FALSE
6919	5250	15765	119500	FALSE	FALSE
6920	4859	15765	121000	FALSE	FALSE
6921	4800	15765	128000	FALSE	FALSE
6922	13600	15765	221000	FALSE	FALSE
6923	5400	15765	140000	FALSE	FALSE
6924	5200	15765	126000	FALSE	FALSE
6925	5000	15765	118500	FALSE	FALSE
6926	5400	15765	85000	FALSE	FALSE
6927	7980	15765	149800	FALSE	FALSE
6928	5000	15765	127900	FALSE	FALSE
6929	4840	15765	116500	FALSE	FALSE
6930	19520	15765	149000	FALSE	FALSE
6931	4960	15765	188000	FALSE	TRUE
6932	5280	15765	154500	FALSE	FALSE
6933	7699	15765	320000	FALSE	FALSE
6934	6450	15765	117000	FALSE	FALSE

6935	5400	15765	105000	FALSE	FALSE
6936	6480	15765	121800	FALSE	FALSE
6937	8235	15765	149000	FALSE	FALSE
6938	6000	15765	114000	FALSE	FALSE
6939	6120	15765	113000	FALSE	FALSE
6940	5980	15765	104500	FALSE	FALSE
6941	6750	15765	125000	FALSE	FALSE
6942	7000	15765	115000	FALSE	FALSE
6943	3500	15765	44900	FALSE	FALSE
6944	3560	15765	70000	FALSE	FALSE
6945	3840	15765	60000	FALSE	FALSE
6946	21780	15765	322000	FALSE	FALSE
6947	7200	15765	231500	FALSE	FALSE
6948	5418	15765	120000	FALSE	FALSE
6949	4641	15765	104900	FALSE	FALSE
6950	6250	15765	254900	FALSE	FALSE
6951	5805	15765	130000	FALSE	FALSE
6952	6700	15765	138000	FALSE	FALSE
6953	4840	15765	113500	FALSE	FALSE
6954	8100	15765	145000	FALSE	FALSE
6955	6784	15765	148000	FALSE	FALSE
6956	6864	15765	120000	FALSE	FALSE
6957	6450	15765	133200	FALSE	FALSE
6958	4800	15765	108000	FALSE	FALSE
6959	6600	15765	149900	FALSE	FALSE
6960	7089	15765	98500	FALSE	FALSE
6961	9180	15765	136000	FALSE	FALSE
6962	5700	15765	138000	FALSE	FALSE
6963	6550	15765	118000	FALSE	FALSE
6964	3600	15765	149000	FALSE	FALSE
6965	3680	15765	103500	FALSE	FALSE
6966	1200	15765	115000	FALSE	FALSE
6967	3600	15765	170000	FALSE	TRUE
6968	3720	15765	55000	FALSE	FALSE
6969	5880	15765	146300	FALSE	FALSE
6970	4880	15765	130000	FALSE	FALSE
6971	5400	15765	194500	FALSE	TRUE
6972	9120	15765	162500	FALSE	FALSE
6973	5805	15765	235000	FALSE	FALSE
6974	3690	15765	129200	FALSE	FALSE
6975	3750	15765	166500	FALSE	FALSE
6976	6280	15765	164900	FALSE	FALSE
6977	4720	15765	51500	FALSE	FALSE

6978	4800	15796	39000	FALSE	FALSE
6979	5848	15796	35000	FALSE	FALSE
6980	3840	15796	27000	FALSE	FALSE
6981	3840	15796	36300	FALSE	FALSE
6982	4680	15796	32500	FALSE	FALSE
6983	8845	15796	80000	FALSE	FALSE
6984	5120	15796	69000	FALSE	FALSE
6985	7662	15796	93000	FALSE	FALSE
6986	11205	15796	95000	FALSE	FALSE
6987	8100	15796	67000	FALSE	FALSE
6988	5160	15796	73000	FALSE	FALSE
6989	6000	15796	69000	FALSE	FALSE
6990	5160	15796	64900	FALSE	FALSE
6991	6360	15796	49900	FALSE	FALSE
6992	7320	15796	420000	TRUE	TRUE
6993	4488	15796	425500	TRUE	TRUE
6994	3600	15796	216000	TRUE	TRUE
6995	3600	15796	131500	TRUE	TRUE
6996	3540	15796	153850	TRUE	FALSE
6997	6728	15796	450000	TRUE	TRUE
6998	5490	15796	387500	TRUE	TRUE
6999	5400	15796	380000	TRUE	TRUE
7000	4148	15796	399000	TRUE	TRUE
7001	3600	15796	200000	TRUE	FALSE
7002	3150	15796	236000	TRUE	FALSE
7003	2240	15796	173000	TRUE	FALSE
7004	6298	15796	130000	TRUE	FALSE
7005	2970	15796	214000	TRUE	TRUE
7006	3600	15796	193500	TRUE	TRUE
7007	4800	15796	162000	TRUE	TRUE
7008	3840	15796	160000	TRUE	FALSE
7009	7200	15796	596800	TRUE	TRUE
7010	3150	15796	30000	FALSE	FALSE
7011	5120	15796	167000	FALSE	FALSE
7012	7200	15796	195000	FALSE	FALSE
7013	5625	15796	184900	FALSE	FALSE
7014	5120	15796	155000	FALSE	FALSE
7015	5120	15796	96500	FALSE	FALSE
7016	6960	15796	104750	FALSE	FALSE
7017	5000	15796	166000	FALSE	FALSE
7018	17473	15796	104900	FALSE	FALSE
7019	5684	15796	89900	FALSE	FALSE
7020	5000	15796	110000	FALSE	FALSE

7021	4800	15796	118500	FALSE	FALSE
7022	6800	15796	82000	FALSE	FALSE
7023	6000	15796	112000	FALSE	FALSE
7024	20801	15796	167900	FALSE	FALSE
7025	5760	15796	193700	FALSE	FALSE
7026	5760	15796	135000	FALSE	FALSE
7027	10950	15796	194000	FALSE	FALSE
7028	7590	15796	122000	FALSE	FALSE
7029	6650	15796	103000	FALSE	FALSE
7030	7200	15796	155000	FALSE	FALSE
7031	7875	15796	116500	FALSE	FALSE
7032	4800	15796	117000	FALSE	FALSE
7033	7130	15796	81000	FALSE	FALSE
7034	6000	15796	115500	FALSE	FALSE
7035	6000	15796	70000	FALSE	FALSE
7036	7764	15796	360000	FALSE	FALSE
7037	3250	15796	198000	FALSE	FALSE
7038	3250	15796	195000	FALSE	FALSE
7039	2640	15796	19000	FALSE	FALSE
7040	2310	15796	11000	FALSE	FALSE
7041	5100	15796	21000	FALSE	FALSE
7042	5760	15796	23000	FALSE	FALSE
7043	2250	15796	104000	FALSE	FALSE
7044	7550	15796	380500	FALSE	FALSE
7045	2100	15796	9000	FALSE	FALSE
7046	5740	15796	84500	FALSE	FALSE
7047	5376	15796	112700	FALSE	FALSE
7048	4800	15796	50000	FALSE	FALSE
7049	5160	15796	19000	FALSE	FALSE
7050	5040	15796	131500	FALSE	FALSE
7051	7009	15796	44900	FALSE	FALSE
7052	3960	15796	12500	FALSE	FALSE
7053	3750	15796	41000	FALSE	FALSE
7054	4484	15796	90000	FALSE	FALSE
7055	3600	15796	70000	FALSE	FALSE
7056	3750	15796	80000	FALSE	FALSE
7057	5200	15796	61000	FALSE	FALSE
7058	9000	15796	128700	FALSE	FALSE
7059	7200	15796	129000	FALSE	FALSE
7060	5136	15796	180000	FALSE	FALSE
7061	6254	15796	169900	FALSE	FALSE
7062	5160	15796	141000	FALSE	FALSE
7063	6075	15796	87000	FALSE	FALSE

7064	5160	15796	98000	FALSE	FALSE
7065	5400	15796	145000	FALSE	FALSE
7066	5117	15796	171000	FALSE	FALSE
7067	8320	15796	187500	FALSE	FALSE
7068	4800	15796	169000	FALSE	FALSE
7069	4680	15796	53000	FALSE	FALSE
7070	4400	15796	142500	FALSE	FALSE
7071	4524	15796	63000	FALSE	FALSE
7072	3588	15796	139000	FALSE	FALSE
7073	4200	15796	118000	FALSE	FALSE
7074	4800	15796	124000	FALSE	FALSE
7075	5400	15796	90000	FALSE	FALSE
7076	5460	15796	56200	FALSE	FALSE
7077	8417	15796	193000	FALSE	TRUE
7078	6150	15796	139000	FALSE	FALSE
7079	6375	15796	112000	FALSE	FALSE
7080	6120	15796	98000	FALSE	FALSE
7081	7560	15796	118750	FALSE	FALSE
7082	5040	15796	152900	FALSE	FALSE
7083	5920	15796	139900	FALSE	FALSE
7084	6600	15796	139900	FALSE	FALSE
7085	9135	15796	119900	FALSE	FALSE
7086	4944	15796	141000	FALSE	FALSE
7087	7750	15796	70000	FALSE	FALSE
7088	7800	15796	127000	FALSE	FALSE
7089	10788	15796	133000	FALSE	FALSE
7090	10816	15796	107000	FALSE	FALSE
7091	5000	15796	81500	FALSE	FALSE
7092	8643	15796	78000	FALSE	FALSE
7093	11487	15796	285000	FALSE	FALSE
7094	4140	15796	89900	FALSE	FALSE
7095	2580	15796	37500	FALSE	FALSE
7096	1920	15796	71500	FALSE	FALSE
7098	3536	15796	127000	FALSE	FALSE
7099	6000	15796	97000	FALSE	FALSE
7100	4720	15796	146000	FALSE	FALSE
7101	7584	15796	140000	FALSE	FALSE
7102	17500	15796	168000	FALSE	FALSE
7103	7076	15796	147500	FALSE	FALSE
7104	7228	15796	120900	FALSE	FALSE
7105	6000	15796	129900	FALSE	FALSE
7106	4560	15796	119900	FALSE	FALSE
7107	6550	15796	115000	FALSE	FALSE

7108	4840	15796	92000	FALSE	FALSE
7109	4800	15796	89000	FALSE	FALSE
7110	4800	15796	78500	FALSE	FALSE
7111	5400	15796	88900	FALSE	FALSE
7112	4800	15796	60000	FALSE	FALSE
7113	6250	15796	172000	FALSE	FALSE
7114	4120	15796	113000	FALSE	FALSE
7115	5952	15796	166000	FALSE	FALSE
7116	4160	15796	109500	FALSE	FALSE
7117	4920	15796	157704	FALSE	FALSE
7118	4032	15796	150000	FALSE	FALSE
7119	3390	15796	133500	FALSE	FALSE
7120	6240	15796	125000	FALSE	FALSE
7121	4758	15796	149900	FALSE	FALSE
7122	4800	15796	255000	FALSE	FALSE
7123	3600	15796	163000	FALSE	FALSE
7124	4914	15796	145500	FALSE	FALSE
7125	4880	15796	105000	FALSE	FALSE
7126	4514	15796	107900	FALSE	FALSE
7127	4800	15796	86500	FALSE	FALSE
7128	6270	15796	230500	FALSE	FALSE
7129	2760	15796	172000	FALSE	FALSE
7130	3000	15796	180000	FALSE	FALSE
7131	3390	15796	134500	FALSE	FALSE
7132	2790	15796	141750	FALSE	FALSE
7133	3600	15796	132200	FALSE	FALSE
7134	3600	15796	185000	FALSE	TRUE
7135	3510	15796	220000	FALSE	FALSE
7136	6732	15796	60000	FALSE	FALSE
7137	4080	15796	27500	FALSE	FALSE
7138	4080	15796	44850	FALSE	FALSE
7139	4853	15796	134000	FALSE	FALSE
7140	6350	15826	55000	FALSE	FALSE
7141	4200	15826	43092	FALSE	FALSE
7142	5084	15826	72600	FALSE	FALSE
7143	5080	15826	53000	FALSE	FALSE
7144	5664	15826	77000	FALSE	FALSE
7145	4620	15826	55000	FALSE	FALSE
7146	5125	15826	109900	FALSE	FALSE
7147	5200	15826	108900	FALSE	FALSE
7148	10500	15826	105000	FALSE	FALSE
7149	9000	15826	129900	FALSE	FALSE
7150	5320	15826	79000	FALSE	FALSE

7151	7560	15826	119000	FALSE	FALSE
7152	4800	15826	65000	FALSE	FALSE
7153	4800	15826	61000	FALSE	FALSE
7154	7245	15826	72500	FALSE	FALSE
7155	7245	15826	92000	FALSE	FALSE
7156	5985	15826	75000	FALSE	FALSE
7157	8550	15826	580000	TRUE	TRUE
7158	4440	15826	133000	TRUE	FALSE
7159	11250	15826	735000	TRUE	FALSE
7160	8400	15826	575000	TRUE	FALSE
7161	5535	15826	179900	TRUE	FALSE
7162	4800	15826	250000	TRUE	FALSE
7163	3600	15826	252000	TRUE	TRUE
7164	5800	15826	420000	TRUE	FALSE
7165	3600	15826	214000	TRUE	TRUE
7166	3600	15826	173000	TRUE	FALSE
7167	3660	15826	125000	TRUE	FALSE
7168	7200	15826	383000	TRUE	TRUE
7169	5760	15826	38000	FALSE	FALSE
7170	5310	15826	62000	FALSE	FALSE
7171	10125	15826	150000	FALSE	FALSE
7172	6000	15826	192000	FALSE	FALSE
7173	5120	15826	186600	FALSE	FALSE
7174	6120	15826	155000	FALSE	FALSE
7175	5376	15826	169000	FALSE	FALSE
7176	5632	15826	143500	FALSE	FALSE
7177	5760	15826	128500	FALSE	FALSE
7178	4800	15826	132000	FALSE	FALSE
7179	4800	15826	110000	FALSE	FALSE
7180	7040	15826	157000	FALSE	FALSE
7181	5888	15826	147900	FALSE	FALSE
7182	6250	15826	158820	FALSE	TRUE
7183	5000	15826	152700	FALSE	FALSE
7184	5000	15826	163000	FALSE	FALSE
7185	5310	15826	124500	FALSE	FALSE
7186	5120	15826	129900	FALSE	FALSE
7187	5000	15826	119500	FALSE	FALSE
7188	4800	15826	147500	FALSE	TRUE
7189	6000	15826	140000	FALSE	FALSE
7190	4800	15826	135000	FALSE	FALSE
7191	6800	15826	103800	FALSE	FALSE
7192	6400	15826	115600	FALSE	FALSE
7193	15738	15826	315000	FALSE	FALSE

7194	9525	15826	160000	FALSE	FALSE
7195	5625	15826	191000	FALSE	FALSE
7196	5000	15826	90400	FALSE	FALSE
7197	19800	15826	218000	FALSE	FALSE
7198	15519	15826	179000	FALSE	FALSE
7199	11025	15826	180000	FALSE	FALSE
7200	5280	15826	81000	FALSE	FALSE
7201	5040	15826	105000	FALSE	FALSE
7202	12600	15826	85000	FALSE	FALSE
7203	5310	15826	92000	FALSE	FALSE
7204	9020	15826	112000	FALSE	FALSE
7205	7200	15826	88000	FALSE	FALSE
7206	6120	15826	99000	FALSE	FALSE
7207	7320	15826	135000	FALSE	FALSE
7208	6837	15826	108500	FALSE	FALSE
7209	4920	15826	85500	FALSE	FALSE
7210	6360	15826	69000	FALSE	FALSE
7211	5000	15826	75000	FALSE	FALSE
7212	5360	15826	130000	FALSE	FALSE
7213	9720	15826	129000	FALSE	FALSE
7214	7764	15826	357500	FALSE	FALSE
7215	7326	15826	22000	FALSE	FALSE
7216	5550	15826	45000	FALSE	FALSE
7217	3600	15826	188000	FALSE	FALSE
7218	3775	15826	173000	FALSE	FALSE
7219	6400	15826	89400	FALSE	FALSE
7220	9950	15826	218000	FALSE	FALSE
7221	5220	15826	55000	FALSE	FALSE
7222	5125	15826	106000	FALSE	FALSE
7223	5125	15826	106000	FALSE	FALSE
7224	4200	15826	44602	FALSE	FALSE
7225	5850	15826	126000	FALSE	FALSE
7226	2850	15826	8000	FALSE	FALSE
7227	2850	15826	13000	FALSE	FALSE
7228	2850	15826	20000	FALSE	FALSE
7229	2160	15826	47000	FALSE	FALSE
7230	7200	15826	150000	FALSE	FALSE
7231	12300	15826	107000	FALSE	FALSE
7232	4900	15826	192900	FALSE	FALSE
7233	6075	15826	55000	FALSE	FALSE
7234	4600	15826	106000	FALSE	FALSE
7235	4800	15826	79000	FALSE	FALSE
7236	6292	15826	90000	FALSE	FALSE

7237	5014	15826	138500	FALSE	FALSE
7238	4940	15826	195000	FALSE	FALSE
7239	4165	15826	155670	FALSE	FALSE
7240	2720	15826	70000	FALSE	FALSE
7241	5800	15826	133000	FALSE	FALSE
7242	4800	15826	155000	FALSE	FALSE
7243	4171	15826	67500	FALSE	FALSE
7244	7920	15826	89900	FALSE	FALSE
7245	6760	15826	163000	FALSE	FALSE
7246	8750	15826	145000	FALSE	FALSE
7247	5280	15826	120000	FALSE	FALSE
7248	5040	15826	135000	FALSE	FALSE
7249	4800	15826	118000	FALSE	FALSE
7250	6480	15826	111000	FALSE	FALSE
7251	7440	15826	129500	FALSE	FALSE
7252	8174	15826	168000	FALSE	FALSE
7253	6360	15826	86500	FALSE	FALSE
7254	4836	15826	123000	FALSE	FALSE
7255	7410	15826	238000	FALSE	FALSE
7256	4920	15826	151500	FALSE	FALSE
7257	6600	15826	176000	FALSE	FALSE
7258	6360	15826	147900	FALSE	FALSE
7259	6120	15826	136900	FALSE	FALSE
7260	7860	15826	137000	FALSE	FALSE
7261	11289	15826	118000	FALSE	FALSE
7262	7320	15826	126000	FALSE	FALSE
7263	5000	15826	105000	FALSE	FALSE
7264	6900	15826	116000	FALSE	FALSE
7265	5500	15826	110000	FALSE	FALSE
7266	6042	15826	125000	FALSE	FALSE
7267	6000	15826	70000	FALSE	FALSE
7268	5280	15826	92000	FALSE	FALSE
7269	7493	15826	110000	FALSE	FALSE
7270	7846	15826	225000	FALSE	FALSE
7271	7150	15826	192000	FALSE	FALSE
7272	5520	15826	180000	FALSE	FALSE
7273	4830	15826	110000	FALSE	TRUE
7274	4248	15826	50000	FALSE	FALSE
7275	2820	15826	73000	FALSE	FALSE
7276	6100	15826	124500	FALSE	FALSE
7277	4760	15826	83000	FALSE	FALSE
7278	5600	15826	127000	FALSE	FALSE
7279	8385	15826	94000	FALSE	FALSE

7280	5000	15826	112000	FALSE	FALSE
7281	6292	15826	199900	FALSE	FALSE
7282	5080	15826	188900	FALSE	FALSE
7283	8520	15826	162000	FALSE	FALSE
7284	7500	15826	125794	FALSE	FALSE
7285	9648	15826	151500	FALSE	FALSE
7286	6050	15826	106500	FALSE	FALSE
7287	8400	15826	145000	FALSE	FALSE
7288	5805	15826	96300	FALSE	FALSE
7289	6950	15826	124000	FALSE	FALSE
7290	9126	15826	108000	FALSE	FALSE
7291	4800	15826	90000	FALSE	FALSE
7292	4800	15826	79900	FALSE	FALSE
7293	6578	15826	109900	FALSE	FALSE
7294	5355	15826	142000	FALSE	FALSE
7295	7128	15826	169900	FALSE	FALSE
7296	7200	15826	151300	FALSE	FALSE
7297	5640	15826	147500	FALSE	FALSE
7298	4960	15826	143000	FALSE	FALSE
7299	5760	15826	252500	FALSE	FALSE
7300	3660	15826	24000	FALSE	FALSE
7301	3660	15826	38000	FALSE	FALSE
7302	3870	15826	163500	FALSE	TRUE
7303	5400	15826	94500	FALSE	FALSE
7304	3510	15826	149000	FALSE	FALSE
7305	4800	15826	55000	FALSE	FALSE
7306	4200	15826	233000	FALSE	FALSE
7307	3060	15826	212500	FALSE	FALSE
7308	4200	15826	111000	FALSE	FALSE
7309	3600	15826	235000	FALSE	TRUE
7310	3300	15826	275000	FALSE	FALSE
7311	4720	15826	21500	FALSE	FALSE
7312	7500	15826	129000	FALSE	FALSE
7313	7750	15826	9200	FALSE	FALSE
7314	3600	15826	23900	FALSE	FALSE
7315	7068	15857	142500	FALSE	FALSE
7316	4040	15857	38000	FALSE	FALSE
7317	6600	15857	72145	FALSE	FALSE
7318	6210	15857	41500	FALSE	FALSE
7319	6840	15857	77000	FALSE	FALSE
7320	6160	15857	20500	FALSE	FALSE
7321	5400	15857	36500	FALSE	FALSE
7322	5520	15857	49900	FALSE	FALSE

7323	4920	15857	32000	FALSE	FALSE
7324	4920	15857	59900	FALSE	FALSE
7325	5445	15857	119900	FALSE	FALSE
7326	5330	15857	103000	FALSE	FALSE
7327	5330	15857	75000	FALSE	FALSE
7328	4800	15857	73000	FALSE	FALSE
7329	7245	15857	78000	FALSE	FALSE
7330	5292	15857	49000	FALSE	FALSE
7331	7448	15857	22000	FALSE	FALSE
7332	7500	15857	425000	TRUE	TRUE
7333	8469	15857	175000	TRUE	FALSE
7334	4375	15857	160000	TRUE	FALSE
7335	3972	15857	206000	TRUE	TRUE
7336	8700	15857	545000	TRUE	TRUE
7337	8540	15857	296000	TRUE	TRUE
7338	3600	15857	219000	TRUE	TRUE
7339	3600	15857	136000	TRUE	FALSE
7340	3600	15857	125000	TRUE	TRUE
7341	2540	15857	135000	TRUE	TRUE
7342	6960	15857	750000	TRUE	TRUE
7343	9240	15857	545000	TRUE	FALSE
7344	6355	15857	129000	TRUE	TRUE
7345	5400	15857	670000	TRUE	TRUE
7346	6000	15857	369000	TRUE	TRUE
7347	12000	15857	545000	TRUE	TRUE
7348	2600	15857	285000	TRUE	TRUE
7349	2071	15857	145000	TRUE	FALSE
7350	4800	15857	210000	TRUE	TRUE
7351	3600	15857	50000	TRUE	FALSE
7352	3625	15857	127000	TRUE	TRUE
7353	7200	15857	15000	FALSE	FALSE
7354	4600	15857	127500	FALSE	FALSE
7355	6600	15857	12000	FALSE	FALSE
7356	5376	15857	135000	FALSE	FALSE
7357	4960	15857	117500	FALSE	FALSE
7358	8640	15857	144000	FALSE	FALSE
7359	5040	15857	162000	FALSE	FALSE
7360	4800	15857	80000	FALSE	FALSE
7361	4800	15857	131000	FALSE	FALSE
7362	5000	15857	155500	FALSE	FALSE
7363	6336	15857	120000	FALSE	FALSE
7364	5460	15857	160000	FALSE	FALSE
7365	4960	15857	83500	FALSE	FALSE

7366	7000	15857	106000	FALSE	FALSE
7367	5120	15857	98000	FALSE	FALSE
7368	5760	15857	131000	FALSE	FALSE
7369	5000	15857	165000	FALSE	FALSE
7370	8320	15857	124500	FALSE	FALSE
7371	6300	15857	138000	FALSE	FALSE
7372	4800	15857	115000	FALSE	FALSE
7373	7680	15857	115000	FALSE	FALSE
7374	5000	15857	178500	FALSE	FALSE
7375	6050	15857	87500	FALSE	FALSE
7376	10125	15857	150000	FALSE	FALSE
7377	11250	15857	149500	FALSE	FALSE
7378	5320	15857	109500	FALSE	FALSE
7379	6120	15857	70000	FALSE	FALSE
7380	5760	15857	125500	FALSE	FALSE
7381	7200	15857	70000	FALSE	FALSE
7382	4840	15857	129900	FALSE	FALSE
7383	6000	15857	78900	FALSE	FALSE
7384	8400	15857	85500	FALSE	FALSE
7385	4800	15857	134900	FALSE	FALSE
7386	5916	15857	79900	FALSE	FALSE
7387	4687	15857	86500	FALSE	FALSE
7388	6625	15857	205000	FALSE	FALSE
7389	5520	15857	32900	FALSE	FALSE
7390	3450	15857	90060	FALSE	FALSE
7391	5850	15857	34000	FALSE	FALSE
7392	1944	15857	168000	FALSE	FALSE
7393	3630	15857	120000	FALSE	FALSE
7394	8410	15857	34000	FALSE	FALSE
7395	4563	15857	70000	FALSE	FALSE
7396	6000	15857	69000	FALSE	FALSE
7397	6171	15857	126250	FALSE	FALSE
7398	4640	15857	65000	FALSE	FALSE
7399	4800	15857	89900	FALSE	FALSE
7400	4950	15857	35000	FALSE	FALSE
7401	3840	15857	85000	FALSE	FALSE
7402	3960	15857	27000	FALSE	FALSE
7403	3750	15857	109000	FALSE	FALSE
7404	4320	15857	69900	FALSE	FALSE
7405	4445	15857	68000	FALSE	FALSE
7406	7200	15857	130000	FALSE	FALSE
7407	9014	15857	224000	FALSE	FALSE
7408	9655	15857	135000	FALSE	FALSE

7409	10000	15857	105000	FALSE	FALSE
7410	9000	15857	115000	FALSE	FALSE
7411	8100	15857	95400	FALSE	FALSE
7412	6250	15857	99900	FALSE	FALSE
7413	6728	15857	217000	FALSE	FALSE
7414	4715	15857	139000	FALSE	FALSE
7415	8472	15857	190000	FALSE	FALSE
7416	5000	15857	143000	FALSE	FALSE
7417	5760	15857	173000	FALSE	FALSE
7418	5400	15857	105900	FALSE	FALSE
7419	7440	15857	135000	FALSE	FALSE
7420	5145	15857	214000	FALSE	FALSE
7421	5160	15857	178000	FALSE	FALSE
7422	6570	15857	212900	FALSE	FALSE
7423	3360	15857	133000	FALSE	FALSE
7424	4800	15857	224900	FALSE	FALSE
7425	5080	15857	203500	FALSE	FALSE
7426	3905	15857	99200	FALSE	FALSE
7427	4800	15857	99900	FALSE	FALSE
7428	5085	15857	147500	FALSE	FALSE
7429	4235	15857	63000	FALSE	FALSE
7430	7248	15857	210000	FALSE	FALSE
7431	5400	15857	140000	FALSE	FALSE
7432	4760	15857	157500	FALSE	FALSE
7433	4816	15857	124000	FALSE	FALSE
7434	7280	15857	95000	FALSE	FALSE
7435	4800	15857	185000	FALSE	FALSE
7436	5654	15857	225500	FALSE	FALSE
7437	6750	15857	227000	FALSE	FALSE
7438	3720	15857	106500	FALSE	FALSE
7439	4080	15857	157900	FALSE	FALSE
7440	6650	15857	88500	FALSE	FALSE
7441	5590	15857	280000	FALSE	TRUE
7442	4326	15857	125000	FALSE	FALSE
7443	4953	15857	115000	FALSE	FALSE
7444	4875	15857	127825	FALSE	FALSE
7445	4880	15857	99900	FALSE	FALSE
7446	7093	15857	190000	FALSE	FALSE
7447	7750	15857	170000	FALSE	FALSE
7448	5040	15857	160000	FALSE	FALSE
7449	15000	15857	195000	FALSE	FALSE
7450	7200	15857	109000	FALSE	FALSE
7451	13510	15857	143000	FALSE	FALSE

7452	7320	15857	144900	FALSE	FALSE
7453	6324	15857	120900	FALSE	FALSE
7454	7686	15857	125500	FALSE	FALSE
7455	7200	15857	130000	FALSE	FALSE
7456	11832	15857	100000	FALSE	FALSE
7457	7625	15857	133000	FALSE	FALSE
7458	5000	15857	131000	FALSE	FALSE
7459	8220	15857	93000	FALSE	FALSE
7460	6000	15857	92000	FALSE	FALSE
7461	5500	15857	114900	FALSE	FALSE
7462	5880	15857	117500	FALSE	FALSE
7463	6700	15857	111000	FALSE	FALSE
7464	6150	15857	44188	FALSE	FALSE
7465	7320	15857	203000	FALSE	FALSE
7466	3600	15857	27500	FALSE	FALSE
7467	5900	15857	115000	FALSE	FALSE
7468	9435	15857	149900	FALSE	FALSE
7469	4720	15857	104000	FALSE	FALSE
7470	4800	15857	99000	FALSE	FALSE
7471	5750	15857	190000	FALSE	FALSE
7472	3600	15857	85000	FALSE	FALSE
7473	10660	15857	187000	FALSE	FALSE
7474	14800	15857	154000	FALSE	FALSE
7475	6450	15857	145000	FALSE	FALSE
7476	6840	15857	178000	FALSE	FALSE
7477	10140	15857	159000	FALSE	FALSE
7478	7740	15857	140000	FALSE	FALSE
7479	6450	15857	106000	FALSE	FALSE
7480	14160	15857	110000	FALSE	FALSE
7481	8120	15857	84500	FALSE	FALSE
7482	6480	15857	124900	FALSE	FALSE
7483	6450	15857	110000	FALSE	FALSE
7484	5640	15857	80000	FALSE	FALSE
7485	4680	15857	94900	FALSE	FALSE
7486	6076	15857	185000	FALSE	FALSE
7487	5040	15857	117000	FALSE	FALSE
7488	4644	15857	100000	FALSE	FALSE
7489	4142	15857	150500	FALSE	FALSE
7490	4880	15857	101500	FALSE	FALSE
7491	5580	15857	222000	FALSE	FALSE
7492	1848	15857	184500	FALSE	FALSE
7493	2880	15857	23000	FALSE	FALSE
7494	4800	15857	187000	FALSE	FALSE

7495	3660	15857	125000	FALSE	FALSE
7496	2884	15857	99900	FALSE	FALSE
7497	3600	15857	42500	FALSE	FALSE
7498	3502	15857	170100	FALSE	FALSE
7499	4560	15857	173000	FALSE	FALSE
7500	3660	15857	105000	FALSE	FALSE
7501	4636	15857	131500	FALSE	FALSE
7502	4880	15857	112000	FALSE	FALSE
7503	3720	15857	130000	FALSE	FALSE
7504	4800	15857	125000	FALSE	FALSE
7505	3600	15857	125000	FALSE	FALSE
7506	5043	15857	119500	FALSE	FALSE
7507	10320	15857	169900	FALSE	FALSE
7508	3465	15857	185000	FALSE	FALSE
7509	2400	15857	129000	FALSE	FALSE
7510	3600	15857	104500	FALSE	FALSE
7511	5490	15857	103000	FALSE	FALSE
7512	4410	15857	148500	FALSE	FALSE
7513	4725	15857	100000	FALSE	FALSE
7514	5625	15857	136924	FALSE	FALSE
7515	2441	15857	138000	FALSE	FALSE
7516	3660	15857	144100	FALSE	FALSE
7517	5490	15857	151500	FALSE	FALSE
7518	4853	15857	134000	FALSE	FALSE
7519	4640	15857	115000	FALSE	FALSE
7520	4800	15857	50000	FALSE	FALSE
7521	3360	15857	21900	FALSE	FALSE
7522	5000	15857	111600	FALSE	FALSE
7523	5000	15857	106800	FALSE	FALSE
7524	6240	15887	68000	FALSE	FALSE
7525	4800	15887	31500	FALSE	FALSE
7526	8532	15887	69900	FALSE	FALSE
7527	6000	15887	99000	FALSE	FALSE
7528	5000	15887	76900	FALSE	FALSE
7529	5000	15887	80000	FALSE	FALSE
7530	4920	15887	83000	FALSE	FALSE
7531	5160	15887	60000	FALSE	FALSE
7532	5850	15887	88500	FALSE	FALSE
7533	5840	15887	68500	FALSE	FALSE
7534	7980	15887	85000	FALSE	FALSE
7535	6776	15887	77000	FALSE	FALSE
7536	6500	15887	111900	FALSE	FALSE
7537	5640	15887	180500	TRUE	FALSE

7538	5166	15887	243000	TRUE	TRUE
7539	2905	15887	207000	TRUE	TRUE
7540	3600	15887	156000	TRUE	TRUE
7541	3600	15887	164900	TRUE	FALSE
7542	3555	15887	202000	TRUE	TRUE
7543	3200	15887	180000	TRUE	TRUE
7544	3600	15887	128900	TRUE	FALSE
7545	3600	15887	181000	TRUE	TRUE
7546	8400	15887	597500	TRUE	TRUE
7547	8250	15887	585000	TRUE	TRUE
7548	4800	15887	350000	TRUE	TRUE
7549	7200	15887	411000	TRUE	TRUE
7550	4800	15887	302000	TRUE	TRUE
7551	5080	15887	205000	TRUE	FALSE
7553	6100	15887	284000	TRUE	TRUE
7554	1410	15887	238000	TRUE	TRUE
7555	6250	15887	213500	FALSE	FALSE
7556	5900	15887	159000	FALSE	FALSE
7557	4800	15887	164900	FALSE	FALSE
7558	5000	15887	119500	FALSE	FALSE
7559	5000	15887	155000	FALSE	FALSE
7560	4800	15887	103000	FALSE	FALSE
7561	4800	15887	146000	FALSE	FALSE
7562	5040	15887	132700	FALSE	FALSE
7563	6000	15887	152000	FALSE	FALSE
7564	6400	15887	107500	FALSE	FALSE
7565	5625	15887	175000	FALSE	FALSE
7566	6500	15887	100000	FALSE	FALSE
7567	5120	15887	103000	FALSE	FALSE
7568	5610	15887	122500	FALSE	FALSE
7569	5632	15887	155000	FALSE	FALSE
7570	5000	15887	128500	FALSE	FALSE
7571	4800	15887	115000	FALSE	FALSE
7572	5120	15887	102000	FALSE	FALSE
7573	5060	15887	140000	FALSE	FALSE
7574	5000	15887	129000	FALSE	FALSE
7575	4800	15887	97000	FALSE	FALSE
7576	16099	15887	398900	FALSE	FALSE
7577	7800	15887	162000	FALSE	FALSE
7578	5080	15887	128500	FALSE	FALSE
7579	6480	15887	83500	FALSE	FALSE
7580	5500	15887	120000	FALSE	FALSE
7581	10752	15887	95000	FALSE	FALSE

7582	7095	15887	123500	FALSE	FALSE
7583	7714	15887	122500	FALSE	FALSE
7584	5248	15887	73000	FALSE	FALSE
7585	5000	15887	109000	FALSE	FALSE
7586	5280	15887	75000	FALSE	FALSE
7587	5000	15887	97000	FALSE	FALSE
7588	6750	15887	91500	FALSE	FALSE
7589	5400	15887	105500	FALSE	FALSE
7590	6016	15887	96000	FALSE	FALSE
7591	6000	15887	120000	FALSE	FALSE
7592	7370	15887	127000	FALSE	FALSE
7593	8352	15887	75000	FALSE	FALSE
7594	7686	15887	94000	FALSE	FALSE
7595	4995	15887	120000	FALSE	FALSE
7596	5000	15887	110000	FALSE	FALSE
7597	5280	15887	87000	FALSE	FALSE
7598	5000	15887	95000	FALSE	FALSE
7599	5000	15887	92900	FALSE	FALSE
7600	9000	15887	26000	FALSE	FALSE
7601	3600	15887	29000	FALSE	FALSE
7602	4800	15887	26000	FALSE	FALSE
7603	5868	15887	45000	FALSE	FALSE
7604	7550	15887	280000	FALSE	FALSE
7605	4720	15887	63000	FALSE	FALSE
7606	4920	15887	58500	FALSE	FALSE
7607	6300	15887	30000	FALSE	FALSE
7608	4360	15887	65000	FALSE	FALSE
7609	4590	15887	75000	FALSE	FALSE
7610	3300	15887	85000	FALSE	FALSE
7611	4500	15887	95200	FALSE	FALSE
7612	4480	15887	72000	FALSE	FALSE
7613	3600	15887	70000	FALSE	FALSE
7614	4760	15887	50000	FALSE	FALSE
7615	4800	15887	85500	FALSE	FALSE
7616	5400	15887	93000	FALSE	FALSE
7617	3600	15887	19900	FALSE	FALSE
7618	9000	15887	158500	FALSE	FALSE
7619	11072	15887	153500	FALSE	FALSE
7620	9440	15887	128000	FALSE	FALSE
7621	6000	15887	123000	FALSE	FALSE
7622	7200	15887	89900	FALSE	FALSE
7623	7200	15887	94000	FALSE	FALSE
7624	7245	15887	95000	FALSE	FALSE

7625	5782	15887	232000	FALSE	FALSE
7626	5500	15887	135000	FALSE	FALSE
7627	5200	15887	167500	FALSE	FALSE
7628	4800	15887	105000	FALSE	FALSE
7629	4320	15887	160900	FALSE	FALSE
7630	5560	15887	129900	FALSE	FALSE
7631	6336	15887	209000	FALSE	FALSE
7632	5000	15887	229000	FALSE	FALSE
7633	5116	15887	160000	FALSE	FALSE
7634	4800	15887	78000	FALSE	FALSE
7635	8400	15887	45000	FALSE	FALSE
7636	4800	15887	133000	FALSE	FALSE
7637	4700	15887	102000	FALSE	FALSE
7638	4520	15887	146000	FALSE	FALSE
7639	4800	15887	94000	FALSE	FALSE
7640	5355	15887	160000	FALSE	FALSE
7641	4800	15887	159900	FALSE	FALSE
7642	6280	15887	143900	FALSE	FALSE
7643	4520	15887	144000	FALSE	FALSE
7644	6250	15887	197000	FALSE	FALSE
7645	4800	15887	120000	FALSE	FALSE
7646	6750	15887	183385	FALSE	FALSE
7647	6375	15887	90000	FALSE	FALSE
7648	6375	15887	90000	FALSE	FALSE
7649	5805	15887	110000	FALSE	FALSE
7650	4284	15887	95000	FALSE	FALSE
7651	5460	15887	79500	FALSE	FALSE
7652	7938	15887	115000	FALSE	FALSE
7653	5360	15887	89000	FALSE	FALSE
7654	6000	15887	193900	FALSE	FALSE
7655	5880	15887	134000	FALSE	FALSE
7656	3700	15887	21000	FALSE	FALSE
7657	3720	15887	74000	FALSE	FALSE
7658	4880	15887	65000	FALSE	FALSE
7659	3870	15887	86000	FALSE	FALSE
7660	6002	15887	280000	FALSE	FALSE
7661	4800	15887	146500	FALSE	FALSE
7662	7200	15887	150000	FALSE	FALSE
7663	11440	15887	173400	FALSE	FALSE
7664	5280	15887	119900	FALSE	FALSE
7665	7095	15887	157000	FALSE	FALSE
7666	5160	15887	124000	FALSE	FALSE
7667	5400	15887	148000	FALSE	FALSE

7668	10000	15887	185000	FALSE	FALSE
7669	7200	15887	160000	FALSE	FALSE
7670	6360	15887	140000	FALSE	FALSE
7671	8308	15887	165000	FALSE	FALSE
7672	5942	15887	120000	FALSE	FALSE
7673	6500	15887	70500	FALSE	FALSE
7674	5808	15887	87900	FALSE	FALSE
7676	7920	15887	145000	FALSE	FALSE
7677	4944	15887	126000	FALSE	FALSE
7678	7198	15887	168500	FALSE	FALSE
7679	5040	15887	110000	FALSE	FALSE
7680	8083	15887	125000	FALSE	FALSE
7681	6555	15887	122000	FALSE	FALSE
7682	7560	15887	118500	FALSE	FALSE
7683	6384	15887	120000	FALSE	FALSE
7684	7560	15887	143000	FALSE	FALSE
7685	7259	15887	155000	FALSE	FALSE
7686	5500	15887	130000	FALSE	FALSE
7687	6700	15887	130000	FALSE	FALSE
7688	6350	15887	97500	FALSE	FALSE
7689	5350	15887	123900	FALSE	FALSE
7690	4800	15887	77000	FALSE	FALSE
7691	7200	15887	177500	FALSE	FALSE
7692	3600	15887	86000	FALSE	FALSE
7693	3640	15887	38000	FALSE	FALSE
7694	12400	15887	217000	FALSE	FALSE
7695	8460	15887	152000	FALSE	FALSE
7696	7560	15887	154000	FALSE	FALSE
7697	6450	15887	135000	FALSE	FALSE
7698	5805	15887	114000	FALSE	FALSE
7699	7950	15887	135000	FALSE	FALSE
7700	6612	15887	133500	FALSE	FALSE
7701	4455	15887	152500	FALSE	FALSE
7702	5040	15887	134900	FALSE	FALSE
7703	7290	15887	144900	FALSE	FALSE
7704	16585	15887	147000	FALSE	FALSE
7705	4440	15887	147000	FALSE	FALSE
7706	4800	15887	123000	FALSE	FALSE
7707	4752	15887	107900	FALSE	FALSE
7708	7875	15887	117000	FALSE	FALSE
7709	6292	15887	147901	FALSE	FALSE
7710	6500	15887	150000	FALSE	FALSE
7711	11016	15887	139000	FALSE	FALSE

```
7712    4589     15887     105000 FALSE FALSE
7713    7504     15887     128000 FALSE FALSE
7714    7881     15887     131000 FALSE FALSE
7715    6500     15887     120000 FALSE FALSE
7716    8103     15887     100000 FALSE FALSE
7717    4880     15887      75000 FALSE FALSE
7718    4680     15887      67000 FALSE FALSE
7719    6300     15887      99000 FALSE FALSE
7720    6534     15887     136000 FALSE FALSE
7721    5074     15887     150000 FALSE FALSE
7722    8996     15887     150000 FALSE FALSE
7723    5520     15887     166900 FALSE FALSE
7724    3690     15887     155000 FALSE FALSE
7725    5828     15887     156000 FALSE FALSE
7726    5280     15887     127500 FALSE FALSE
7727    3700     15887     190000 FALSE  TRUE
7728    4920     15887     176000 FALSE FALSE
7729    3600     15887     238000 FALSE  TRUE
7730    6500     15887     168500 FALSE FALSE
7731    5130     15887     163000 FALSE FALSE
7732    2400     15887     171000 FALSE FALSE
7733    3600     15887     152000 FALSE FALSE
7734    4000     15887     219500 FALSE  TRUE
7735    4560     15887     242000 FALSE FALSE
7736    5200     15887     224500 FALSE FALSE
7737    4920     15887     230500 FALSE FALSE
7738    3900     15887     253500 FALSE FALSE
7739    4680     15887     233500 FALSE FALSE
7740    3750     15887     174500 FALSE  TRUE
7741    4080     15887     299000 FALSE FALSE
7742    3000     15887     206000 FALSE FALSE
7743    5616     15887     210000 FALSE FALSE
7744    3045     15887     165000 FALSE FALSE
7745   18480     15887     210000 FALSE FALSE
7746    3270     15887     285000 FALSE  TRUE
7747    6400     15887     120000 FALSE FALSE
7748    3399     15887     120000 FALSE FALSE
7749    3600     15887      94900 FALSE FALSE
 [ reached 'max' / getOption("max.print") -- omitted 16751 rows ]
```

```r
model2=lm(sqrt(Sale_price)~District +  Extwall +     Stories   + Year_Built+   District*Yea
            Fbath  +    log(Lotsize)   + Sale_date +District* log(Lotsize),df_clean4)
# model2=lm(Sale_price^(0.4)~.,df_clean3)


summ2=summary(model2); summ2
```

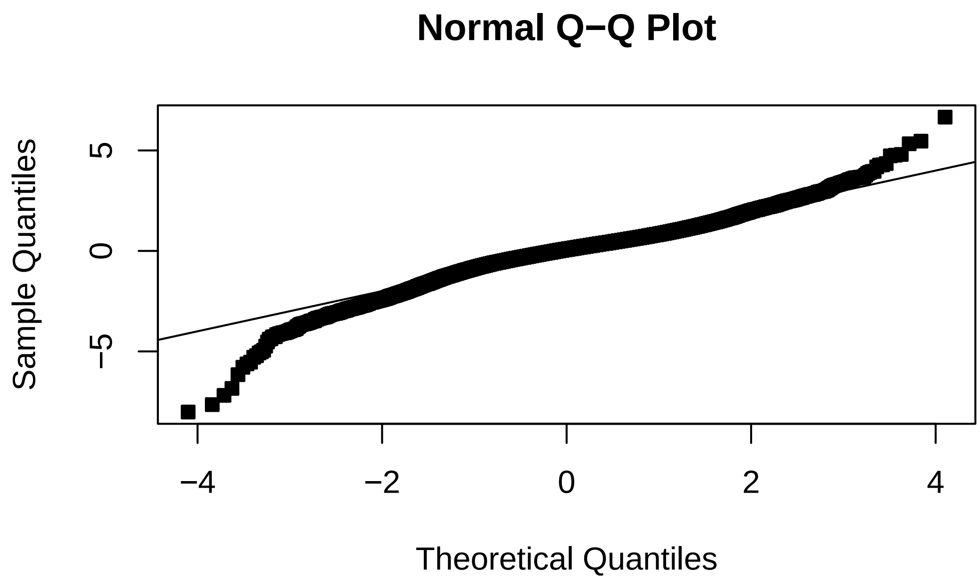
```
Call:
lm(formula = sqrt(Sale_price) ~ District + Extwall + Stories +
    Year_Built + District * Year_Built + Fin_sqft + Units + Bdrms +
    Fbath + log(Lotsize) + Sale_date + District * log(Lotsize),
    data = df_clean4)

Residuals:
    Min      1Q  Median      3Q     Max
-547.36  -35.61    5.89   41.47  474.14

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.396e+03  9.630e+01  14.494  < 2e-16 ***
District               -2.441e+02  9.158e+00 -26.656  < 2e-16 ***
ExtwallBlock           -6.108e+00  6.044e+00  -1.011 0.312259
ExtwallBrick            1.146e+01  1.183e+00   9.684  < 2e-16 ***
ExtwallFiber-Cement     2.442e+01  6.115e+00   3.993 6.55e-05 ***
ExtwallFrame           -1.069e+01  1.599e+00  -6.682 2.41e-11 ***
ExtwallMasonry / Frame  8.342e+00  2.797e+00   2.983 0.002858 **
ExtwallPrem Wood        1.492e+01  9.195e+00   1.622 0.104761
ExtwallStone            9.094e+00  2.497e+00   3.641 0.000272 ***
ExtwallStucco           1.537e+01  3.511e+00   4.378 1.20e-05 ***
Stories1                4.059e+01  1.680e+01   2.416 0.015681 *
Stories1.5              5.480e+01  1.677e+01   3.268 0.001084 **
Stories2                6.143e+01  1.670e+01   3.678 0.000236 ***
Year_Built             -9.488e-01  5.316e-02 -17.847  < 2e-16 ***
Fin_sqft                9.479e-02  1.565e-03  60.560  < 2e-16 ***
Units1                  1.238e+02  1.200e+01  10.312  < 2e-16 ***
Units2                  2.178e+01  1.201e+01   1.813 0.069845 .
Units3                 -2.191e+01  1.293e+01  -1.695 0.090108 .
Bdrms0                  1.493e+02  3.025e+01   4.936 8.03e-07 ***
Bdrms1                  1.229e+02  1.648e+01   7.455 9.28e-14 ***
Bdrms2                  1.344e+02  1.472e+01   9.133  < 2e-16 ***
Bdrms3                  1.393e+02  1.462e+01   9.524  < 2e-16 ***
```

```
Bdrms4                  1.213e+02  1.456e+01   8.327  < 2e-16 ***
Bdrms5                  1.187e+02  1.456e+01   8.151 3.78e-16 ***
Bdrms6                  9.863e+01  1.459e+01   6.762 1.39e-11 ***
Bdrms7                  6.771e+01  1.555e+01   4.353 1.35e-05 ***
Bdrms8                  9.538e+01  1.636e+01   5.829 5.65e-09 ***
Fbath0                 -6.106e+01  2.227e+01  -2.742 0.006118 **
Fbath1                 -3.703e+01  1.553e+01  -2.385 0.017096 *
Fbath2                 -1.015e+01  1.543e+01  -0.658 0.510580
Fbath3                  3.187e+01  1.532e+01   2.080 0.037538 *
Fbath4                  6.184e+01  1.643e+01   3.764 0.000168 ***
log(Lotsize)            3.120e+01  3.416e+00   9.133  < 2e-16 ***
Sale_date               5.995e-03  4.263e-04  14.064  < 2e-16 ***
District:Year_Built     1.315e-01  5.309e-03  24.762  < 2e-16 ***
District:log(Lotsize)  -8.262e-01  3.469e-01  -2.382 0.017230 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 71.77 on 24407 degrees of freedom
Multiple R-squared:  0.4635,    Adjusted R-squared:  0.4627
F-statistic: 602.4 on 35 and 24407 DF,  p-value: < 2.2e-16
```

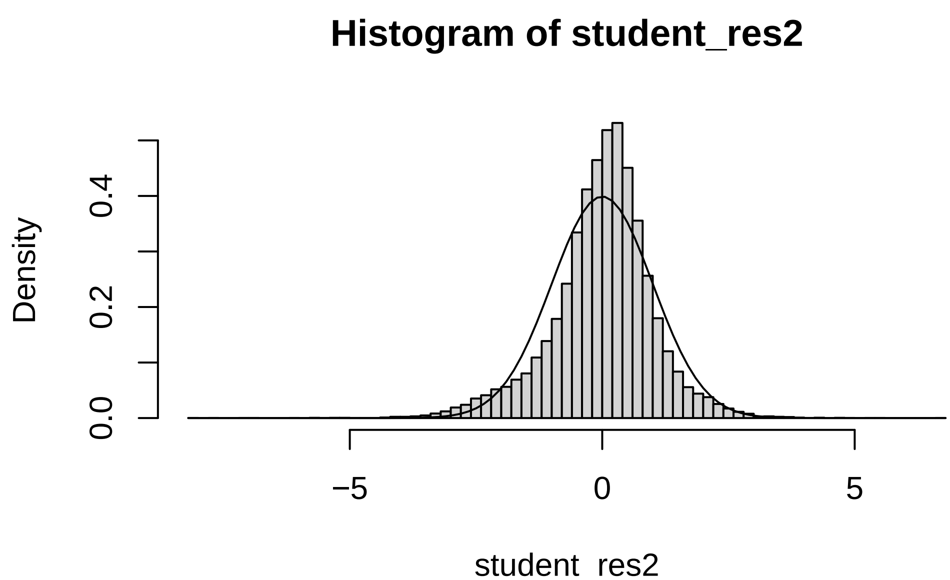
```r
summ2$adj.r.squared
```

```
[1] 0.4626948
```

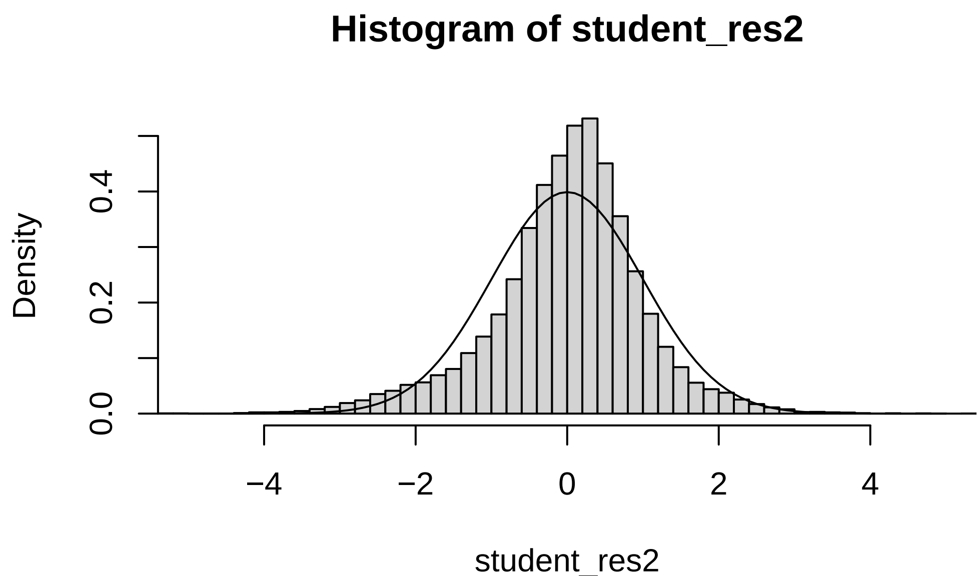
```r
student_res2=rstudent(model2)

MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



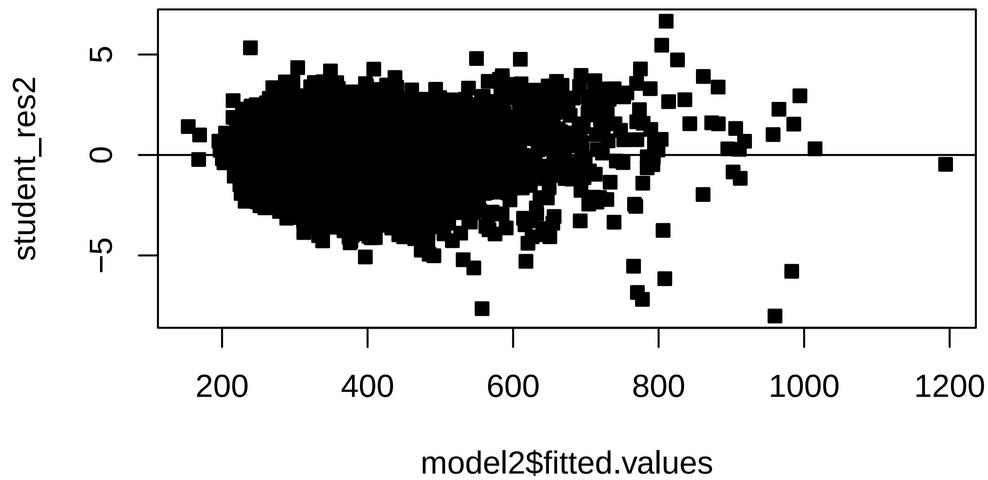

```r
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



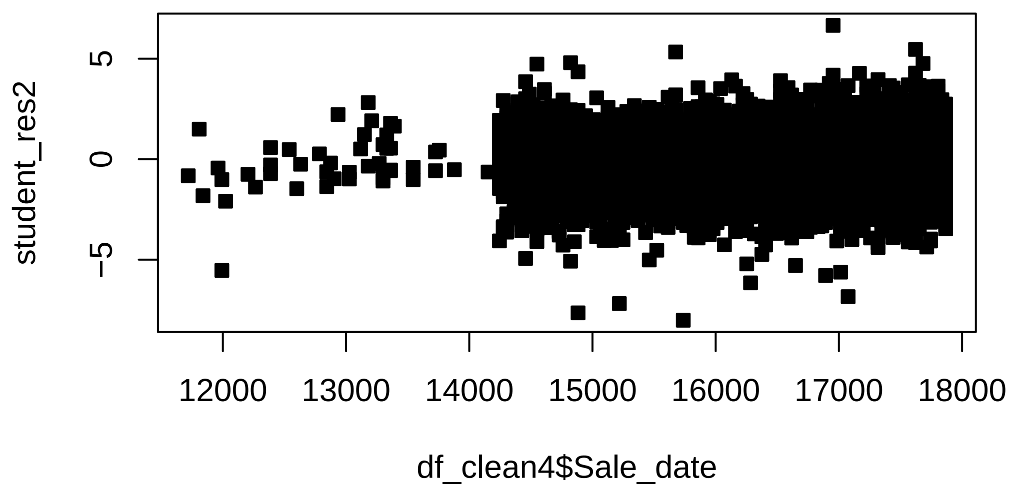
```
hist(student_res2,freq=F,xlim=c(-5,5),breaks=100)
curve(dnorm(x,0,1),add=T)
```



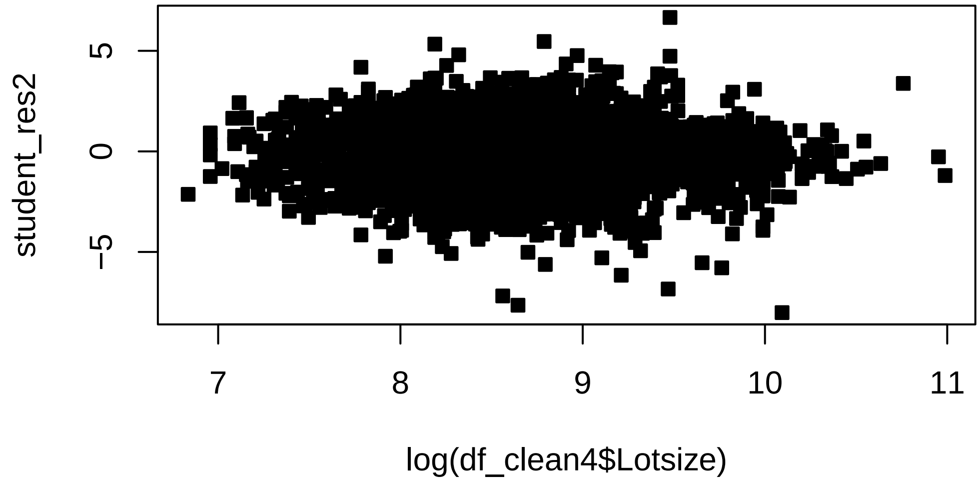

```
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```



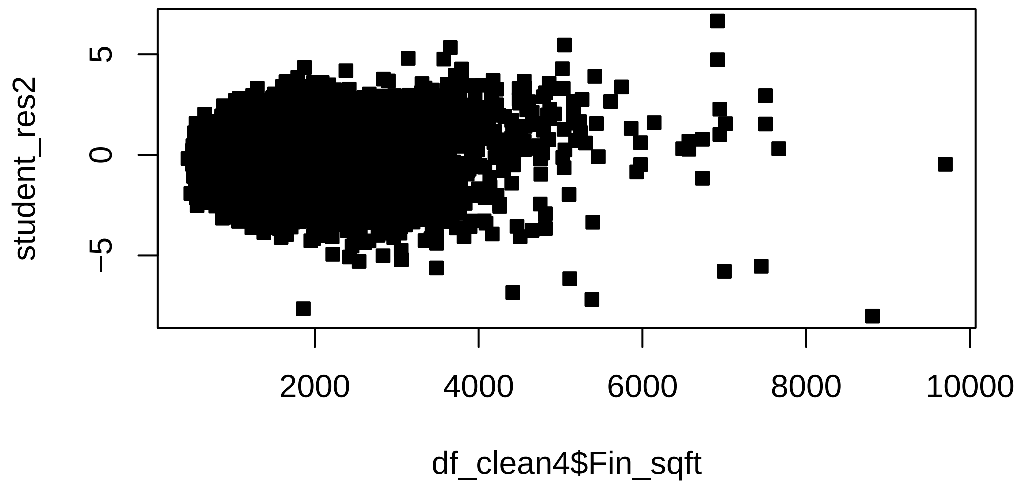

```r
plot(df_clean4$Sale_date ,student_res2,pch=22,bg=1)
```



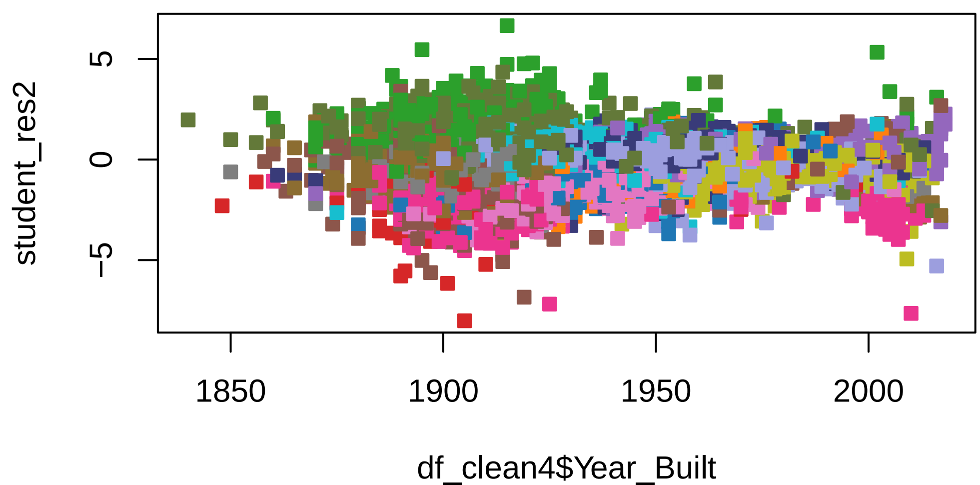
```r
plot(log(df_clean4$Lotsize) ,student_res2,pch=22,bg=1)
```




```r
plot(df_clean4$Fin_sqft ,student_res2,pch=22,bg=1)
```




```
#Year built is probably nonlinear...
plot(df_clean4$Year_Built ,student_res2,pch=22,bg=custom_palette[df_clean4$District],col=c
```



6.5 Homework questions

Exercise 6.2. Consider a ML regression model for bread quality against bake time and 3 types of yeast (A, B and C). Write out the dummy variables for the variable yeast type. What is the interpretation of the coefficient of each of the dummy variables, in this context?

Exercise 6.3. Suppose we regress real estate price against number of bathrooms. What is the difference in interpretation between representing number of bedrooms with dummy variables versus a continuous variable?

Exercise 6.4. Consider a ML regression model for bread quality against bake time and 3 types of yeast (A, B and C). Write out the regression equation that includes an interaction between yeast and bake time. What is the interpretation of the coefficient for each of the interaction effects? Compare and contrast the regression model in Exercise 6.2 to this one.

Complete the Chapter 8 questions.

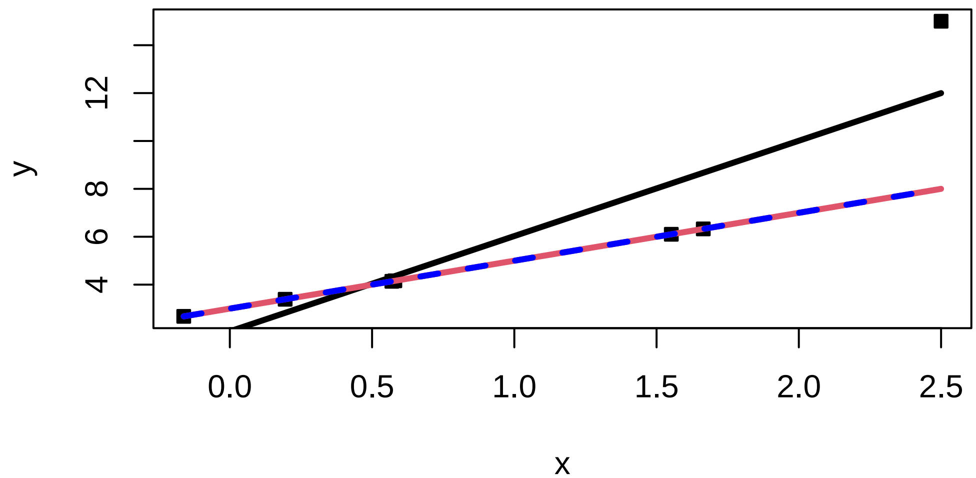
7 Leverage and Influence

7.1 Influential observations and leverage

Recall that violations of model assumptions are more likely at remote points, and these violations may be hard to detect from inspection of the ordinary residuals because their residuals will usually be smaller. Points that are outlying in the x-direction are known as **leverage points**. **Influential points** are not only remote in terms of the specific values for the regressors, but the observed response is not consistent with the values that would be predicted based on only the other data points. It is important to find these influential points and assess their impact on the model.

Below gives an example of an influential point. The seventh point in the data set is outlying in the x-direction, and it's response value is not consistent with the regression line based on the other six observations:

```
set.seed(330)
x=c(rnorm(6),2.5)
y=x*2+3
y[7]=y[7]+7
plot(x,y,pch=22,bg=1)
a=lm(y~x)
curve(a$coefficients[1]+x*a$coefficients[2],add=T,lwd=3)
curve(x*2+3,add=T,col=2,lwd=3)

a2=lm(y[-7]~x[-7])
curve(a2$coefficients[1]+x*a2$coefficients[2],add=T,lwd=3,col='blue',lty=2)
```




```r
a$coefficients
```

```
(Intercept)           x
   2.048937    3.979977
```

Sometimes we find that a regression coefficient may have a sign that does not make engineering or scientific sense, a regressor known to be important may be statistically insignificant, or a model that fits the data well and that is logical from an application – environment perspective may produce poor predictions. These situations may be the result of one or, perhaps, a few influential observations.

Recall the hat matrix $H = X(X^\top X)^{-1}X^\top$, as well as that it holds that $\mathrm{Var}\left[\hat{\epsilon}\right] = \sigma^2(I - H)$ and $\mathrm{Var}\left[\hat{Y}\right] = \sigma^2 H$. Note that h_{ij} can be interpreted as the amount of leverage exerted by the ith observation y_i on the jth fitted value $\hat{y}_j$. We usually focus attention on the diagonal elements h_{ii} of the hat matrix H, which may be written as

$$h_{ii} = x_i^\top (X^\top X)^{-1} x_i,$$

where $X_i^\top$ is the ith row of X. The hat matrix diagonal is a standardized measure of the distance of the ith observation from the center (or centroid) of the x-space. Therefore, large values of h_{ii} implies that x_i is potentially influential. Furthermore, note that $rank(H) = p$ since the trace of an idempotent matrix equals its rank, which means that $\bar{h} = p/n$. It follows that values well above p/n, say $h_{ii} > 2p/n$, can be called leverage points.

```r
X=as.matrix(cbind(rep(1,length(x)),x))
# or

X=model.matrix(a)
hat=X%*%solve(t(X)%*%X)%*%t(X)

diag(hat)
```

```
        1         2         3         4         5         6         7 
0.2027453 0.2288737 0.2596869 0.1751432 0.1735495 0.3887329 0.5712686 
```

```r
p=2
n=7
diag(hat)>2*p/n
```

```
    1     2     3     4     5     6     7 
FALSE FALSE FALSE FALSE FALSE FALSE FALSE 
```

7.2 Cook's Distance

Cook's Distance is one way to incorporate both the X and Y values into an outlyingness measure:

$$D_i\left(X^\top X, p, MSE\right) \equiv D_i = \frac{\left(\widehat{\beta}_{(i)} - \widehat{\beta}\right)^\top X^\top X\left(\widehat{\beta}_{(i)} - \widehat{\beta}\right)}{pMSE}, \ i = \in [n],$$

where $\widehat{\beta}_{(i)}$ is the OLS estimator with the ith point removed.Large values of Cook's distance signal a leverage point.

What do we mean by a large value? We can compare D_i to the 50th percentile of the $F_{p,n-p}$ distribution. This gives the interpretation that deleting the ith point moves the estimate to the boundary of a 50% confidence interval. $F_{p,n-p} \approx 1$, and so usually take $D_i \geq 1$ to be large.

Observe that

$$D_i = \frac{r_i^2}{p}\frac{\mathrm{Var}\left(\widehat{Y}_i\right)}{\mathrm{Var}\left(\hat{\epsilon}_i\right)} = \frac{r_i^2}{p}\frac{h_{ii}}{1-h_{ii}}, \quad i = 1, 2, \ldots, n,$$

where it is important to recall that r_i is the studentized residual. Now, the quantity $\frac{h_{ii}}{1-h_{ii}}$ can be shown to be the distance from the vector x_i to the centroid of the remaining data.

Therefore, D_i is the product of outlyingness in both the X and Y directions. We may also write D_i as

$$D_i = \frac{\left\|\hat{y}_{(i)} - \hat{y}\right\|^2}{pMSE},$$

which allows for the interpretation: The Cook's distance of the ith point is the normalized distance between the fitted value with and without point i.

```
#cut off
cooks.distance(a)
```

```
           1            2            3            4            5            6 
0.1708029420 0.2516095165 0.0180669722 0.0009569213 0.0011772793 0.2002829110 
           7 
3.3311562309 
```

```
cooks.distance(a)>1
```

```
    1     2     3     4     5     6     7 
FALSE FALSE FALSE FALSE FALSE FALSE  TRUE 
```

```
df=data.frame(cbind(y,x))
df[cooks.distance(a)>1,]
```

```
   y   x
7 15 2.5
```

7.3 Data depth functions

A more modern approach and nonparametric approach to outlier detection is through data depth. A data depth function gives meaning to centrality, order and outlyingness in spaces beyond $\mathbb{R}$. A data depth function is a function which takes a sample and a point, and returns how central the point is, with respect to the sample. Depth functions can be written as $\mathrm{D}\colon \mathbb{R}^d \times \text{Sample} \to \mathbb{R}^+$. There are different definitions of depth, so I will give a few.

Let $S^{d-1} = \{x \in \mathbb{R}^d \colon \|x\| = 1\}$ be the set of unit vectors in $\mathbb{R}^d$, let $\mathbb{X}_n = \{(Y_1, X_{1,1}, \dots, X_{1,p-1}), \dots, (Y_n, X_{n,1}, \dots, X$ let $\mathbb{X}_n^\top u$ be $\mathbb{X}_n$ projected onto $u \in S^{d-1}$ and let $\widehat{F}_u$ be the empirical CDF with respect to $\mathbb{X}_n^\top u$.

The halfspace depth D_H of a point $x \in \mathbb{R}^d$ with respect to a distribution F over $\mathbb{R}^d$ is

$$\mathrm{D}_H(x; F) = \inf_{u \in S^{d-1}} \widehat{F}_u(x^\top u) \wedge (1 - F_u(x^\top u)) = \inf_{u \in S^{d-1}} F_u(x^\top u).$$

Given a translation and scale equivariant location estimate μ and a translation and scale invariant scale estimate σ, the outlyingness at $x \in \mathbb{R}^d$ is defined as

$$O(x) = \sup_{u \in S^{d-1}} \frac{\left|x^\top u - \mu(\mathbb{X}_n^\top u)\right|}{\sigma(\mathbb{X}_n^\top u)}.$$

Define projection depth as

$$\mathrm{D}_p(x) = (1 + O(x))^{-1}.$$

In order to detect outliers, we look for observations that have low depth. See, continuing our toy example:

```
# install.packages('ddalpha')
depths=ddalpha::depth.projection(cbind(x,y),cbind(x,y))
depths
```

```
[1] 0.276409011 0.255272074 0.500000000 0.973754328 0.973046927 0.338954415
[7] 0.001695292
```

```
depths<0.015
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

Example 7.1. Recall example Example 6.6. Check for leverage and influential points in the proposed models. Compute all three measures of leverage/influence/outlyingness introduced in this lesson. What do you find?

I will load in the data below:

We can now analyze the data:

```
# df=df[df$Lotsize<70000,]


custom_palette <- c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
```

```r
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
  "#bcbd22", "#17becf", "#393b79",
  "#8c6d31", "#9c9ede", "#637939", "#eb348f"
)

# Our model from the previous lecture
df=df_clean2[-which.max(df_clean2$Lotsize),]
df=df[df$Lotsize>0,]
df['district_3']=df['District']==3
df['district_4']=df['District']==4
df['district_15']=df['District']==15
model2=lm(Sale_price~.-district_3-district_4-district_15+district_3*Year_Built+district_3*


# Compute residuals
student_res2=rstudent(model2)


summ2=summary(model2); summ2
```

```
Call:
lm(formula = Sale_price ~ . - district_3 - district_4 - district_15 +
    district_3 * Year_Built + district_3 * Lotsize + district_4 *
    Lotsize + district_4 * Year_Built + district_15 * Year_Built +
    district_15 * Lotsize + district_3 * Fin_sqft + district_4 *
    Fin_sqft + district_15 * Fin_sqft, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-303393  -24422    -942   23352  719561

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1.129e+06  3.625e+04 -31.143  < 2e-16 ***
District                    5.695e+03  7.701e+01  73.950  < 2e-16 ***
ExtwallBlock               -3.944e+03  3.688e+03  -1.069 0.284879
ExtwallBrick                6.725e+03  7.245e+02   9.281  < 2e-16 ***
ExtwallFiber-Cement         3.847e+04  3.754e+03  10.246  < 2e-16 ***
ExtwallFrame               -4.986e+03  9.786e+02  -5.095 3.51e-07 ***
ExtwallMasonry / Frame      3.751e+03  1.711e+03   2.192 0.028404 *
ExtwallPrem Wood            1.894e+04  5.596e+03   3.385 0.000714 ***
```

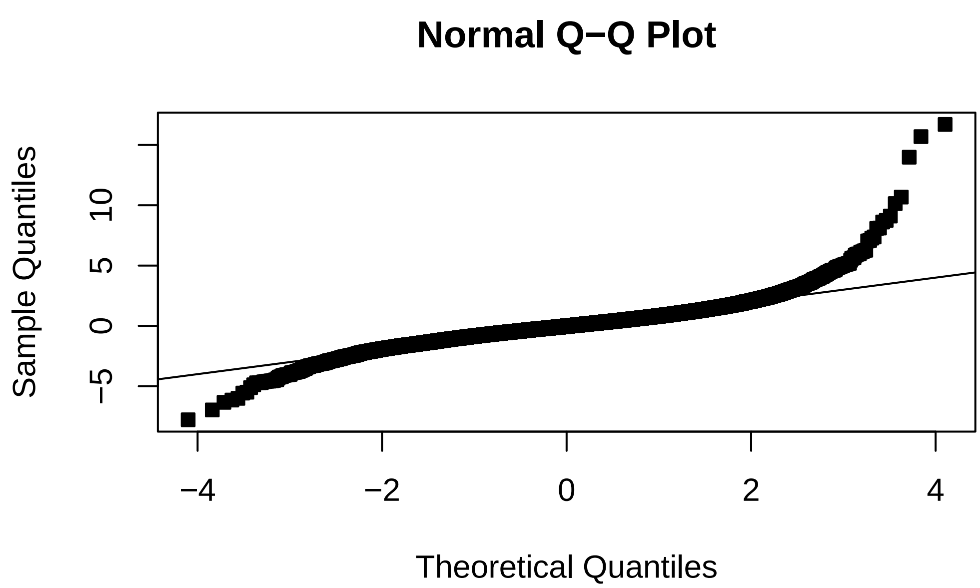
```
ExtwallStone               1.162e+04  1.531e+03   7.594 3.22e-14 ***
ExtwallStucco              3.514e+03  2.149e+03   1.635 0.102070    
Stories1                   3.684e+02  1.047e+04   0.035 0.971935    
Stories1.5                 1.317e+04  1.046e+04   1.259 0.207954    
Stories2                   1.451e+04  1.042e+04   1.392 0.163802    
Year_Built                 4.629e+02  1.606e+01  28.823  < 2e-16 ***
Fin_sqft                   6.048e+01  1.080e+00  55.986  < 2e-16 ***
Units1                     5.778e+04  7.389e+03   7.820 5.48e-15 ***
Units2                    -9.774e+03  7.386e+03  -1.323 0.185735    
Units3                    -4.087e+04  7.940e+03  -5.147 2.67e-07 ***
Bdrms0                     1.155e+05  1.854e+04   6.227 4.82e-10 ***
Bdrms1                     7.894e+04  1.016e+04   7.766 8.41e-15 ***
Bdrms2                     8.331e+04  9.086e+03   9.170  < 2e-16 ***
Bdrms3                     8.791e+04  9.033e+03   9.733  < 2e-16 ***
Bdrms4                     7.905e+04  9.001e+03   8.782  < 2e-16 ***
Bdrms5                     7.507e+04  9.002e+03   8.340  < 2e-16 ***
Bdrms6                     6.149e+04  9.006e+03   6.828 8.80e-12 ***
Bdrms7                     2.462e+04  9.572e+03   2.573 0.010100 *  
Bdrms8                     1.242e+04  1.008e+04   1.233 0.217672    
Fbath0                    -1.936e+04  1.388e+04  -1.395 0.162964    
Fbath1                    -1.050e+04  9.795e+03  -1.072 0.283695    
Fbath2                     7.289e+03  9.751e+03   0.748 0.454721    
Fbath3                     2.935e+04  9.641e+03   3.044 0.002339 ** 
Fbath4                     6.757e+04  1.025e+04   6.595 4.35e-11 ***
Lotsize                    1.419e+00  1.131e-01  12.545  < 2e-16 ***
Sale_date                  4.839e+00  2.603e-01  18.587  < 2e-16 ***
district_3TRUE             9.189e+05  1.373e+05   6.693 2.23e-11 ***
district_4TRUE             1.088e+06  2.504e+05   4.346 1.39e-05 ***
district_15TRUE            4.475e+05  1.212e+05   3.692 0.000223 ***
Year_Built:district_3TRUE  -5.100e+02  7.205e+01  -7.078 1.51e-12 ***
Lotsize:district_3TRUE     1.227e+01  4.803e-01  25.549  < 2e-16 ***
Lotsize:district_4TRUE    -3.796e+00  1.569e+00  -2.420 0.015512 *  
Year_Built:district_4TRUE  -5.558e+02  1.306e+02  -4.255 2.09e-05 ***
Year_Built:district_15TRUE -2.829e+02  6.295e+01  -4.493 7.05e-06 ***
Lotsize:district_15TRUE    3.140e+00  1.213e+00   2.589 0.009620 ** 
Fin_sqft:district_3TRUE    6.621e+01  1.517e+00  43.639  < 2e-16 ***
Fin_sqft:district_4TRUE   -2.049e+01  4.181e+00  -4.901 9.61e-07 ***
Fin_sqft:district_15TRUE  -1.161e+01  3.163e+00  -3.672 0.000241 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43870 on 24429 degrees of freedom
Multiple R-squared:  0.7301,    Adjusted R-squared:  0.7296
```

```
F-statistic:  1468 on 45 and 24429 DF,  p-value: < 2.2e-16
```

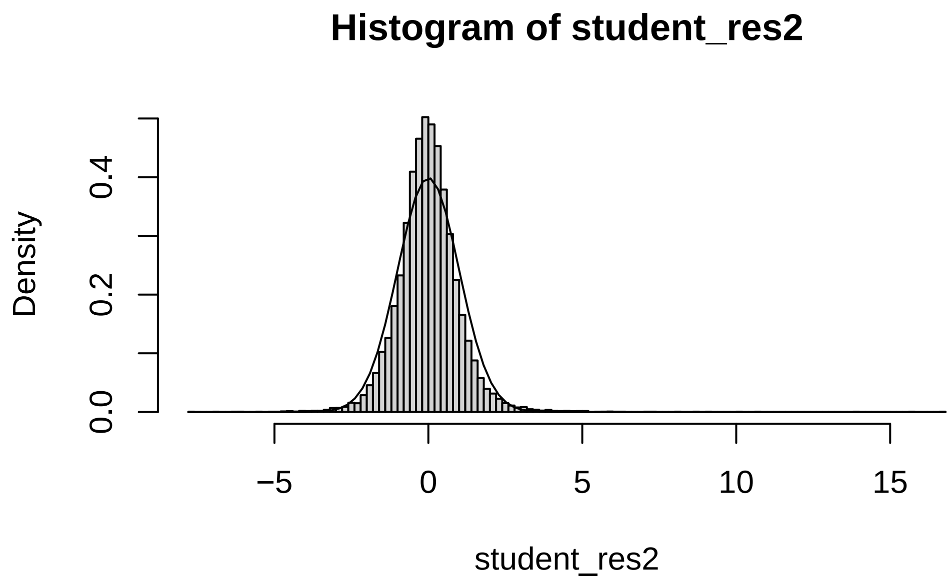
```
summ2$adj.r.squared
```

```
[1] 0.7295574
```

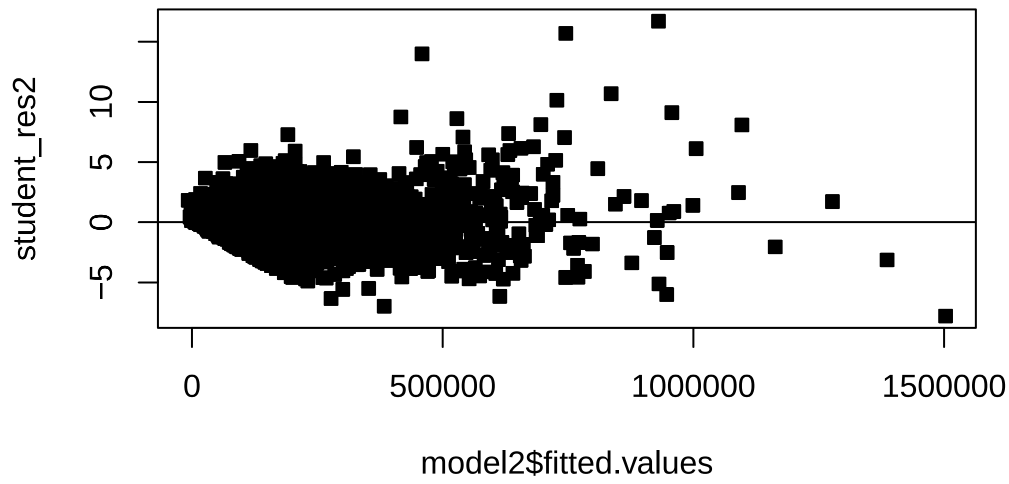
```
# Compute residual analysis


MSE2=summ2$sigma^2
qqnorm(student_res2,pch=22,bg=1)
abline(0,1)
```



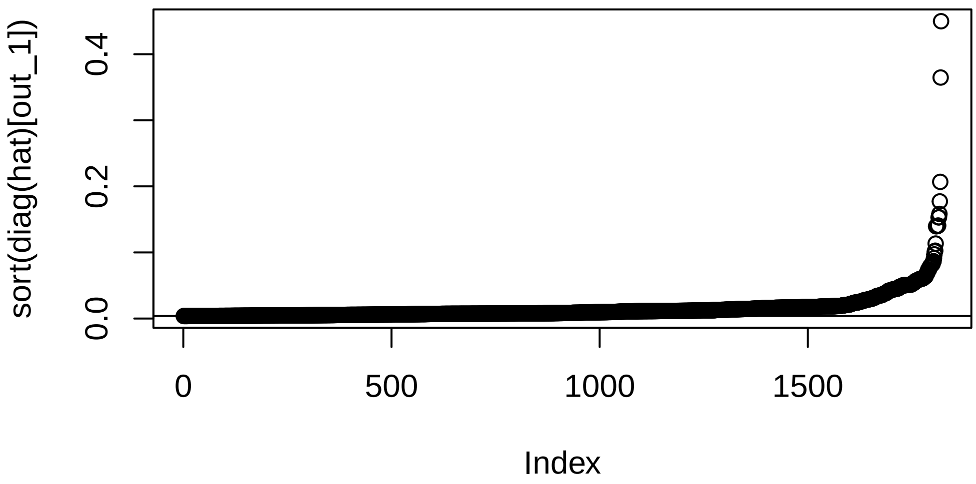

```
hist(student_res2,freq=F,breaks=100)
curve(dnorm(x,0,1),add=T)
```



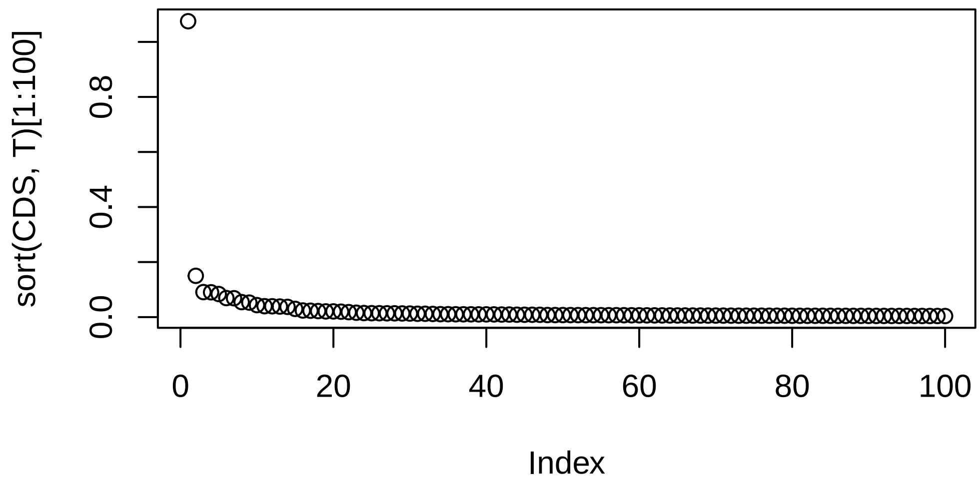

```
# hist(student_res2,freq=F,breaks=100)
# curve(dnorm(x,0,1),add=T)
plot(model2$fitted.values,student_res2,pch=22,bg=1)
abline(h=0)
```




```
# First measure

X=model.matrix(model2)
hat=X%*%solve(t(X)%*%X)%*%t(X)

# diag(hat)
p=ncol(X)
n=nrow(X)
out_1=which(diag(hat)>2*p/n)
plot(sort(diag(hat)[out_1]))
abline(h=2*p/n)
```




```
# I would still look at those after the elbow



# Cooks distances
CDS=cooks.distance(model2)
plot(sort(CDS,T)[1:100])
```




```r
which(CDS>1)
```

```
22532
22398
```

```r
max(CDS)
```

```
[1] 1.075144
```

```r
df[CDS>1,]
```

```
      District Extwall Stories Year_Built Fin_sqft Units Bdrms Fbath Lotsize
22532        3   Block       1       1960     4323     1     4     3   72480
      Sale_date Sale_price District District District
22532     17713    1250000     TRUE    FALSE    FALSE
```

```r
# I would still look at those two values that are far from the other distances
# I would still look at those before the elbow
```

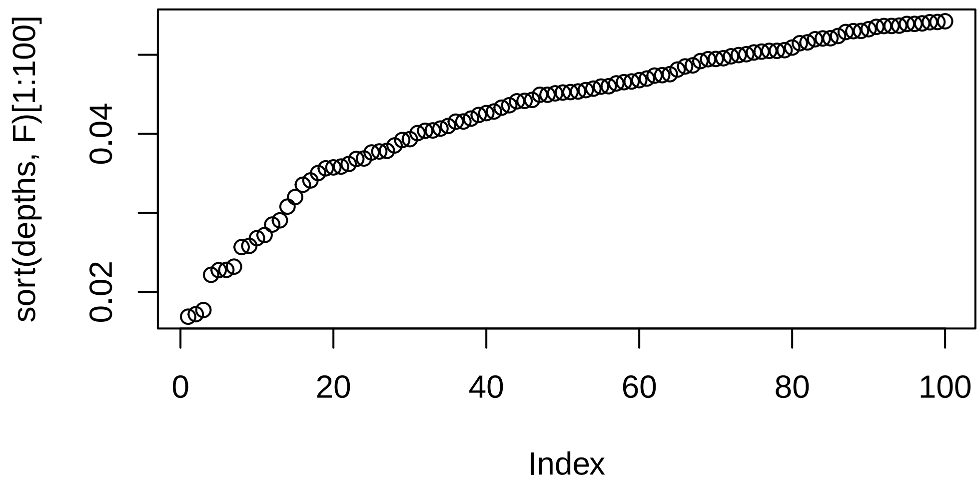
```r
# We may only look at numeric values for depth functions - so we can either
numer=NULL
for(i in names(df)){
  if(!is.factor(df[1,i])){
    numer=c(numer,i)
  }
}
numer
```

```
[1] "District"    "Year_Built"  "Fin_sqft"    "Lotsize"     "Sale_date"
[6] "Sale_price"  "district_3"  "district_4"  "district_15"
```

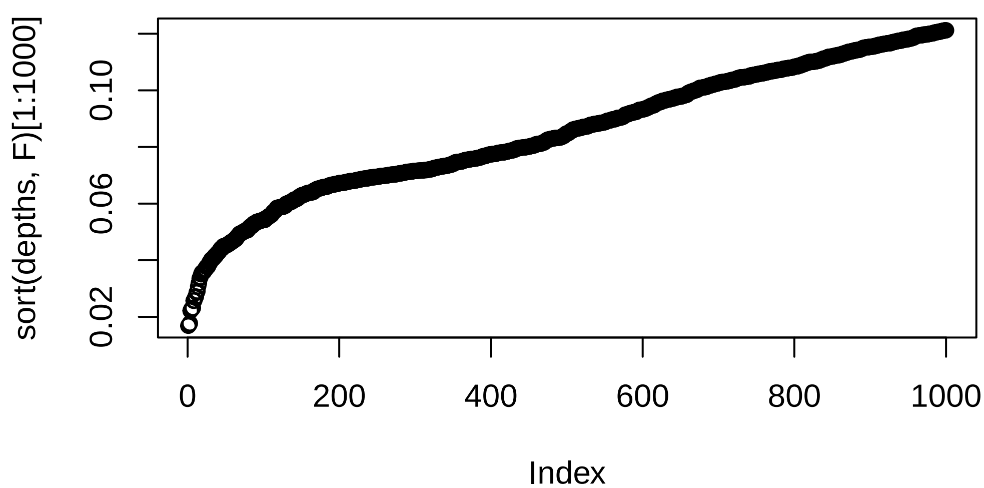
```r
df_mat=as.matrix(df[,numer])
depths=ddalpha::depth.projection(df_mat,df_mat)
which(depths<0.1)[1:10]
```

```
 [1]   5  64 130 326 345 471 473 593 637 669
```

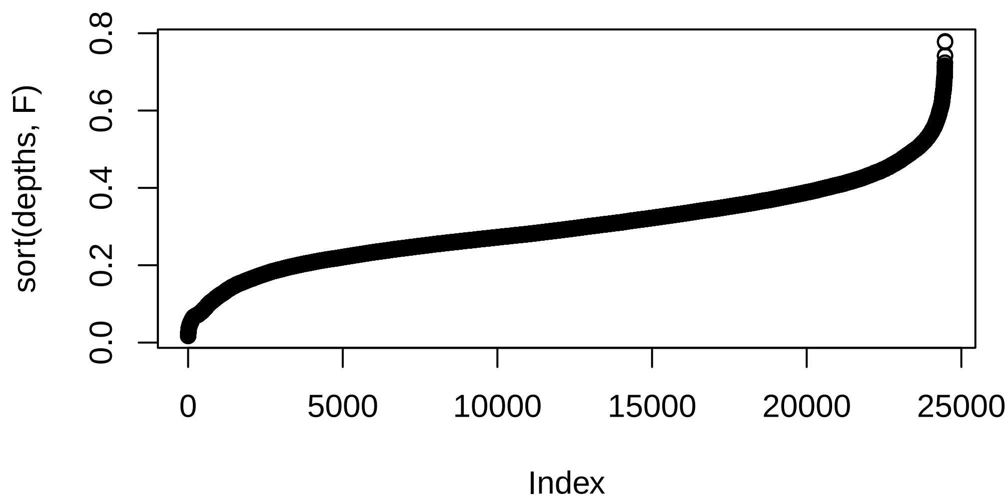
```r
plot(sort(depths,F)[1:100])
```




```
# Notice there is a crack around 0.035, I would look at those observations
plot(sort(depths,F)[1:1000])
```



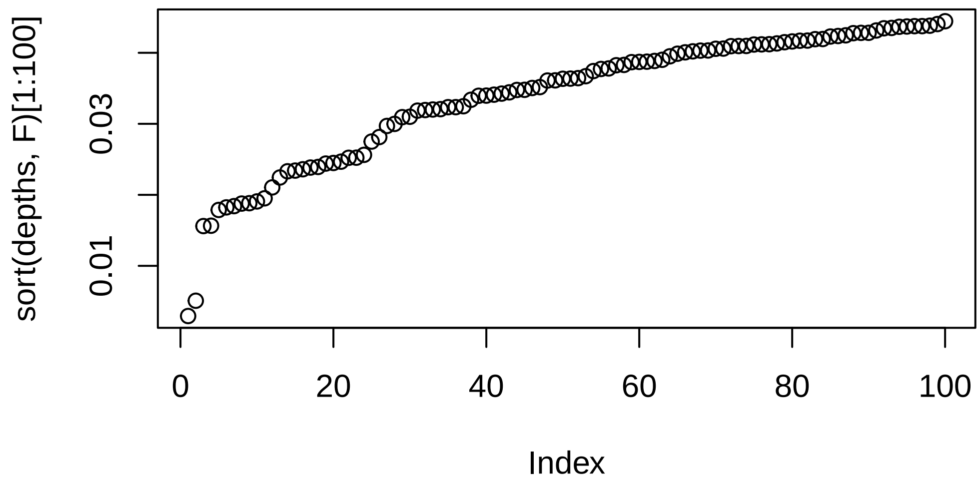
```r
plot(sort(depths,F))
```




```r
# OR
depths=ddalpha::depth.projection(cbind(X,df$Sale_price),cbind(X,df$Sale_price))
which(depths<0.1)[1:10]
```

```
[1]  3  4  5  7 15 16 17 20 22 26
```

```r
plot(sort(depths,F)[1:100])
```




```
which.max(diag(hat))
```

```
22532
22398
```

```
which.max(CDS)
```

```
22532
22398
```

```
which.min(depths)
```

```
[1] 22398
```

```
# Hugely expensive home!
df[which.min(depths),]
```

```
      District Extwall Stories Year_Built Fin_sqft Units Bdrms Fbath Lotsize
22532        3   Block       1       1960     4323     1     4     3   72480
      Sale_date Sale_price District District District
22532     17713    1250000     TRUE    FALSE    FALSE
```

```r
df[which.max(CDS),]
```

```
      District Extwall Stories Year_Built Fin_sqft Units Bdrms Fbath Lotsize
22532        3   Block       1       1960     4323     1     4     3   72480
      Sale_date Sale_price District District District
22532     17713    1250000     TRUE    FALSE    FALSE
```

```r
df[which.max(diag(hat)),]
```

```
      District Extwall Stories Year_Built Fin_sqft Units Bdrms Fbath Lotsize
22532        3   Block       1       1960     4323     1     4     3   72480
      Sale_date Sale_price District District District
22532     17713    1250000     TRUE    FALSE    FALSE
```

```r
model3=lm(Sale_price~.-district_3-district_4-district_15+district_3*Year_Built+district_3*

# OR

model4=lm(Sale_price~.-district_3-district_4-district_15+district_3*Year_Built+district_3*

# Compare
s=summary(model3)
summary(model3)
```

```
Call:
lm(formula = Sale_price ~ . - district_3 - district_4 - district_15 +
    district_3 * Year_Built + district_3 * Lotsize + district_4 *
    Lotsize + district_4 * Year_Built + district_15 * Year_Built +
    district_15 * Lotsize + district_3 * Fin_sqft + district_4 *
    Fin_sqft + district_15 * Fin_sqft, data = df[-(order(depths)[1:100]),
    ])

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-275556  -24244   -1138   22988  320067

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1.048e+06  3.556e+04 -29.483  < 2e-16 ***
District                    5.685e+03  7.266e+01  78.235  < 2e-16 ***
ExtwallBlock               -2.286e+03  3.485e+03  -0.656 0.511872
ExtwallBrick                6.808e+03  6.867e+02   9.914  < 2e-16 ***
ExtwallFiber-Cement         4.306e+04  3.557e+03  12.107  < 2e-16 ***
ExtwallFrame               -4.593e+03  9.248e+02  -4.967 6.86e-07 ***
ExtwallMasonry / Frame      4.277e+03  1.626e+03   2.630 0.008545 **
ExtwallPrem Wood            2.179e+04  5.317e+03   4.098 4.18e-05 ***
ExtwallStone                1.053e+04  1.449e+03   7.267 3.78e-13 ***
ExtwallStucco               6.748e+03  2.045e+03   3.300 0.000969 ***
Stories1                   -2.495e+04  1.022e+04  -2.443 0.014591 *
Stories1.5                 -1.181e+04  1.020e+04  -1.158 0.246999
Stories2                   -8.609e+03  1.017e+04  -0.846 0.397364
Year_Built                  4.540e+02  1.529e+01  29.696  < 2e-16 ***
Fin_sqft                    5.631e+01  1.032e+00  54.554  < 2e-16 ***
Units1                      5.444e+04  6.991e+03   7.787 7.11e-15 ***
Units2                     -1.130e+04  6.987e+03  -1.617 0.105980
Units3                     -4.301e+04  7.520e+03  -5.720 1.08e-08 ***
Bdrms0                      9.220e+04  1.761e+04   5.236 1.66e-07 ***
Bdrms1                      4.984e+04  9.823e+03   5.074 3.93e-07 ***
Bdrms2                      5.665e+04  8.828e+03   6.417 1.42e-10 ***
Bdrms3                      6.222e+04  8.778e+03   7.089 1.39e-12 ***
Bdrms4                      5.422e+04  8.747e+03   6.199 5.76e-10 ***
Bdrms5                      5.172e+04  8.745e+03   5.914 3.39e-09 ***
Bdrms6                      3.924e+04  8.749e+03   4.485 7.31e-06 ***
Bdrms7                      1.455e+04  9.306e+03   1.564 0.117931
Bdrms8                      8.431e+03  9.786e+03   0.861 0.388983
Fbath0                     -2.592e+04  1.516e+04  -1.710 0.087331 .
Fbath1                     -1.398e+04  1.199e+04  -1.166 0.243478
Fbath2                      4.532e+03  1.195e+04   0.379 0.704540
Fbath3                      2.618e+04  1.190e+04   2.199 0.027856 *
Fbath4                      3.017e+04  1.261e+04   2.392 0.016750 *
Lotsize                     1.693e+00  1.133e-01  14.950  < 2e-16 ***
Sale_date                   4.690e+00  2.460e-01  19.061  < 2e-16 ***
district_3TRUE              1.033e+06  1.361e+05   7.590 3.30e-14 ***
district_4TRUE              1.060e+06  2.364e+05   4.482 7.43e-06 ***
district_15TRUE             4.386e+05  1.144e+05   3.834 0.000127 ***
Year_Built:district_3TRUE  -5.544e+02  7.159e+01  -7.745 9.95e-15 ***
```

```
Lotsize:district_3TRUE      1.745e+01  8.874e-01  19.670  < 2e-16 ***
Lotsize:district_4TRUE     -2.501e+00  1.728e+00  -1.447 0.147918
Year_Built:district_4TRUE  -5.488e+02  1.235e+02  -4.442 8.95e-06 ***
Year_Built:district_15TRUE -2.770e+02  5.942e+01  -4.661 3.16e-06 ***
Lotsize:district_15TRUE     3.255e+00  1.145e+00   2.843 0.004469 **
Fin_sqft:district_3TRUE     3.990e+01  1.725e+00  23.127  < 2e-16 ***
Fin_sqft:district_4TRUE    -1.681e+01  4.447e+00  -3.780 0.000157 ***
Fin_sqft:district_15TRUE   -1.344e+01  2.992e+00  -4.491 7.11e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41380 on 24329 degrees of freedom
Multiple R-squared:  0.6712,    Adjusted R-squared:  0.6706
F-statistic:  1104 on 45 and 24329 DF,  p-value: < 2.2e-16
```

```r
summary(model2)
```

```
Call:
lm(formula = Sale_price ~ . - district_3 - district_4 - district_15 +
    district_3 * Year_Built + district_3 * Lotsize + district_4 *
    Lotsize + district_4 * Year_Built + district_15 * Year_Built +
    district_15 * Lotsize + district_3 * Fin_sqft + district_4 *
    Fin_sqft + district_15 * Fin_sqft, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-303393  -24422    -942   23352  719561

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1.129e+06  3.625e+04 -31.143  < 2e-16 ***
District                    5.695e+03  7.701e+01  73.950  < 2e-16 ***
ExtwallBlock               -3.944e+03  3.688e+03  -1.069 0.284879
ExtwallBrick                6.725e+03  7.245e+02   9.281  < 2e-16 ***
ExtwallFiber-Cement         3.847e+04  3.754e+03  10.246  < 2e-16 ***
ExtwallFrame               -4.986e+03  9.786e+02  -5.095 3.51e-07 ***
ExtwallMasonry / Frame      3.751e+03  1.711e+03   2.192 0.028404 *
ExtwallPrem Wood            1.894e+04  5.596e+03   3.385 0.000714 ***
ExtwallStone                1.162e+04  1.531e+03   7.594 3.22e-14 ***
ExtwallStucco               3.514e+03  2.149e+03   1.635 0.102070
```

```
Stories1                     3.684e+02  1.047e+04   0.035 0.971935    
Stories1.5                   1.317e+04  1.046e+04   1.259 0.207954    
Stories2                     1.451e+04  1.042e+04   1.392 0.163802    
Year_Built                   4.629e+02  1.606e+01  28.823  < 2e-16 ***
Fin_sqft                     6.048e+01  1.080e+00  55.986  < 2e-16 ***
Units1                       5.778e+04  7.389e+03   7.820 5.48e-15 ***
Units2                      -9.774e+03  7.386e+03  -1.323 0.185735    
Units3                      -4.087e+04  7.940e+03  -5.147 2.67e-07 ***
Bdrms0                       1.155e+05  1.854e+04   6.227 4.82e-10 ***
Bdrms1                       7.894e+04  1.016e+04   7.766 8.41e-15 ***
Bdrms2                       8.331e+04  9.086e+03   9.170  < 2e-16 ***
Bdrms3                       8.791e+04  9.033e+03   9.733  < 2e-16 ***
Bdrms4                       7.905e+04  9.001e+03   8.782  < 2e-16 ***
Bdrms5                       7.507e+04  9.002e+03   8.340  < 2e-16 ***
Bdrms6                       6.149e+04  9.006e+03   6.828 8.80e-12 ***
Bdrms7                       2.462e+04  9.572e+03   2.573 0.010100 *  
Bdrms8                       1.242e+04  1.008e+04   1.233 0.217672    
Fbath0                      -1.936e+04  1.388e+04  -1.395 0.162964    
Fbath1                      -1.050e+04  9.795e+03  -1.072 0.283695    
Fbath2                       7.289e+03  9.751e+03   0.748 0.454721    
Fbath3                       2.935e+04  9.641e+03   3.044 0.002339 ** 
Fbath4                       6.757e+04  1.025e+04   6.595 4.35e-11 ***
Lotsize                      1.419e+00  1.131e-01  12.545  < 2e-16 ***
Sale_date                    4.839e+00  2.603e-01  18.587  < 2e-16 ***
district_3TRUE               9.189e+05  1.373e+05   6.693 2.23e-11 ***
district_4TRUE               1.088e+06  2.504e+05   4.346 1.39e-05 ***
district_15TRUE              4.475e+05  1.212e+05   3.692 0.000223 ***
Year_Built:district_3TRUE   -5.100e+02  7.205e+01  -7.078 1.51e-12 ***
Lotsize:district_3TRUE       1.227e+01  4.803e-01  25.549  < 2e-16 ***
Lotsize:district_4TRUE      -3.796e+00  1.569e+00  -2.420 0.015512 *  
Year_Built:district_4TRUE   -5.558e+02  1.306e+02  -4.255 2.09e-05 ***
Year_Built:district_15TRUE -2.829e+02  6.295e+01  -4.493 7.05e-06 ***
Lotsize:district_15TRUE      3.140e+00  1.213e+00   2.589 0.009620 ** 
Fin_sqft:district_3TRUE      6.621e+01  1.517e+00  43.639  < 2e-16 ***
Fin_sqft:district_4TRUE     -2.049e+01  4.181e+00  -4.901 9.61e-07 ***
Fin_sqft:district_15TRUE    -1.161e+01  3.163e+00  -3.672 0.000241 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43870 on 24429 degrees of freedom
Multiple R-squared:  0.7301,    Adjusted R-squared:  0.7296 
F-statistic:  1468 on 45 and 24429 DF,  p-value: < 2.2e-16
```

```r
# Notice that some of the coefficients moved several standard errors!! This is a huge chan
sort(abs(model3$coefficients-model2$coefficients)/s$coefficients[,2],T)
```

```
 Fin_sqft:district_3TRUE     Lotsize:district_3TRUE
             15.24505072                 5.84301996
                Fin_sqft                     Bdrms2
              4.04375955                 3.02074093
                  Fbath4                     Bdrms1
              2.96653316                 2.96269835
                  Bdrms3                     Bdrms4
              2.92678462                 2.83832863
                  Bdrms5                     Bdrms6
              2.67068305                 2.54317351
                Stories1                 Stories1.5
              2.47860999                 2.44804272
                 Lotsize                   Stories2
              2.42348227                 2.27328085
             (Intercept)             ExtwallStucco
              2.26165432                 1.58135003
                  Bdrms0       ExtwallFiber-Cement
              1.32148050                 1.29194315
                  Bdrms7           district_3TRUE
              1.08240741                 0.83667557
 Fin_sqft:district_4TRUE              ExtwallStone
              0.82686340                 0.75295649
  Lotsize:district_4TRUE  Year_Built:district_3TRUE
              0.74951182                 0.62078382
Fin_sqft:district_15TRUE                 Sale_date
              0.60937648                 0.60556257
              Year_Built          ExtwallPrem Wood
              0.58139942                 0.53606561
                  Units1              ExtwallBlock
              0.47841864                 0.47569367
                  Fbath0              ExtwallFrame
              0.43234373                 0.42503952
                  Bdrms8     ExtwallMasonry / Frame
              0.40777572                 0.32346851
                  Fbath1                     Units3
              0.29018635                 0.28524471
                  Fbath3                     Fbath2
              0.26581945                 0.23068338
                  Units2                   District
```

```
              0.21782363                  0.13426306
            ExtwallBrick               district_4TRUE
              0.12107709                  0.12052136
Lotsize:district_15TRUE Year_Built:district_15TRUE
              0.10019176                  0.09899458
          district_15TRUE  Year_Built:district_4TRUE
              0.07738453                  0.05719434
```

How should we treat influential observations? The easiest course of action is removal. If there are many influential observations, then you might want to try robust model fitting methods, which automatically account for outliers and influential observations.

7.4 Homework questions

Complete the Chapter 6 textbook questions.

Exercise 7.1. What are the three methods we have learned for detecting influential/leverage points?

Exercise 7.2. Compute the hat values, Cook's distances and the depth values for the body weight example. Are there any influential/leverage points/outliers?

Exercise 7.3. Compute the hat values, Cook's distances and the depth values for the cars example. Are there any outliers/influential/leverage points?

Exercise 7.4. Fit a model without location to the real estate data of your choosing. Compute the hat values, Cook's distances and the depth values for the cars example. Are there any influential/leverage points/outliers? Print out the influential/leverage points/outliers. Why do you think they are outlying? Should we remove them?

8 Multicollinearity

8.1 Multicollinearity and the problems it creates

A serious problem that may dramatically impact the usefulness of a regression model is multicollinearity, or near-linear dependence among the regression variables. That is multicollinearity refers to near-linear dependence among the regressors. The regressors are the columns of the X matrix, so clearly an exact linear dependence among the regressors would result in a singular $X^\top X$. This will impact our ability to estimate β.

To elaborate, assume that the regressor variables and the response have been centered and scaled to unit length. The matrix $X^\top X$ is then a $p \times p$ correlation matrix (of the vector of regressors) and $X^\top Y$ is the vector of correlations between the regressors and response. Recall that a set of vectors $v_1, \dots, v_n$ are linearly dependent if there exists $c \neq 0$ such that $\sum_{i=1}^n c_i v_i = 0$. If the columns of X are linearly dependent, then $X^\top X$ is not invertible! We say there is multicollinearity if there exists $c \neq 0$ such that $\sum_{i=1}^n c_i v_i < \epsilon$ for some small ϵ.

Multicollinearity results in large variances and covariances for the least - squares estimators of the regression coefficients. Let $A = (X^\top X)^{-1}$, where the regressors have been centered and scaled to unit length. That is, columns $2, \dots, p$ of X have their mean subtracted and are divided by their respective norms. Then

$$A_{jj} = (1 - R_j^2)^{-1},$$

where R_j^2 is the coefficient of multiple determination from the regression of X_j on the remaining $p-1$ regressor variables. Now, recall that $\mathrm{Var}\left[\hat{\beta}_j\right] = A_{jj}\sigma^2$. What happens when the correlation is approximately 1 between X_j and another regressor? It is easy to see that the the variance of coefficient j goes to ∞ as $\mathrm{corr}\left[X_j, X_i\right] \to 1$.

This huge variance results in large magnitude of the least squares estimators of the regression coefficients. We have that $\mathrm{E}\left[\left\|\hat{\beta} - \beta\right\|^2\right] = \sum_{i=1}^p \mathrm{Var}\left[\hat{\beta}_i\right] = \sigma^2 \,\mathrm{trace}(X^\top X)^{-1}$. Recall! $\mathrm{trace}(A)$ is the sum of its eigenvalues. If $X^\top X$ has near linearly dependent columns, then some of the eigenvalues $\lambda_1, \dots, \lambda_p$ will be near 0 (why?). Thus,

$$\mathrm{E}\left[\left\|\hat{\beta} - \beta\right\|^2\right] = \sigma^2 \,\mathrm{trace}(X^\top X)^{-1} = \sigma^2 \sum_{i=1}^p \frac{1}{\lambda_i}.$$

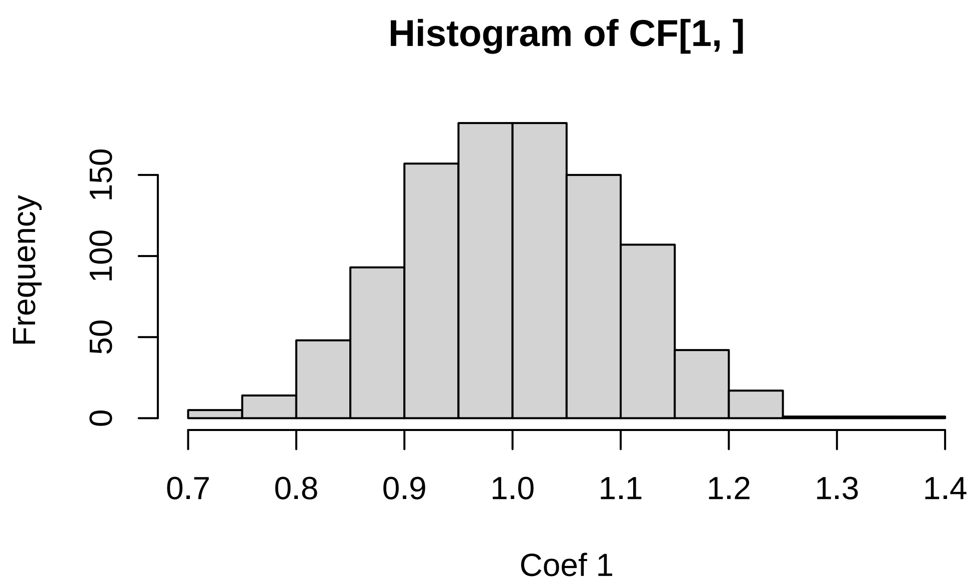
We can also show that

$$\mathrm{E}\left[\widehat{\beta}^{\top}\widehat{\beta}\right]=\beta^{\top}\beta+\sigma^{2}\operatorname{Tr}\left(X^{\top}X\right)^{-1},$$

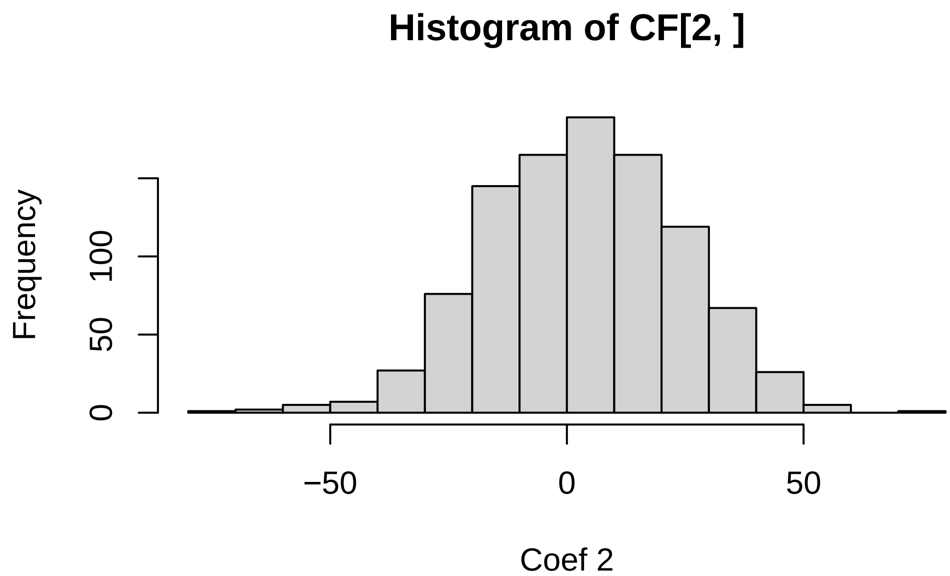
which gives the same interpretation.

We can also observe this empirically.

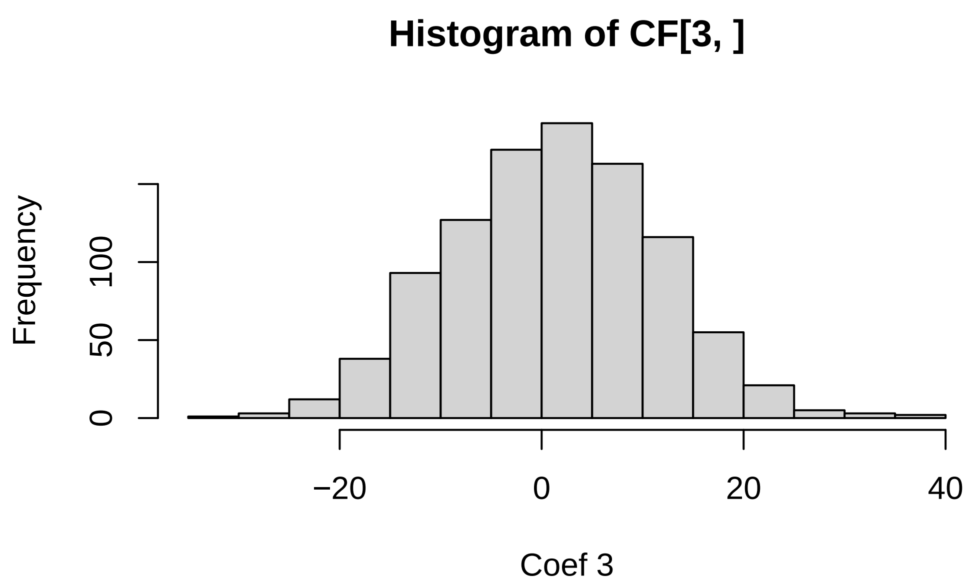
```
# Let's simulate data from a regression model with highly correlated regressors.
simulate_coef=function(){
  n=100
  X=rnorm(n)
  X2=2*X+rnorm(n,0,0.01)
  Y=1+2*X+X2*2+rnorm(n)
  return(coef(lm(Y~X+X2)))
}

# See the HUGE variance in the estimated coefficients? O this is from MCL!
CF=replicate(1000,simulate_coef())
hist(CF[1,],xlab='Coef 1')
```




```
hist(CF[2,],xlab='Coef 2')
```




```
hist(CF[3,],xlab='Coef 3')
```



It is easy to see from this analysis that multicollinearity is a serious problem, and we should check for it in regression modelling.

8.2 Multicollinearity Diagnostics

8.2.1 VIF

We need some diagnostics to detect multicollinearity. The first of which is the **correlation matrix**, which is good for detecting pairwise correlations, but not so much for more complicated dependencies! To correct this, a popular diagnostic is the **variance inflation factor** (VIF): these are the diagonals of $(X^\top X)^{-1}$. A VIF that exceeds 3, 5 or 10 is an indication that the associated regression coefficients are poorly estimated because of multicollinearity. The variance inflation factor can be written as $(1-R_j^2)^{-1}$, where R_j^2 is the coefficient of determination obtained from regressing x_j on the other regressor variables. For categorical variables, we may look at their VIF together, instead of for the individual dummy variables. This is done via the generalized VIF (GVIF), which was developed by our very own Georges Monette and John Fox (Fox and Monette 1992). We can consider the $(GVIF^{(1/(2\text{number of dummy variables}))})$. However, we want to compare these to the square root of our rules of thumb, so $\sqrt{3}, \sqrt{5}, \sqrt{10}$. The values $(GVIF^{(1/(2\text{number of dummy variables}))})$ are computed automatically in R. When your model has interaction effects, or polynomial terms, it is best to exclude those and compute the VIFs. In summary:

- Compare the VIF of continuous variables to 3, 5 or 10, above those values signals multicollinearity
- Compare the $(GVIF^{(1/(2\text{number of dummy variables}))})$ of categorical variables to $\sqrt{3}, \sqrt{5},$ or $\sqrt{10}$, above those values signals multicollinearity
- When computing VIF and GVIF, it is best to exclude interaction effects and or polynomial terms
- The given thresholds are rules of thumb, and we should not discard evidence of multicollinearity of we are very close to a threshold, e.g., 9.9.

8.2.2 Condition number

One can also look at the eigenvalues of $X^\top X$, where the regressors are centered and normalized to unit length. If the eigenvalues are small, this indicates multicollinearity. One metric computed from the eigenvalues is the **condition number** of $X^\top X$: $\kappa = \max \lambda_j / \min \lambda_j$, where the regressors are centered and normalized to unit length. Condition numbers between 100 and 1000 imply moderate to strong multicollinearity, and if κ exceeds 1000, severe multicollinearity is indicated. Diagonalizing via $X^\top X = \Lambda D \Lambda^\top$ yields the eigenvectors, which help us determine the exact dependence between variables is. You can check the eigenvectors associated with the small eigenvalues. Components that are large in the eigenvector indicate that that variable is contributing to the multicollinearity. Again, for this computation, it is best to exclude interaction effects and or polynomial terms.

> **Note**
>
> While the method of least squares will generally produce poor estimates of the individual model parameters when strong multicollinearity is present, this does not necessarily imply that the fitted model is a poor predictor.
>
> 1. If predictions are confined to regions of the X-space where the multicollinearity holds approximately, the fitted model often produces satisfactory predictions.
> 2. The linear combinations $X\beta$ may be estimated well, even if β is not.

Example 8.1. Recall example Example 6.6. Check for multicollinearity in the proposed models.

I will load in the data below:

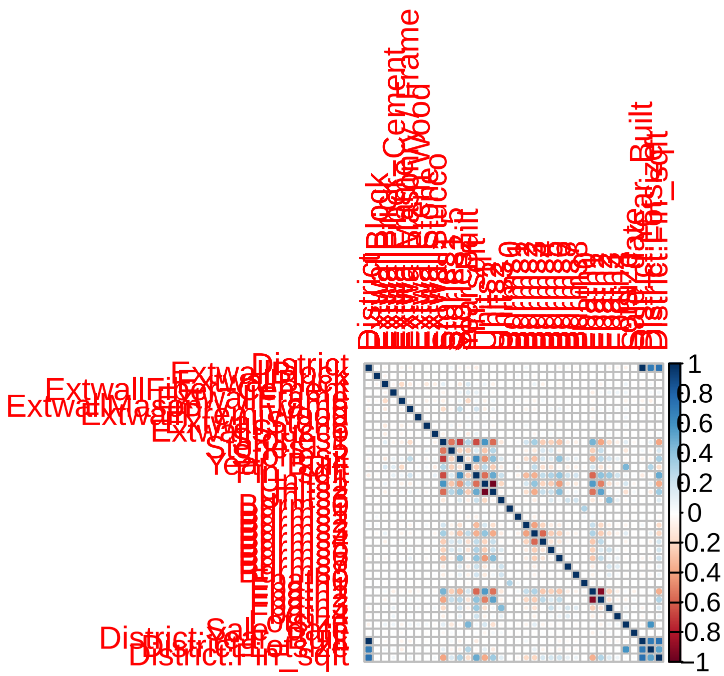
```
model2=lm(Sale_price~.+District*Year_Built+District*Lotsize+District*Fin_sqft
          ,df)




X=model.matrix(model2)
depths=ddalpha::depth.projection(cbind(X,df$Sale_price),cbind(X,df$Sale_price))

model3=lm(Sale_price~.+District*Year_Built+District*Lotsize+District*Fin_sqft
         ,df[-(order(depths)[1:100]),])



summary(model3)
```

```
Call:
lm(formula = Sale_price ~ . + District * Year_Built + District *
    Lotsize + District * Fin_sqft, data = df[-(order(depths)[1:100]),
    ])

Residuals:
    Min      1Q  Median      3Q     Max
```

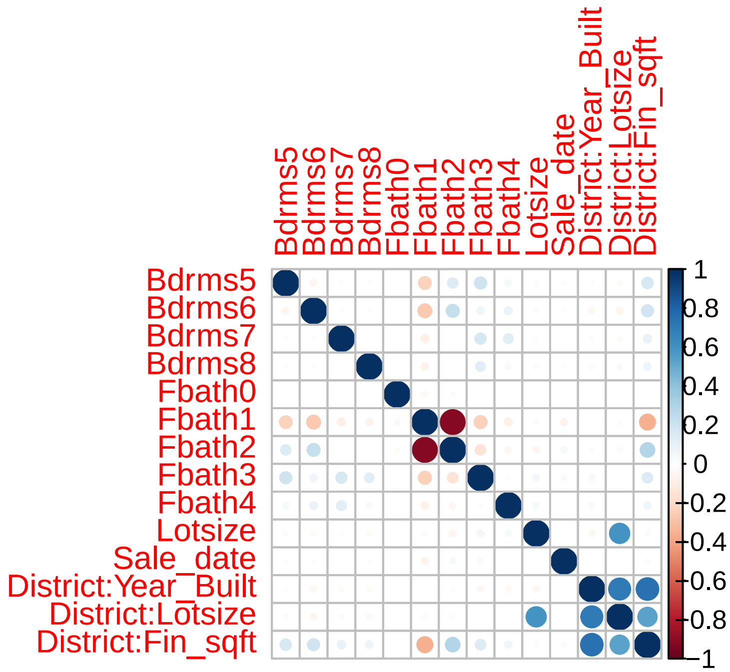
```
-605729  -27681     733   26954  454626

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)             3.497e+05  7.854e+04   4.453 8.51e-06 ***
District               -9.652e+04  7.589e+03 -12.718  < 2e-16 ***
ExtwallBlock           -2.883e+03  4.445e+03  -0.649 0.516580    
ExtwallBrick            7.562e+03  8.735e+02   8.658  < 2e-16 ***
ExtwallFiber-Cement     2.919e+04  4.511e+03   6.471 9.93e-11 ***
ExtwallFrame           -5.128e+03  1.178e+03  -4.353 1.35e-05 ***
ExtwallMasonry / Frame  3.248e+03  2.064e+03   1.574 0.115595    
ExtwallPrem Wood        7.980e+03  6.991e+03   1.142 0.253654    
ExtwallStone            3.273e+03  1.847e+03   1.772 0.076384 .  
ExtwallStucco           1.336e+04  2.588e+03   5.161 2.47e-07 ***
Stories1                3.254e+04  1.237e+04   2.630 0.008535 ** 
Stories1.5              4.380e+04  1.235e+04   3.547 0.000390 ***
Stories2                5.023e+04  1.230e+04   4.084 4.44e-05 ***
Year_Built             -3.760e+02  3.874e+01  -9.705  < 2e-16 ***
Fin_sqft                1.244e+02  1.579e+00  78.778  < 2e-16 ***
Units1                  1.073e+05  8.840e+03  12.138  < 2e-16 ***
Units2                  2.757e+04  8.840e+03   3.118 0.001820 ** 
Units3                 -1.979e+04  9.521e+03  -2.079 0.037654 *  
Bdrms0                  1.410e+05  2.227e+04   6.332 2.46e-10 ***
Bdrms1                  1.175e+05  1.217e+04   9.655  < 2e-16 ***
Bdrms2                  1.236e+05  1.087e+04  11.374  < 2e-16 ***
Bdrms3                  1.257e+05  1.080e+04  11.639  < 2e-16 ***
Bdrms4                  1.130e+05  1.075e+04  10.508  < 2e-16 ***
Bdrms5                  1.126e+05  1.075e+04  10.477  < 2e-16 ***
Bdrms6                  9.129e+04  1.076e+04   8.482  < 2e-16 ***
Bdrms7                  6.829e+04  1.147e+04   5.951 2.70e-09 ***
Bdrms8                  6.948e+04  1.213e+04   5.729 1.02e-08 ***
Fbath0                 -4.625e+04  1.725e+04  -2.681 0.007341 ** 
Fbath1                 -3.279e+04  1.262e+04  -2.598 0.009390 ** 
Fbath2                 -1.305e+04  1.256e+04  -1.040 0.298543    
Fbath3                  3.011e+04  1.248e+04   2.412 0.015867 *  
Fbath4                  4.767e+04  1.340e+04   3.558 0.000374 ***
Lotsize                 7.107e-01  3.166e-01   2.245 0.024785 *  
Sale_date               4.769e+00  3.138e-01  15.196  < 2e-16 ***
District:Year_Built     5.486e+01  3.923e+00  13.985  < 2e-16 ***
District:Lotsize        1.054e-01  3.591e-02   2.934 0.003348 ** 
District:Fin_sqft      -5.619e+00  1.440e-01 -39.024  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 52780 on 24338 degrees of freedom
Multiple R-squared:  0.5262,    Adjusted R-squared:  0.5255
F-statistic: 750.9 on 36 and 24338 DF,  p-value: < 2.2e-16
```

```r
# Correlations
X=model.matrix(model3)
corrplot::corrplot(cor(X[,-1]))
```




```r
corrplot::corrplot(cor(X[,-(1:23)]))
```




```r
# VIFS
# predictor removes the interactions
car::vif(model3,type = 'predictor')
```

```
GVIFs computed for predictors

                   GVIF Df GVIF^(1/(2*Df))                Interacts With
District       5.938550  7        1.135698 Year_Built, Lotsize, Fin_sqft
Extwall        1.408741  8        1.021650                            --
Stories        2.837098  3        1.189814                            --
Year_Built    27.273468  3        1.734962                      District
Fin_sqft   40881.039870  3        5.869309                      District
Units          3.279782  3        1.218917                            --
Bdrms          4.229339  9        1.083410                            --
Fbath          3.333999  5        1.127967                            --
Lotsize    12316.565475  3        4.805607                      District
Sale_date      1.018330  1        1.009123                            --
                                                                       Other Predictor
District                                 Extwall, Stories, Units, Bdrms, Fbath, Sale_dat
Extwall       District, Stories, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Lotsize, Sale_dat
Stories       District, Extwall, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Lotsize, Sale_dat
```

```
Year_Built                Extwall, Stories, Fin_sqft, Units, Bdrms, Fbath, Lotsize, Sale_date
Fin_sqft                Extwall, Stories, Year_Built, Units, Bdrms, Fbath, Lotsize, Sale_date
Units      District, Extwall, Stories, Year_Built, Fin_sqft, Bdrms, Fbath, Lotsize, Sale_date
Bdrms      District, Extwall, Stories, Year_Built, Fin_sqft, Units, Fbath, Lotsize, Sale_date
Fbath      District, Extwall, Stories, Year_Built, Fin_sqft, Units, Bdrms, Lotsize, Sale_date
Lotsize                Extwall, Stories, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Sale_date
Sale_date      District, Extwall, Stories, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Lotsize
```

```
model4=lm(Sale_price~Extwall +Stories+Year_Built+Fin_sqft+Units+Bdrms+Fbath+Sale_date+Dist
        ,df[-(order(depths)[1:100]),])

car::vif(model4,type = 'predictor')
```

```
GVIFs computed for predictors
```

```
                   GVIF Df GVIF^(1/(2*Df))       Interacts With
Extwall        1.395529  8        1.021048                   --
Stories        2.794397  3        1.186810                   --
Year_Built    10.284838  3        1.474686             District
Fin_sqft   31689.695476  3        5.625395             District
Units          3.242258  3        1.216582                   --
Bdrms          4.186673  9        1.082800                   --
Fbath          3.331086  5        1.127869                   --
Sale_date      1.018215  1        1.009066                   --
District       5.553636  5        1.187019 Year_Built, Fin_sqft
                                                                      Other Predictors
Extwall      Stories, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Sale_date, District
Stories      Extwall, Year_Built, Fin_sqft, Units, Bdrms, Fbath, Sale_date, District
Year_Built                Extwall, Stories, Fin_sqft, Units, Bdrms, Fbath, Sale_date
Fin_sqft                Extwall, Stories, Year_Built, Units, Bdrms, Fbath, Sale_date
Units      Extwall, Stories, Year_Built, Fin_sqft, Bdrms, Fbath, Sale_date, District
Bdrms      Extwall, Stories, Year_Built, Fin_sqft, Units, Fbath, Sale_date, District
Fbath      Extwall, Stories, Year_Built, Fin_sqft, Units, Bdrms, Sale_date, District
Sale_date      Extwall, Stories, Year_Built, Fin_sqft, Units, Bdrms, Fbath, District
District                          Extwall, Stories, Units, Bdrms, Fbath, Sale_date
```

```
model4=lm(Sale_price~Extwall +Stories+Year_Built+
            Lotsize+Units+Bdrms+Fbath+Sale_date+District*Year_Built+District+District*Lots
        ,df[-(order(depths)[1:100]),])
```

```r
X=model.matrix(model4)
car::vif(model4,type = 'predictor')
```

```
GVIFs computed for predictors

                   GVIF Df GVIF^(1/(2*Df))      Interacts With
Extwall        1.278176  8        1.015458                  --
Stories        2.158353  3        1.136808                  --
Year_Built    12.133152  3        1.515871            District
Lotsize    10081.538963  3        4.647875            District
Units          3.188879  3        1.213220                  --
Bdrms          3.148134  9        1.065785                  --
Fbath          2.677922  5        1.103519                  --
Sale_date      1.017871  1        1.008896                  --
District       1.381580  5        1.032851 Year_Built, Lotsize
                                                                  Other Predictors
Extwall      Stories, Year_Built, Lotsize, Units, Bdrms, Fbath, Sale_date, District
Stories      Extwall, Year_Built, Lotsize, Units, Bdrms, Fbath, Sale_date, District
Year_Built                Extwall, Stories, Lotsize, Units, Bdrms, Fbath, Sale_date
Lotsize                Extwall, Stories, Year_Built, Units, Bdrms, Fbath, Sale_date
Units      Extwall, Stories, Year_Built, Lotsize, Bdrms, Fbath, Sale_date, District
Bdrms      Extwall, Stories, Year_Built, Lotsize, Units, Fbath, Sale_date, District
Fbath      Extwall, Stories, Year_Built, Lotsize, Units, Bdrms, Sale_date, District
Sale_date    Extwall, Stories, Year_Built, Lotsize, Units, Bdrms, Fbath, District
District                           Extwall, Stories, Units, Bdrms, Fbath, Sale_date
```

```r
model4=lm(Sale_price~Extwall +Stories+
            Lotsize+Units+Bdrms+Fbath+Sale_date+District+District*Lotsize
         ,df[-(order(depths)[1:100]),])

car::vif(model4,type = 'predictor')
```

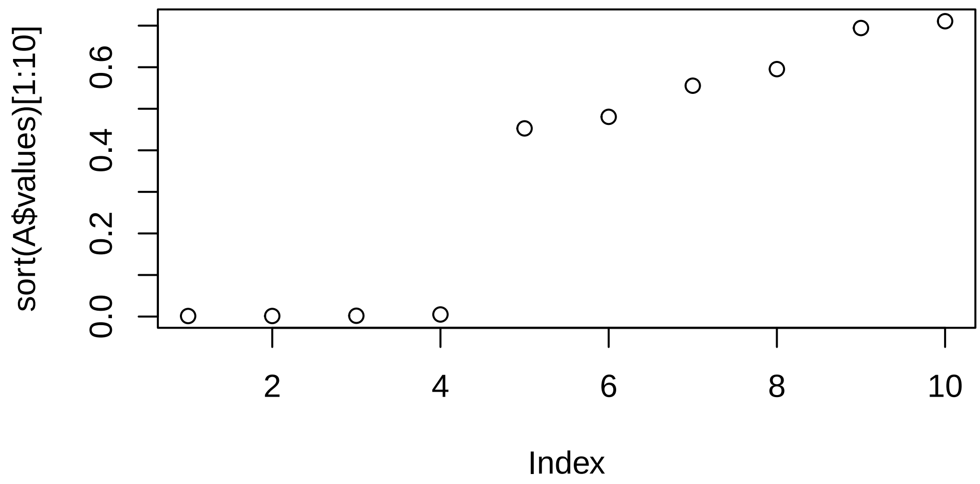
```
GVIFs computed for predictors

              GVIF Df GVIF^(1/(2*Df)) Interacts With
Extwall   1.196395  8        1.011270             --
Stories   2.068411  3        1.128772             --
Lotsize   1.111905  3        1.017836       District
Units     3.165637  3        1.211742             --
```

```
Bdrms     3.063017  9        1.064163           --
Fbath     2.653707  5        1.102517           --
Sale_date 1.017665  1        1.008794           --
District  1.111905  3        1.017836        Lotsize
                                                          Other Predictors
Extwall     Stories, Lotsize, Units, Bdrms, Fbath, Sale_date, District
Stories     Extwall, Lotsize, Units, Bdrms, Fbath, Sale_date, District
Lotsize               Extwall, Stories, Units, Bdrms, Fbath, Sale_date
Units     Extwall, Stories, Lotsize, Bdrms, Fbath, Sale_date, District
Bdrms     Extwall, Stories, Lotsize, Units, Fbath, Sale_date, District
Fbath     Extwall, Stories, Lotsize, Units, Bdrms, Sale_date, District
Sale_date     Extwall, Stories, Lotsize, Units, Bdrms, Fbath, District
District              Extwall, Stories, Units, Bdrms, Fbath, Sale_date
```

```
# Let's go back to

model4=lm(Sale_price~Extwall +Stories+Year_Built+
            Lotsize+Units+Bdrms+Fbath+Sale_date+District*Year_Built+District+District*Lots
         ,df[-(order(depths)[1:100]),])

# remove interactions to compute the condition number

model4=lm(Sale_price~Extwall +Stories+Year_Built+
            Lotsize+Units+Bdrms+Fbath+Sale_date+District
         ,df[-(order(depths)[1:100]),])


X=model.matrix(model4)

# Correlations
# X=X[,-1]
X2=apply(X,2,function(x){(x-mean(x))/sqrt(sum((x-mean(x))^2))}); dim(X2)
```

```
[1] 24375    33
```

```
X2[,1]=X[,1]
X=X2
A=eigen(t(X)%*%X)

plot(sort(A$values)[1:10])
```




```r
which.min(A$values)
```

```
[1] 33
```

```r
dim(A$vectors)
```

```
[1] 33 33
```

```r
#Condition number
max(A$values)/min(A$values)
```

```
[1] 19002497
```

```r
rownames(A$vectors)=names(model4$coefficients)

# Which components are important here

A$vectors[,which.min(A$values)]
```

```
          (Intercept)          ExtwallBlock          ExtwallBrick
         0.000000e+00          2.157874e-04          5.978469e-04
  ExtwallFiber-Cement          ExtwallFrame ExtwallMasonry / Frame
         7.423672e-04         -6.847029e-04         -2.499551e-04
     ExtwallPrem Wood          ExtwallStone         ExtwallStucco
         6.226662e-05          4.436149e-04         -4.341195e-05
             Stories1           Stories1.5              Stories2
        -1.096523e-01         -7.806434e-02         -9.473269e-02
           Year_Built               Lotsize                Units1
        -9.817844e-04          1.334084e-03          6.123567e-02
               Units2                Units3                Bdrms0
         5.775492e-02          1.114147e-02         -1.374989e-02
               Bdrms1                Bdrms2                Bdrms3
        -4.871769e-02         -2.715348e-01         -4.006905e-01
               Bdrms4                Bdrms5                Bdrms6
        -3.387437e-01         -1.808355e-01         -1.715717e-01
               Bdrms7                Bdrms8                Fbath0
        -5.178211e-02         -3.999879e-02          3.083921e-02
               Fbath1                Fbath2                Fbath3
         5.084776e-01          4.976796e-01          1.911793e-01
               Fbath4             Sale_date              District
         6.281346e-02         -2.373692e-04         -2.413410e-04
```

```
# round(A$vectors[,which.min(A$values)],1)
round(A$vectors[,which.min(A$values)][abs(round(A$vectors[,which.min(A$values)],1))>0],1)
```

```
  Stories1 Stories1.5   Stories2     Units1     Units2     Bdrms2     Bdrms3
      -0.1       -0.1       -0.1        0.1        0.1       -0.3       -0.4
    Bdrms4     Bdrms5     Bdrms6     Bdrms7     Fbath1     Fbath2     Fbath3
      -0.3       -0.2       -0.2       -0.1        0.5        0.5        0.2
    Fbath4
       0.1
```

```
min_4=order(A$values)[1:4]
A$vectors[,min_4]
```

```
                                [,1]          [,2]          [,3]          [,4]
(Intercept)             0.000000e+00  0.000000e+00  0.0000000000  0.000000e+00
ExtwallBlock            2.157874e-04 -9.248583e-05  0.0002779889 -1.667631e-03
ExtwallBrick            5.978469e-04 -1.428128e-04 -0.0010585073 -6.556687e-04
```

```
ExtwallFiber-Cement     7.423672e-04 -4.286608e-04 -0.0007023650 -4.203278e-05
ExtwallFrame           -6.847029e-04  5.086148e-05 -0.0005434478 -4.403961e-03
ExtwallMasonry / Frame -2.499551e-04  5.955587e-04 -0.0009708358  9.946691e-04
ExtwallPrem Wood        6.226662e-05  2.760081e-05  0.0001286685 -2.165509e-05
ExtwallStone            4.436149e-04 -1.433889e-04 -0.0003263289 -3.862695e-04
ExtwallStucco          -4.341195e-05 -8.011335e-04  0.0001710557 -1.563241e-03
Stories1               -1.096523e-01 -6.007637e-01 -0.2607427050  2.131969e-03
Stories1.5             -7.806434e-02 -4.332066e-01 -0.1881242671  9.060758e-04
Stories2               -9.473269e-02 -5.191819e-01 -0.2266943948 -2.650768e-03
Year_Built             -9.817844e-04 -3.955217e-04  0.0018717800  2.586720e-03
Lotsize                 1.334084e-03  8.206240e-04 -0.0020666455 -6.621107e-04
Units1                  6.123567e-02 -3.705218e-02  0.0595563386 -6.979876e-01
Units2                  5.775492e-02 -3.591622e-02  0.0578395460 -6.828344e-01
Units3                  1.114147e-02 -7.410112e-03  0.0101288065 -1.715009e-01
Bdrms0                 -1.374989e-02  8.383116e-03 -0.0135203489 -1.164538e-02
Bdrms1                 -4.871769e-02  2.977888e-02 -0.0480132104 -1.179837e-02
Bdrms2                 -2.715348e-01  1.655464e-01 -0.2678050333 -5.423330e-02
Bdrms3                 -4.006905e-01  2.445033e-01 -0.3950766818 -7.970937e-02
Bdrms4                 -3.387437e-01  2.071965e-01 -0.3344989333 -6.761525e-02
Bdrms5                 -1.808355e-01  1.111307e-01 -0.1789307374 -3.566710e-02
Bdrms6                 -1.715717e-01  1.064450e-01 -0.1715086899 -3.431019e-02
Bdrms7                 -5.178211e-02  3.353689e-02 -0.0558980648 -1.545699e-02
Bdrms8                 -3.999879e-02  2.518976e-02 -0.0426789439 -1.354742e-02
Fbath0                  3.083921e-02  6.468728e-03 -0.0276744921  2.916203e-03
Fbath1                  5.084776e-01  1.037524e-01 -0.4491473633 -1.266652e-03
Fbath2                  4.976796e-01  1.008202e-01 -0.4404995267  1.570382e-03
Fbath3                  1.911793e-01  3.888777e-02 -0.1731428547  8.703757e-04
Fbath4                  6.281346e-02  1.210395e-02 -0.0561070476 -3.668197e-03
Sale_date              -2.373692e-04  7.838366e-04  0.0002838833  7.182291e-03
District               -2.413410e-04 -4.293919e-04  0.0006999058  7.545907e-04
```

```r
# round(A$vectors[,which.min(A$values)],1)
for(i in min_4)
  print(round(A$vectors[,i][abs(round(A$vectors[,i],1))>0],1))
```

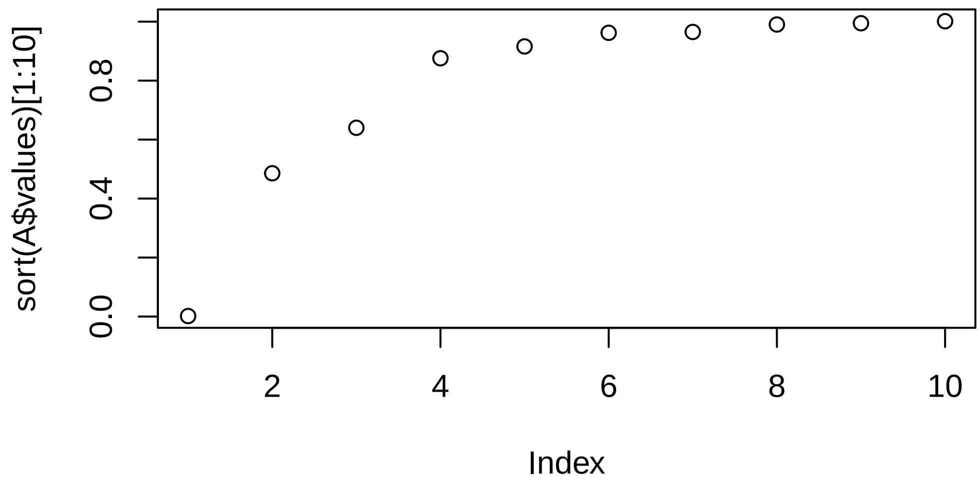
```
  Stories1 Stories1.5   Stories2     Units1     Units2     Bdrms2     Bdrms3
      -0.1       -0.1       -0.1        0.1        0.1       -0.3       -0.4
    Bdrms4     Bdrms5     Bdrms6     Bdrms7     Fbath1     Fbath2     Fbath3
      -0.3       -0.2       -0.2       -0.1        0.5        0.5        0.2
    Fbath4
       0.1
```

```
  Stories1 Stories1.5   Stories2     Bdrms2     Bdrms3     Bdrms4     Bdrms5
      -0.6       -0.4       -0.5        0.2        0.2        0.2        0.1
    Bdrms6     Fbath1     Fbath2
       0.1        0.1        0.1
  Stories1 Stories1.5   Stories2     Units1     Units2     Bdrms2     Bdrms3
      -0.3       -0.2       -0.2        0.1        0.1       -0.3       -0.4
    Bdrms4     Bdrms5     Bdrms6     Bdrms7     Fbath1     Fbath2     Fbath3
      -0.3       -0.2       -0.2       -0.1       -0.4       -0.4       -0.2
    Fbath4
      -0.1
Units1 Units2 Units3 Bdrms2 Bdrms3 Bdrms4
  -0.7   -0.7   -0.2   -0.1   -0.1   -0.1
```

```
model5=lm(Sale_price~Extwall +Year_Built+
            Lotsize+Bdrms+Sale_date+District
         ,df[-(order(depths)[1:100]),])


X=model.matrix(model5)

# Correlations
# X=X[,-1]
X2=apply(X,2,function(x){(x-mean(x))/sqrt(sum((x-mean(x))^2))}); dim(X2)
```

```
[1] 24375    22
```

```
X2[,1]=X[,1]
X=X2
A=eigen(t(X)%*%X)

plot(sort(A$values)[1:10])
```




```r
which.min(A$values)
```

```
[1] 22
```

```r
dim(A$vectors)
```

```
[1] 22 22
```

```r
#Condition number
max(A$values)/min(A$values)
```

```
[1] 14485425
```

```r
rownames(A$vectors)=names(model5$coefficients)

# Which components are important here

A$vectors[,which.min(A$values)]
```

```
          (Intercept)            ExtwallBlock            ExtwallBrick
         0.000000e+00           -4.381720e-05            4.819699e-04
  ExtwallFiber-Cement            ExtwallFrame ExtwallMasonry / Frame
         1.010497e-04            1.468968e-03            1.348000e-03
     ExtwallPrem Wood            ExtwallStone           ExtwallStucco
        -2.910105e-05           -1.075400e-04           -1.670649e-04
           Year_Built                 Lotsize                  Bdrms0
        -1.509724e-03            9.687646e-04            2.236661e-02
               Bdrms1                  Bdrms2                  Bdrms3
         7.531941e-02            4.183638e-01            6.175005e-01
               Bdrms4                  Bdrms5                  Bdrms6
         5.236352e-01            2.813968e-01            2.687451e-01
               Bdrms7                  Bdrms8               Sale_date
         8.635652e-02            6.651558e-02           -4.121579e-04
             District
        -5.815139e-04
```

```
# round(A$vectors[,which.min(A$values)],1)
round(A$vectors[,which.min(A$values)][abs(round(A$vectors[,which.min(A$values)],1))>0],1)
```

```
Bdrms1 Bdrms2 Bdrms3 Bdrms4 Bdrms5 Bdrms6 Bdrms7 Bdrms8
   0.1    0.4    0.6    0.5    0.3    0.3    0.1    0.1
```

```
# This is fine
```

Example 8.2. Example 9.1 from the textbook - The Acetylene Data. Below presents data concerning the percentage of conversion of n - heptane to acetylene and three explanatory variables. These are typical chemical process data for which a full quadratic response surface in all three regressors is often considered to be an appropriate tentative model. Let's build the model and see how the extrapolation performs.

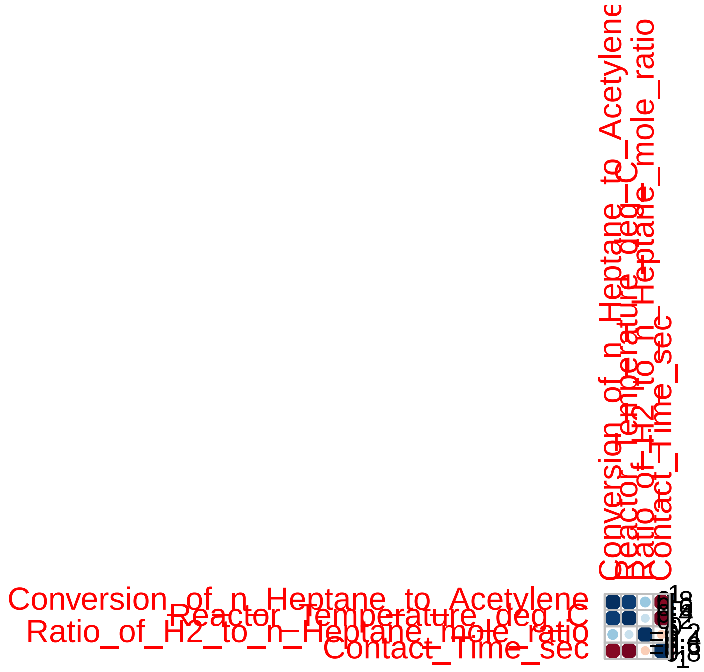
```
########### Example 2




df <- data.frame(
  Conversion_of_n_Heptane_to_Acetylene = c(49.0, 50.2, 50.5, 48.5, 47.5, 44.5, 28.0, 31.5,
  Reactor_Temperature_deg_C = c(1300, 1300, 1300, 1300, 1300, 1300, 1200, 1200, 1200, 1200
  Ratio_of_H2_to_n_Heptane_mole_ratio = c(7.5, 9.0, 11.0, 13.5, 17.0, 23.0, 5.3, 7.5, 11.0
```

```r
  Contact_Time_sec = c(0.0120, 0.0120, 0.0115, 0.0130, 0.0135, 0.0120, 0.0400, 0.0380, 0.0
)

# Printing the dataframe
head(df)
```

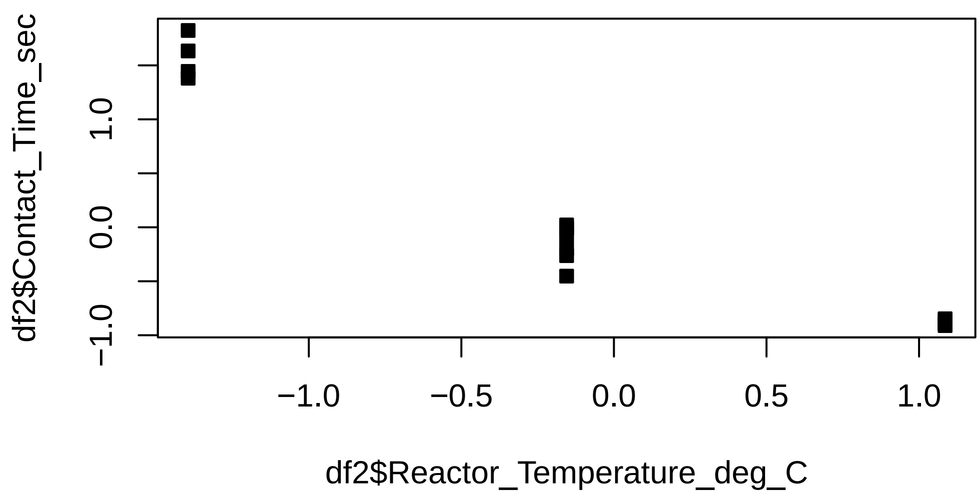
```
  Conversion_of_n_Heptane_to_Acetylene Reactor_Temperature_deg_C
1                                 49.0                      1300
2                                 50.2                      1300
3                                 50.5                      1300
4                                 48.5                      1300
5                                 47.5                      1300
6                                 44.5                      1300
  Ratio_of_H2_to_n_Heptane_mole_ratio Contact_Time_sec
1                                 7.5           0.0120
2                                 9.0           0.0120
3                                11.0           0.0115
4                                13.5           0.0130
5                                17.0           0.0135
6                                23.0           0.0120
```

```r
# For standardizing via Z scores
unit_norm=function(x){
  x=x-mean(x)
  return(sqrt(length(x)-1)*x/sqrt(sum(x^2)))
}

df2=df

#standardizing the regressors
df2[,2:4]=apply(df[,2:4],2,unit_norm)

df2=data.frame(df2)
corrplot::corrplot(cor(df2))
```

```
Warning in corrplot::corrplot(cor(df2)): Not been able to calculate text margin,
please try again with a clean new empty window using {plot.new(); dev.off()} or
reduce tl.cex
```




```
# Observe the high correlations between Contact time and Reactor temperature
plot(df2$Reactor_Temperature_deg_C,df2$Contact_Time_sec,pch=22,bg=1)
```



```r
orig=names(df2)
names(df2)=c('P','t','H','C')

df2$t2=df2$t^2
df2$H2=df2$H^2
df2$C2=df2$C^2

RS=lm(P~t+H+C+t*H+t*C+C*H+t2+H2+C2  ,df2)
summary(RS)
```

```
Call:
lm(formula = P ~ t + H + C + t * H + t * C + C * H + t2 + H2 + 
    C2, data = df2)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.3499 -0.3411  0.1297  0.5011  0.6720 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  35.8958     1.0916  32.884 5.26e-08 ***
t             4.0038     4.5087   0.888 0.408719    
H             2.7783     0.3071   9.048 0.000102 ***
C            -8.0423     6.0707  -1.325 0.233461    
t2          -12.5236    12.3238  -1.016 0.348741    
H2           -0.9727     0.3746  -2.597 0.040844 *  
C2          -11.5932     7.7063  -1.504 0.183182    
t:H          -6.4568     1.4660  -4.404 0.004547 ** 
t:C         -26.9804    21.0213  -1.283 0.246663    
H:C          -3.7681     1.6553  -2.276 0.063116 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9014 on 6 degrees of freedom
Multiple R-squared:  0.9977,    Adjusted R-squared:  0.9943 
F-statistic: 289.7 on 9 and 6 DF,  p-value: 3.225e-07
```

```r
# Crazy high VIF
car::vif(RS)
```

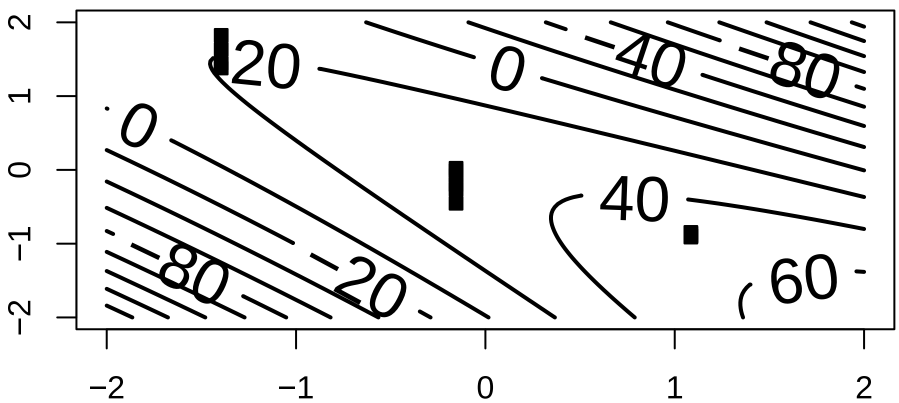
```
there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif

          t           H           C          t2          H2          C2 
 375.247759    1.740631  680.280039 1762.575365    3.164318 1156.766284 
        t:H         t:C         H:C 
  31.037059 6563.345193   35.611286 
```

```r
car::vif(RS,type='predictor')
```

```
GVIFs computed for predictors

           GVIF Df GVIF^(1/(2*Df)) Interacts With Other Predictors
t  51654.740516  6        2.470354           H, C       t2, H2, C2
H  51654.740516  6        2.470354           t, C       t2, H2, C2
C  51654.740516  6        2.470354           t, H       t2, H2, C2
t2  1762.575365  1       41.983037           --    t, H, C, H2, C2
H2     3.164318  1        1.778853           --    t, H, C, t2, C2
C2  1156.766284  1       34.011267           --    t, H, C, t2, H2
```

```r
# hidden extrapolation - be careful

# Create a grid to extrapolate over
c_grid=seq(-2,2,l=100)
t_grid=seq(-2,2,l=100)
g=expand.grid(t_grid,c_grid)

# Adding a fixed value of H=-0.6082179
g=cbind(g,rep(-0.6082179,nrow(g)))
# Adding H^2
g=cbind(g,g^2)

# Making new dataframe for predictions
new_dat=data.frame(g)
names(new_dat)=c("t","C","H","t2","C2","H2")

# Predicting the values on the grid
Z=predict.lm(RS,new_dat)

contour(c_grid,t_grid,matrix(Z, ncol = length(t_grid)),lwd=2,labcex=2,ylim=c(-1,1)*2,xlim=
```

```r
points(df2$t,df2$C,pch=22,bg=1)
```




```r
## Notice that as soon as we leave the region with data, the response becomes negative.
# Recall that it is a percentage and can't be negative
## This shows that mild extrapolation dangerous!
```

Example 8.3. Use the NFL data from the textbook - regress the number of wins against all variables. Check for multicollinearity. Propose a method to resolve the multicollinearity.

```r
#####################################################
#
#
#
#            NFL DATA EXAMPLE
#
#
#####################################################

df=MPV::table.b1
# Note
```

```r
df
```

```
    y   x1   x2   x3   x4  x5   x6   x7   x8   x9
1  10 2113 1985 38.9 64.7   4  868 59.7 2205 1917
2  11 2003 2855 38.8 61.3   3  615 55.0 2096 1575
3  11 2957 1737 40.1 60.0  14  914 65.6 1847 2175
4  13 2285 2905 41.6 45.3  -4  957 61.4 1903 2476
5  10 2971 1666 39.2 53.8  15  836 66.1 1457 1866
6  11 2309 2927 39.7 74.1   8  786 61.0 1848 2339
7  10 2528 2341 38.1 65.4  12  754 66.1 1564 2092
8  11 2147 2737 37.0 78.3  -1  761 58.0 1821 1909
9   4 1689 1414 42.1 47.6  -3  714 57.0 2577 2001
10  2 2566 1838 42.3 54.2  -1  797 58.9 2476 2254
11  7 2363 1480 37.3 48.0  19  984 67.5 1984 2217
12 10 2109 2191 39.5 51.9   6  700 57.2 1917 1758
13  9 2295 2229 37.4 53.6  -5 1037 58.8 1761 2032
14  9 1932 2204 35.1 71.4   3  986 58.6 1709 2025
15  6 2213 2140 38.8 58.3   6  819 59.2 1901 1686
16  5 1722 1730 36.6 52.6 -19  791 54.4 2288 1835
17  5 1498 2072 35.3 59.3  -5  776 49.6 2072 1914
18  5 1873 2929 41.1 55.3  10  789 54.3 2861 2496
19  6 2118 2268 38.2 69.6   6  582 58.7 2411 2670
20  4 1775 1983 39.3 78.3   7  901 51.7 2289 2202
21  3 1904 1792 39.7 38.1  -9  734 61.9 2203 1988
22  3 1929 1606 39.7 68.8 -21  627 52.7 2592 2324
23  4 2080 1492 35.5 68.8  -8  722 57.8 2053 2550
24 10 2301 2835 35.3 74.1   2  683 59.7 1979 2110
25  6 2040 2416 38.7 50.0   0  576 54.9 2048 2628
26  8 2447 1638 39.9 57.1  -8  848 65.3 1786 1776
27  2 1416 2649 37.4 56.3 -22  684 43.8 2876 2524
28  0 1503 1503 39.3 47.0  -9  875 53.5 2560 2241
```

```r
summary(df)
```

```
       y                x1             x2              x3              x4       
 Min.   : 0.000   Min.   :1416   Min.   :1414   Min.   :35.10   Min.   :38.10  
 1st Qu.: 4.000   1st Qu.:1896   1st Qu.:1714   1st Qu.:37.38   1st Qu.:52.42  
 Median : 6.500   Median :2111   Median :2106   Median :38.85   Median :57.70  
 Mean   : 6.964   Mean   :2110   Mean   :2127   Mean   :38.64   Mean   :59.40  
 3rd Qu.:10.000   3rd Qu.:2303   3rd Qu.:2474   3rd Qu.:39.70   3rd Qu.:68.80  
```

```
 Max.   :13.000   Max.   :2971   Max.   :2929   Max.   :42.30   Max.   :78.30  
       x5               x6               x7              x8      
 Min.   :-22.00   Min.   : 576.0   Min.   :43.80   Min.   :1457  
 1st Qu.: -5.75   1st Qu.: 710.5   1st Qu.:54.77   1st Qu.:1848  
 Median :  1.00   Median : 787.5   Median :58.65   Median :2050  
 Mean   :  0.00   Mean   : 789.9   Mean   :58.16   Mean   :2110  
 3rd Qu.:  6.25   3rd Qu.: 869.8   3rd Qu.:61.10   3rd Qu.:2320  
 Max.   : 19.00   Max.   :1037.0   Max.   :67.50   Max.   :2876  
       x9      
 Min.   :1575  
 1st Qu.:1913  
 Median :2101  
 Mean   :2128  
 3rd Qu.:2328  
 Max.   :2670  
```

```r
names(df)
```

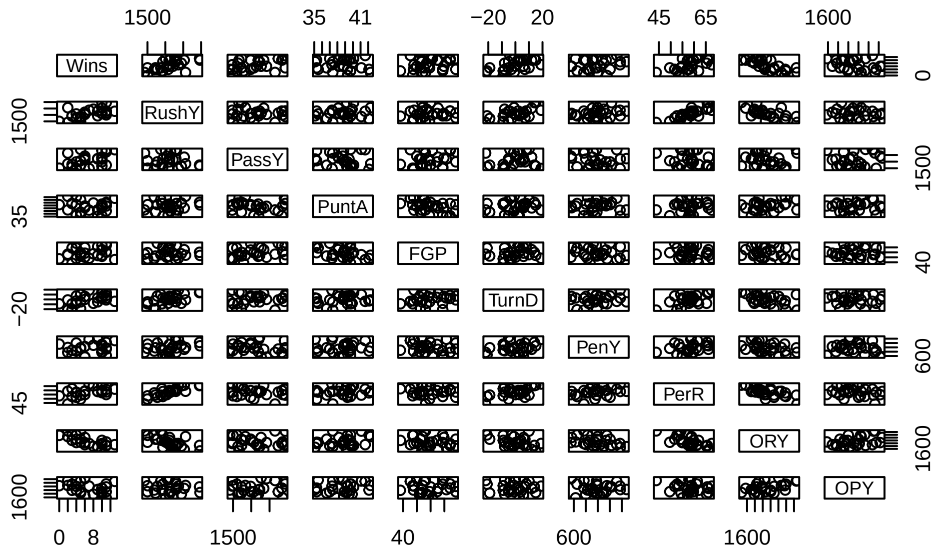
```
[1] "y"  "x1" "x2" "x3" "x4" "x5" "x6" "x7" "x8" "x9"
```

```r
names(df)=c("Wins","RushY","PassY",
            "PuntA","FGP","TurnD",
            "PenY","PerR","ORY","OPY")
summary(df)
```

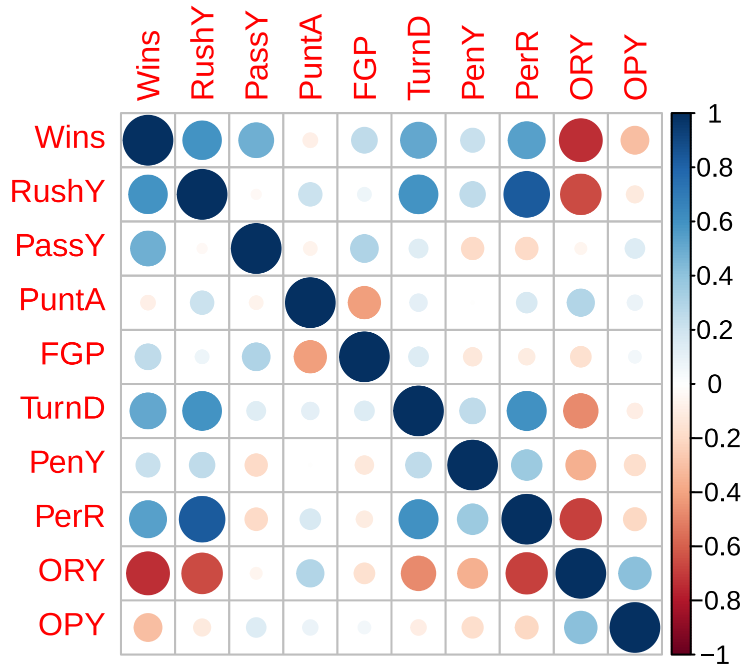
```
      Wins            RushY          PassY          PuntA            FGP       
 Min.   : 0.000   Min.   :1416   Min.   :1414   Min.   :35.10   Min.   :38.10  
 1st Qu.: 4.000   1st Qu.:1896   1st Qu.:1714   1st Qu.:37.38   1st Qu.:52.42  
 Median : 6.500   Median :2111   Median :2106   Median :38.85   Median :57.70  
 Mean   : 6.964   Mean   :2110   Mean   :2127   Mean   :38.64   Mean   :59.40  
 3rd Qu.:10.000   3rd Qu.:2303   3rd Qu.:2474   3rd Qu.:39.70   3rd Qu.:68.80  
 Max.   :13.000   Max.   :2971   Max.   :2929   Max.   :42.30   Max.   :78.30  
     TurnD             PenY             PerR            ORY      
 Min.   :-22.00   Min.   : 576.0   Min.   :43.80   Min.   :1457  
 1st Qu.: -5.75   1st Qu.: 710.5   1st Qu.:54.77   1st Qu.:1848  
 Median :  1.00   Median : 787.5   Median :58.65   Median :2050  
 Mean   :  0.00   Mean   : 789.9   Mean   :58.16   Mean   :2110  
 3rd Qu.:  6.25   3rd Qu.: 869.8   3rd Qu.:61.10   3rd Qu.:2320  
 Max.   : 19.00   Max.   :1037.0   Max.   :67.50   Max.   :2876  
      OPY      
```

```
Min.   :1575
1st Qu.:1913
Median :2101
Mean   :2128
3rd Qu.:2328
Max.   :2670
```

```
plot(df)
```




```
model=lm(Wins~.,data=df)
################## Multicollinearity

###### Corr plot
corrplot::corrplot(cor(df))
```




```r
###### VIF
car::vif(model)
```

```
   RushY     PassY     PuntA       FGP     TurnD      PenY      PerR       ORY 
4.827645 1.420161 2.126597 1.566107 1.924035 1.275979 5.414572 4.535643 
     OPY 
1.423390 
```

```r
# recalculating via the X'X matrix
X=model.matrix(model)
dim(X)
```

```
[1] 28 10
```

```r
#standardizing
X2=apply(X,2,function(x){(x-mean(x))/sqrt(sum((x-mean(x))^2))}); dim(X2)
```

```
[1] 28 10
```

```r
#replacing with column of ones again
X2[,1]=X[,1]
X=X2
diag(solve(t(X)%*%X))
```

```
(Intercept)       RushY       PassY       PuntA         FGP       TurnD
 0.03571429  4.82764538  1.42016105  2.12659726  1.56610698  1.92403474
       PenY        PerR         ORY         OPY
 1.27597850  5.41457162  4.53564335  1.42338989
```

```r
# observe that
model=lm(RushY~.,data=df[,-1])
s=summary(model)
(1-s$r.squared)^(-1)
```

```
[1] 4.827645
```

```r
###### Condition Number / Eigenvalue / Eigenvector
X=model.matrix(model)
dim(X)
```

```
[1] 28  9
```

```r
#standardizing
X2=apply(X,2,function(x){(x-mean(x))/sqrt(sum((x-mean(x))^2))}); dim(X2)
```

```
[1] 28  9
```

```r
X2[,1]=X[,1]
X=X2


ev=eigen(t(X)%*%X)
xev=ev$values
condition_number=max(xev)/min(xev)
condition_number
```

```
[1] 215.646
```

```r
sort(xev)
```

```
[1]  0.1298424  0.3552922  0.5328487  0.6867363  0.8232325  1.2191453  1.6977743
[8]  2.5551283 28.0000000
```

```r
#index of minimum eigenvalue
minn=which.min(xev)
ev$vectors[,minn]
```

```
[1] -1.675215e-16 -2.109094e-01  2.615778e-01 -4.250523e-02  1.442042e-01
[6] -4.744502e-02 -6.475229e-01 -6.423531e-01  1.741790e-01
```

```r
rownames(ev$vectors)=names(model$coefficients)
ev$vectors
```

```
                     [,1]          [,2]          [,3]          [,4]
(Intercept)  1.000000e+00 -3.434947e-17  5.656001e-17  1.332883e-16
PassY       -1.931673e-17  8.040232e-02  4.548137e-01 -4.590825e-01
PuntA        1.733313e-16  2.847481e-02 -5.282204e-01 -5.050070e-01
FGP          6.643674e-18 -1.498563e-02  6.360469e-01 -4.652414e-02
TurnD        1.358771e-18 -4.436052e-01  9.254740e-02 -4.700157e-01
PenY        -4.747269e-17 -3.638914e-01 -1.751243e-01  1.469074e-01
PerR        -4.843659e-17 -5.401530e-01 -1.437332e-01 -1.847958e-01
ORY         -9.496374e-17  5.364518e-01 -2.217080e-01 -1.153371e-01
OPY          1.353659e-16  2.893988e-01  2.291129e-02 -4.920345e-01
                     [,5]          [,6]          [,7]          [,8]
(Intercept)  1.753269e-17  4.497241e-17 -8.295922e-18  9.898359e-17
PassY        3.414787e-01  5.912719e-01  2.410126e-01 -8.451722e-02
PuntA        2.835269e-01  3.495064e-02 -3.785463e-01 -4.145009e-01
FGP         -1.687798e-01 -2.417375e-01 -6.140270e-01 -3.567854e-01
TurnD       -9.900605e-03 -1.291665e-01 -2.199263e-01  6.984198e-01
PenY        -5.210958e-01  6.822653e-01 -2.680418e-01 -6.487362e-02
PerR        -4.221330e-02 -3.015818e-01  2.274534e-01 -2.994520e-01
ORY         -9.681973e-02  2.538335e-02 -3.566234e-01  3.161428e-01
OPY         -7.012299e-01 -1.302810e-01  3.499387e-01 -1.101525e-01
                     [,9]
(Intercept) -1.675215e-16
PassY       -2.109094e-01
PuntA        2.615778e-01
```

```
FGP         -4.250523e-02
TurnD        1.442042e-01
PenY        -4.744502e-02
PerR        -6.475229e-01
ORY         -6.423531e-01
OPY          1.741790e-01
```

```r
# names(model$coefficients)[abs(round(ev$vectors[,minn],1))>0]
round(ev$vectors[,minn],1)
```

```
(Intercept)       PassY       PuntA         FGP       TurnD        PenY
        0.0        -0.2         0.3         0.0         0.1         0.0
       PerR         ORY         OPY
       -0.6        -0.6         0.2
```

```r
# What should we do?

df$RushDiff=(df$RushY-df$ORY)
# df$RushDiffPer=df$RushY/(df$RushY+df$ORY)

# paste(names(df),collapse='+')

model=lm(Wins~PassY+PuntA+FGP+TurnD+PenY+PerR+OPY+RushDiff,data=df)
summary(model)
```

```
Call:
lm(formula = Wins ~ PassY + PuntA + FGP + TurnD + PenY + PerR +
    OPY + RushDiff, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6976 -0.8333  0.0823  0.7845  2.7968

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.5581613 12.7312595  -0.515 0.612409
PassY        0.0037880  0.0008196   4.622 0.000186 ***
PuntA       -0.0208741  0.2052446  -0.102 0.920057
FGP          0.0249704  0.0407274   0.613 0.547073
```

```
TurnD       -0.0029187  0.0465199  -0.063 0.950628
PenY         0.0021519  0.0031747   0.678 0.506053
PerR         0.1388836  0.1504572   0.923 0.367542
OPY         -0.0023446  0.0012749  -1.839 0.081587 .
RushDiff     0.0023289  0.0011199   2.080 0.051347 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.822 on 19 degrees of freedom
Multiple R-squared:  0.8071,    Adjusted R-squared:  0.7258
F-statistic: 9.935 on 8 and 19 DF,  p-value: 2.305e-05
```

```r
corrplot::corrplot(cor(df))
car::vif(model)
```

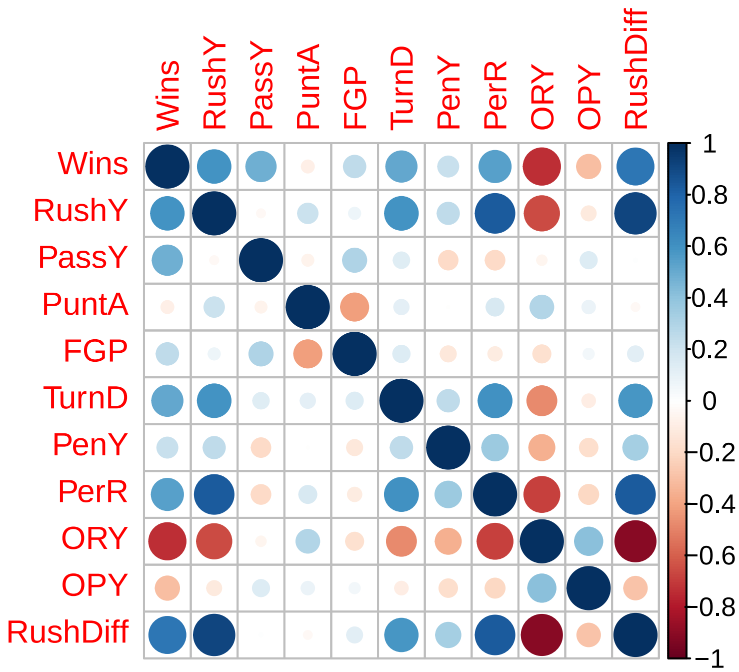
```
   PassY    PuntA      FGP    TurnD     PenY     PerR      OPY RushDiff
1.360875 1.347757 1.514088 1.914974 1.230133 5.347319 1.162295 4.774672
```

```r
model=lm(Wins~PassY+PuntA+FGP+TurnD+PenY+OPY+RushDiff,data=df)
summary(model)
```

```
Call:
lm(formula = Wins ~ PassY + PuntA + FGP + TurnD + PenY + OPY +
    RushDiff, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-3.9325 -0.9011 -0.1062  0.9020  3.0495

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.8643720  9.8338943   0.088 0.930833
PassY        0.0035079  0.0007586   4.624 0.000164 ***
PuntA        0.0143946  0.2009097   0.072 0.943594
FGP          0.0156562  0.0393115   0.398 0.694659
TurnD        0.0114784  0.0436650   0.263 0.795336
PenY         0.0023024  0.0031587   0.729 0.474510
OPY         -0.0021934  0.0012596  -1.741 0.096996 .
RushDiff     0.0031495  0.0006787   4.641 0.000158 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.815 on 20 degrees of freedom
Multiple R-squared:  0.7984,	Adjusted R-squared:  0.7279
F-statistic: 11.32 on 7 and 20 DF,  p-value: 9.485e-06
```

```
corrplot::corrplot(cor(df))
```




```
car::vif(model)
```

```
   PassY    PuntA      FGP    TurnD     PenY      OPY RushDiff 
1.174416 1.301051 1.421150 1.699715 1.226886 1.143100 1.766523 
```

There are four primary sources/causes of multicollinearity :

- The data collection method employed: In this case, some of the variables are typically confounded and/or the experiment/study was not well-designed.
- Constraints on the model or in the population: Sometimes, only certain combinations of levels of variables can be observed together.

- Model specification: Sometimes you have two or more variables in your model that are measuring the same thing.
- An overdefined model: You have too many variables in your model.

Make sure to check for each of these. Sometimes, regression coefficients have the wrong sign. This is likely due to one of the following:

- The range of some of the regressors is too small – if the range of some of the regressors is too small, then the variance of $\widehat{\beta}$ is high.
- Important regressors have not been included in the model.
- **Multicollinearity is present.**
- Computational errors have been made.

Multicollinearity can be cured with:

1. more data (lol often not possible), 2, model re-specification: Can you include a function of the variables that preserves the information, but aren't linearly dependent? Can you remove a variable?
2. Or, a modified version of regression, one of Lasso, ridge or elastic net regression.

8.3 Homework questions

Complete the Chapter 9 textbook questions.

Exercise 8.1. Check for multicollinearity in all of our past examples.

Exercise 8.2. Summarize the 3 multicollinearity diagnostics.

9 Variable/Model Selection

9.1 Variable Selection

In the preceding lessons we have assumed that the regressor variables included in the model are known to be important. Our focus was on techniques to ensure that the functional form of the model was correct and that the underlying assumptions were not violated. In previous lessons, we have employed the classical approach to regression model selection, which assumes that we have a very good idea of the basic form of the model and that we know all (or nearly all) of the regressors that should be used.

Our approach so far can be summarized as follows:

- Fit the full model.
- Perform a thorough analysis of this model, including a full residual analysis and investigation of multicollinearity.
- Determine if transformations of the response or of some of the regressors are necessary.
- Use the t-tests/F-tests on the individual regressors to edit the model.
- Perform a thorough analysis of the edited model, especially a residual analysis, to determine the model's adequacy.

In many problems the analyst has a rather large pool of possible candidate regressors, of which only a few are likely to be important. Finding an appropriate subset of regressors for the model is often called the **variable selection** problem .

> **Note**
>
> Variable selection can address multicollinearity, through the removal of unnecessary variables.

Building a regression model that includes only a subset of the available regressors involves two conflicting objectives:

1. We would like the model to include as many regressors as possible so that the information about Y contained in these factors can influence the predicted value of Y.
2. We want the model to include as few regressors as possible because the variance of $\hat{Y}$ increases as the number of regressors increases. Also the more regressors there are in a model, the greater the costs of data collection and model maintenance.

Therefore, we seek a model that is a compromise between these two objectives. There are several algorithms that can be used for variable selection, and these procedures frequently specify different subsets of the candidate regressors as best.

Variable selection is often developed in an idealized setting: Assumed that the correct functional specification of the regressors is known, and no outliers are present. In practice – it's messy. Residual analysis is useful in revealing functional forms of regressors that might be investigated, in pointing out new candidate regressors, and for identifying defects in the data such as outliers. The effect of influential observations should also be determined. Although ideally these problems should be solved simultaneously, an iterative approach is often employed, in which (1) a particular variable selection strategy is employed and then (2) the resulting subset model is checked for correct functional specification, outliers, and influential observations. This may indicate that step 1 must be repeated. Several iterations may be required to produce an adequate model.

None of the variable selection procedures described are guaranteed to produce the best model for a given data set. In fact, there is usually not a single best model but rather several equally good ones. Variable selection algorithms are heavily algorithmic so it is tempting to place a lot of confidence in the results of a particular procedure. Beware of this - experience, professional judgment and subjective considerations are also critical to the variable selection problem. Variable selection procedures should be used in conjunction with these.

Two key aspects of the variable selection problem are:

1. Generating the subset models.
2. Deciding if one subset is better than another.

9.2 Deciding if one subset is better than another

9.2.1 Coefficients of determination

Recall that

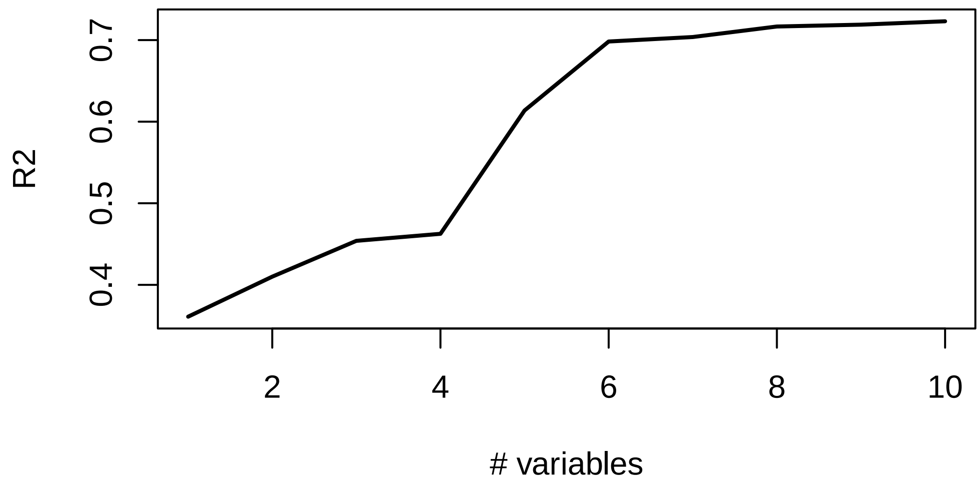
$$R^2 = \frac{SSModel}{SST} = 1 - \frac{SSE}{SST}.$$

We know that R^2 increases as p increases, so we cannot choose the model with the largest R^2. However, we could add variables one at a time until the marginal increase in R^2 is small.

Example 9.1. Recall example Example 6.6. Investigate the R^2 curve through adding the variables one at a time.

```
custom_palette = c(
  "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728",
  "#9467bd", "#8c564b", "#e377c2", "#7f7f7f",
```

```
  "#bcbd22", "#17becf", "#393b79",
  "#8c6d31", "#9c9ede", "#637939", "#eb348f"
)

df_clean4$d_3=df_clean4$District==3
var_list=names(df_clean4)[-c(11,12)]
df2=df_clean4[-11142,]
#previous...
full_model=lm(Sale_price~ District +  Extwall +
            Stories  +    Year_Built  + Fin_sqft  +
            Units    +     Bdrms    +
            Fbath  +    Lotsize    + Sale_date +District*Lotsize,df2)



###########

# R2
r2=c()
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
  model2=lm(Sale_price~.,df2[,vars])
  sm=summary(model2)
  r2=c(r2,sm$r.squared)
}
plot(r2,type='l',lwd=2,ylab="R2",xlab="# variables")
```




```r
var_list[1:6]
```

```
[1] "District"    "Extwall"     "Stories"     "Year_Built" "Fin_sqft"
[6] "Units"
```

Notice how the curve levels off at 6 variables. This would indicate that we are not gaining much explained variation beyond the first 6 variables. These variables are printed above.

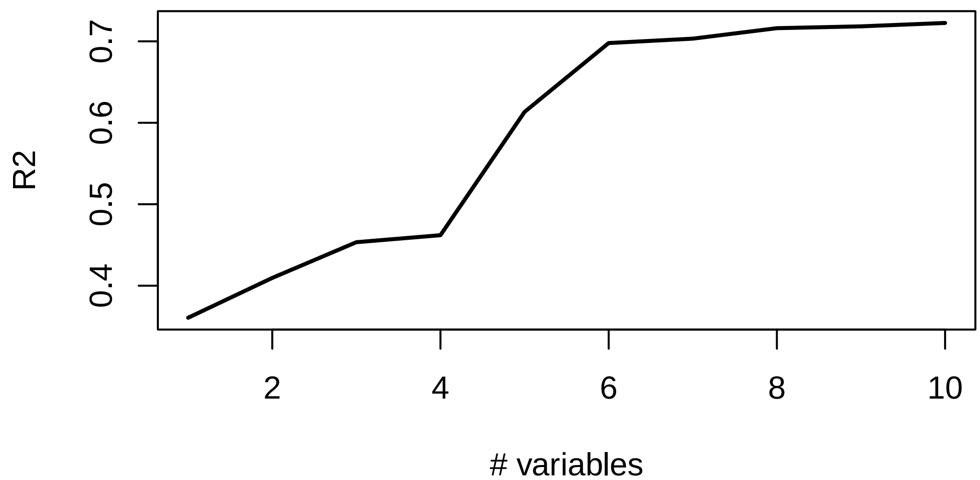
Recall that

$$\bar{R}^2 = 1 - \frac{n-1}{n-p}(1 - \bar{R}^2).$$

The $\bar{R}^2$ statistic does not necessarily increase as additional regressors are introduced into the model. In fact, it can be shown that if s regressors are added to the model, the new $\bar{R}^2$ will exceed the original $\bar{R}^2$ if and only if the partial F statistic for testing the significance of the s additional regressors exceeds 1. Consequently, one criterion for selection of an optimum subset model is to choose the model that has a maximum $\bar{R}^2$. In addition, note that maximizing $\bar{R}^2$ is equivalent to choosing the model with the minimum MSE.

```r
# R2 adjusted
r2=c()
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
```

```
  model2=lm(Sale_price~.,df2[,vars])
  sm=summary(model2)
  r2=c(r2,sm$adj.r.squared)
}
plot(r2,type='l',lwd=2,ylab="R2",xlab="# variables")
```




```
var_list[1:6]
```

```
[1] "District"   "Extwall"    "Stories"    "Year_Built" "Fin_sqft"
[6] "Units"
```

9.2.2 Mallows C_k criterion

Consider the complete model:

$$(C)\ Y = \beta_1 + \beta_2 X_1 + \cdots + \beta_p X_p + \epsilon$$

and a reduced model with $(k-1)$ explanatory variables

$$Y = \beta_1 + \beta_1 X_1 + \cdots + \beta_k X_k + \epsilon.$$

Mallows (1973) defined the C_k statistic as

$$C_k = \frac{SSE_R}{MSE_C} - (n - 2k)$$

where k is the number of columns in X in the reduced model.

Observe that if $k = p$ then

$$C_k = C_p = dfE_C - [n - 2p] = p = k.$$

Let Bias $= \mathrm{E}\left[\hat{Y}_i\right] - \mathrm{E}\left[Y_i\right]$ where $\hat{Y}_i$ are generated from the reduced model. It can be shown that

$$E\left[C_k \mid \text{Bias} = 0\right] = k.$$

That is, if the regression model based on the subset of $k-1$ predictors is unbiased, then we expect C_k to be roughly k. In addition, if there is bias, then $E[C_k] > k$. Therefore, it is suggested that from all possible models, we choose the model with C_k close to k **and** $C_k \leq k$. This is a littble bit subjective: For example, the smallest C_k is $C_4 = 4.1$ and the next smallest is $C_5 = 4.7$. C_4 is smallest but $C_4 > 4$ – the model may be biased. Now, C_5 is the second smallest, $C_5 < 5$ implies the model is likely not biased.

Example 9.2. Recall example Example 6.6. Investigate the Mallows C_k of the models considered in Example 9.9. In addition, compare the `model_1` and `model_2` below, using `full_model` as the complete model.

```
# Mallows C
# install.packages('olsrr')

# Recall the complete model
full_model=lm(Sale_price~ District +  Extwall +
             Stories   +    Year_Built  + Fin_sqft   +
             Units    +     Bdrms     +
             Fbath   +    Lotsize    + Sale_date +District*Lotsize,df2)


# Potential models
model_1=lm(Sale_price~ District +  Fin_sqft   +  Sale_date ,df2)

model_2=lm(Sale_price~ District +  Extwall +
                 Stories   +    Year_Built  + Fin_sqft   +
                 Units    +     Bdrms     +
                 Fbath   +    Lotsize    + Sale_date ,df2)
```

```
olsrr::ols_mallows_cp(model_1,full_model)
```

```
[1] 17131
```

```
olsrr::ols_mallows_cp(model_2,full_model)
```

```
[1] 3505.586
```

```
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
  model=lm(Sale_price~.,df_clean4[,vars])
  print(paste0("k" ,i," mallows C_k ",
    olsrr::ols_mallows_cp(model,full_model),sep=""))
}
```

```
[1] "k1 mallows C_k 40548.5008352938"
[1] "k2 mallows C_k 35586.0286701987"
[1] "k3 mallows C_k 31143.8465791983"
[1] "k4 mallows C_k 30228.81611156"
[1] "k5 mallows C_k 14773.9666340633"
[1] "k6 mallows C_k 6191.00700048948"
[1] "k7 mallows C_k 5624.03575517497"
[1] "k8 mallows C_k 4298.06818957892"
[1] "k9 mallows C_k 4039.89603591008"
[1] "k10 mallows C_k 3622.28930435513"
```

If every potential model has a high value for Mallows C_k, this is an indication that some important predictor variables are likely missing from each model. Therefore, we may wish to revaluate the real estate model.

Example 9.3. Use the NFL data from the textbook - Consider the following models provided below. Select the best model from `model`, `model_1` and `model_2` given below, using each of the coefficient of determination criteria and Mallows C_k.

```
df=MPV::table.b1
# Note
names(df)=c("Wins","RushY","PassY",
```

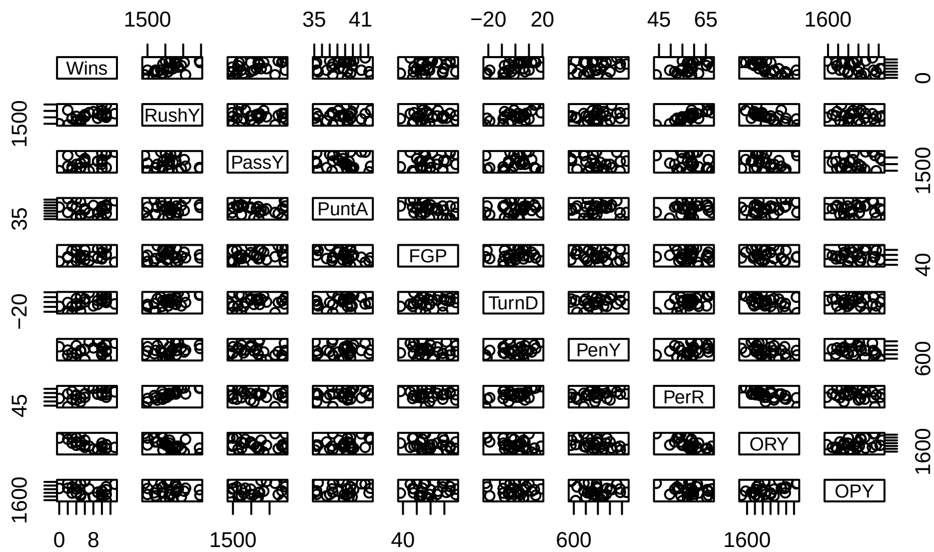
```r
              "PuntA","FGP","TurnD",
              "PenY","PerR","ORY","OPY")
summary(df)
```

```
      Wins            RushY           PassY           PuntA            FGP       
 Min.   : 0.000   Min.   :1416   Min.   :1414   Min.   :35.10   Min.   :38.10  
 1st Qu.: 4.000   1st Qu.:1896   1st Qu.:1714   1st Qu.:37.38   1st Qu.:52.42  
 Median : 6.500   Median :2111   Median :2106   Median :38.85   Median :57.70  
 Mean   : 6.964   Mean   :2110   Mean   :2127   Mean   :38.64   Mean   :59.40  
 3rd Qu.:10.000   3rd Qu.:2303   3rd Qu.:2474   3rd Qu.:39.70   3rd Qu.:68.80  
 Max.   :13.000   Max.   :2971   Max.   :2929   Max.   :42.30   Max.   :78.30  
     TurnD             PenY             PerR            ORY      
 Min.   :-22.00   Min.   : 576.0   Min.   :43.80   Min.   :1457  
 1st Qu.: -5.75   1st Qu.: 710.5   1st Qu.:54.77   1st Qu.:1848  
 Median :  1.00   Median : 787.5   Median :58.65   Median :2050  
 Mean   :  0.00   Mean   : 789.9   Mean   :58.16   Mean   :2110  
 3rd Qu.:  6.25   3rd Qu.: 869.8   3rd Qu.:61.10   3rd Qu.:2320  
 Max.   : 19.00   Max.   :1037.0   Max.   :67.50   Max.   :2876  
      OPY      
 Min.   :1575  
 1st Qu.:1913  
 Median :2101  
 Mean   :2128  
 3rd Qu.:2328  
 Max.   :2670  
```

```r
head(df)
```

```
  Wins RushY PassY PuntA  FGP TurnD PenY PerR  ORY  OPY
1   10  2113  1985  38.9 64.7     4  868 59.7 2205 1917
2   11  2003  2855  38.8 61.3     3  615 55.0 2096 1575
3   11  2957  1737  40.1 60.0    14  914 65.6 1847 2175
4   13  2285  2905  41.6 45.3    -4  957 61.4 1903 2476
5   10  2971  1666  39.2 53.8    15  836 66.1 1457 1866
6   11  2309  2927  39.7 74.1     8  786 61.0 1848 2339
```

```r
plot(df)
```




```
df$RushDiff=(df$RushY-df$ORY)

model=lm(Wins~PassY+PuntA+FGP+TurnD+PenY+OPY+RushDiff,data=df)
model_1=lm(Wins~ TurnD+PenY+PerR+ORY+OPY ,df)
model_2=lm(Wins~ TurnD+PenY+ORY+OPY,df)




# 3 ADJ R2
summary(model_1)$r.squared
```

```
[1] 0.5848535
```

```
summary(model_2)$r.squared
```

```
[1] 0.5843435
```

```
summary(model)$r.squared
```

```
[1] 0.7984089
```

```r
# 3 ADJ R2
summary(model_1)$adj.r.squared
```

```
[1] 0.490502
```

```r
summary(model_2)$adj.r.squared
```

```
[1] 0.5120555
```

```r
summary(model)$adj.r.squared
```

```
[1] 0.7278521
```

```r
##### Mallows C
olsrr::ols_mallows_cp(model_1,model) # k=6
```

```
[1] 25.187
```

```r
olsrr::ols_mallows_cp(model_2,model) # k=5
```

```
[1] 23.23759
```

```r
olsrr::ols_mallows_cp(model,model)
```

```
[1] 8
```

We would select `model` as the final model, since it has the lowest values of the Mallows C_k that is close to the number of variables. In addition, the R^2 jumps significantly when moving from either `model_1` or `model_2` to `model`, so again, we would select `model`. Lastly, the adjusted \$R^2 is the highest for `model`.

9.3 Akaike information criterion

- Akaike proposed an information criterion, AIC, based on maximizing the expected *entropy* of the model.
- Entropy is simply a measure of the expected information, in this case the Kullback – Leibler divergence.
- $$AIC = n \log \frac{SSE}{n} + 2p + constant$$
- The decrease in SSE is balanced by the $2p$ penalty
- AIC can be computed for other models, but should only be compared within model classes

9.4 Bayesian information criterion/Schwartz information criterion

- Sawa and Schwartz extended AIC, both called BIC.
- This criterion places a greater penalty on adding regressors as the sample size increases.
- $$BIC = n \log \frac{SSE}{n} + p \log n + constant$$
- BIC can be computed for other models, but should only be compared within model classes

Example 9.4. Recall example Example 6.6. Investigate the AIC and BIC of the models considered in Example 9.9. In addition, compare the `model_1`, `model_2` and the `full_model` using AIC and BIC.

```
# Mallows C
# install.packages('olsrr')

# Recall the complete model
full_model=lm(Sale_price~ District +  Extwall +
            Stories  +    Year_Built  + Fin_sqft  +
            Units    +     Bdrms    +
            Fbath  +    Lotsize    + Sale_date +District*Lotsize,df_clean4)


# Potential models
model_1=lm(Sale_price~ District +  Fin_sqft   +  Sale_date ,df_clean4)

model_2=lm(Sale_price~ District +  Extwall +
```

```r
                  Stories   +    Year_Built  + Fin_sqft  +
                  Units    +     Bdrms    +
                  Fbath  +    Lotsize    + Sale_date ,df_clean4)

# Aic
AIC(full_model)
```

```
[1] 589136.3
```

```r
AIC(model_1)
```

```
[1] 602019.6
```

```r
AIC(model_2)
```

```
[1] 592279.3
```

```r
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
  model=lm(Sale_price~.,df_clean4[,vars])
  print(paste0("k" ,i," AIC ",
               AIC(model),sep=""))
}
```

```
[1] "k1 AIC 612758.302056694"
[1] "k2 AIC 610831.898564347"
[1] "k3 AIC 608957.605436705"
[1] "k4 AIC 608552.940741352"
[1] "k5 AIC 600431.017183595"
[1] "k6 AIC 594395.93413908"
[1] "k7 AIC 593955.469235904"
[1] "k8 AIC 592860.729755852"
[1] "k9 AIC 592640.317423882"
[1] "k10 AIC 592279.290409805"
```

```r
# Bic
BIC(full_model)
```

```
[1] 589638.7
```

```
BIC(model_1)
```

```
[1] 602165.5
```

```
BIC(model_2)
```

```
[1] 592668.3
```

```
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
  model=lm(Sale_price~.,df_clean4[,vars])
  print(paste0("k" ,i," AIC ",
               BIC(model),sep=""))
}
```

```
[1] "k1 AIC 612887.96764318"
[1] "k2 AIC 611026.396944075"
[1] "k3 AIC 609176.416113899"
[1] "k4 AIC 608779.855517702"
[1] "k5 AIC 600666.0360591"
[1] "k6 AIC 594655.265312051"
[1] "k7 AIC 594287.737301273"
[1] "k8 AIC 593233.518316997"
[1] "k9 AIC 593021.210084183"
[1] "k10 AIC 592668.287169261"
```

We would select `full_model` as the final model, since it has the lowest values of both AIC and BIC.

Example 9.5. Use the NFL data from the textbook - Consider the following models provided below. Select the best model from `model`, `model_1` and `model_2` given below, using each of AIC and BIC.

```
model=lm(Wins~PassY+PuntA+FGP+TurnD+PenY+OPY+RushDiff,data=df)
model_1=lm(Wins~ TurnD+PenY+PerR+ORY+OPY ,df)
model_2=lm(Wins~ TurnD+PenY+ORY+OPY,df)

# Bic
```

```r
BIC(model)
```

```
[1] 133.4221
```

```r
BIC(model_1)
```

```
[1] 146.9846
```

```r
BIC(model_2)
```

```
[1] 143.6868
```

```r
# Aic
AIC(model)
```

```
[1] 121.4323
```

```r
AIC(model_1)
```

```
[1] 137.6592
```

```r
AIC(model_2)
```

```
[1] 135.6936
```

We would select `model` as the final model, since it has the lowest values of both AIC and BIC.

9.5 Algorithms for model selection

We now discuss algorithms which auto select the "best" model.

9.5.1 All subsets regression

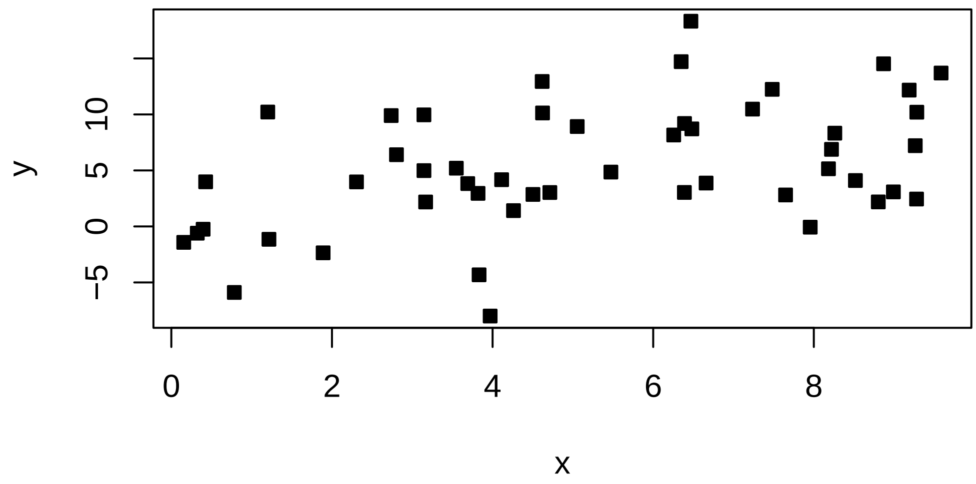
All subsets regression, also known as best subsets regression, is one way to select the best subset of regressors. This technique proceeds as the name implies - it involves evaluating all possible combinations of regressors to identify the model that best fits the data according to a chosen criterion. For instance, if the criterion is AIC, then we may compute the AIC for all possible 2^p models. All subsets regression would choose the model with the lowest AIC. Any of the four criterion (AIC, BIC, Mallows C_k or adjusted R^2) can be used for all subsets regression.

All subsets regression considers all possible subsets of regressors, ensuring that the best model is not missed by the algorithm. However, this has some downsides. First, it can be very computationally intensive for data sets with a large number of regressors. Second, if not properly controlled, examining many models can lead to **overfitting**, especially with small data sets.

Overfitting is when the model fits the data set so well, that it misses the underlying relationship between the independent and dependent variables. This results in a model that fits the data set well, but performs poorly at predicting new, unseen data due to its overly complex structure.

We can observe this in the example below:

```
library(ggplot2)

set.seed(4265)

# Generate sample data
n = 50
x = runif(n, min = 0, max = 10)
y = 1 + x + rnorm(n, sd = 5)

# Create a data frame
data = data.frame(x = x, y = y)
plot(data,pch=22,bg=1)
```



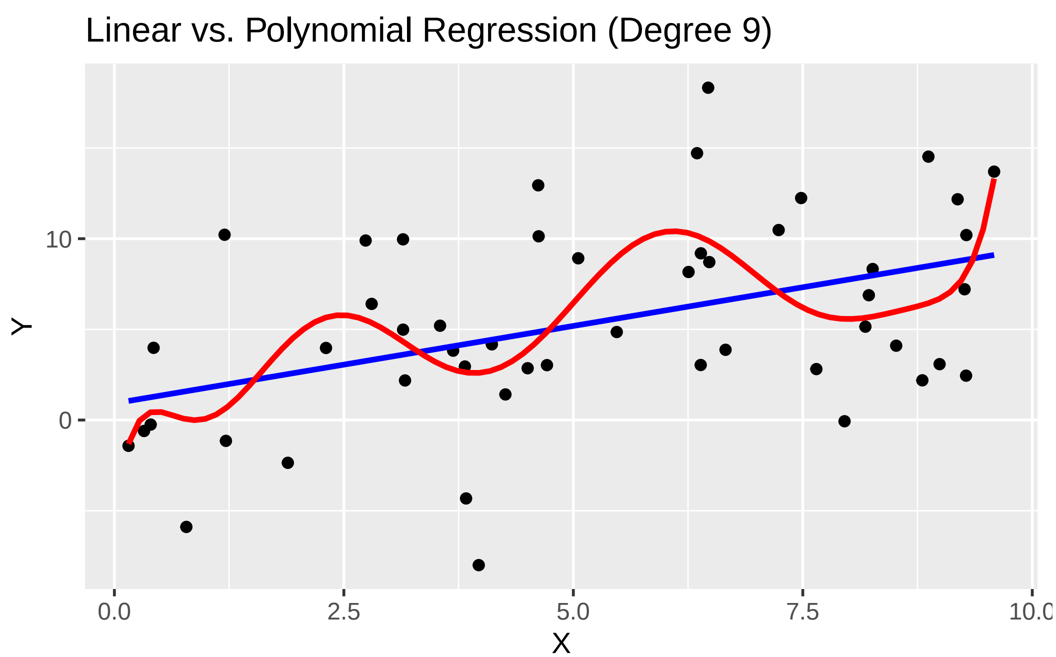

```
# complexity of the model
complexity=9

# Fit a linear regression model
linear_model = lm(y ~ x, data = data)

# Fit a polynomial regression model (degree = complexity) This model has more terms.
poly_model = lm(y ~ poly(x, complexity), data = data)

# Make predictions
linear_predictions = predict(linear_model, data)
poly_predictions = predict(poly_model, data)

# Calculate mean squared error (MSE)
linear_mse = mean((data$y - linear_predictions)^2)
poly_mse = mean((data$y - poly_predictions)^2)

# Print MSE values
cat("Linear Model MSE:", linear_mse, "\n")
```

```
Linear Model MSE: 24.03022
```

```
cat("Polynomial Model MSE:", poly_mse, "\n")
```

```
Polynomial Model MSE: 19.69118
```

```
# Notice how the squared error is lower for the more complex model?


# Plot the results - this is GG plot, another way to plot in R
ggplot(data, aes(x, y)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue", formula = y ~ x) +
  geom_smooth(method = "lm", se = FALSE, color = "red", formula = y ~ poly(x,complexity))
  labs(title = "Linear vs. Polynomial Regression (Degree 9)",
       x = "X", y = "Y")
```




```
# Generate new data from the same process
N = 500
xx = runif(n, min = 0, max = 10)
yy = 1 + x + rnorm(n, sd = 5)
nd = data.frame('x'=xx,'y'=yy)
```

```r
gen_linear_predictions = predict(linear_model, nd)
gen_poly_predictions = predict(poly_model, nd)

# Calculate mean squared error (MSE)
gen_linear_mse = mean((data$y - gen_linear_predictions)^2)
gen_poly_mse = mean((data$y - gen_poly_predictions)^2)

# Print MSE values
cat("Linear Model MSE:", gen_linear_mse, "\n")
```

```
Linear Model MSE: 45.06416
```

```r
cat("Polynomial Model MSE:", gen_poly_mse, "\n")
```

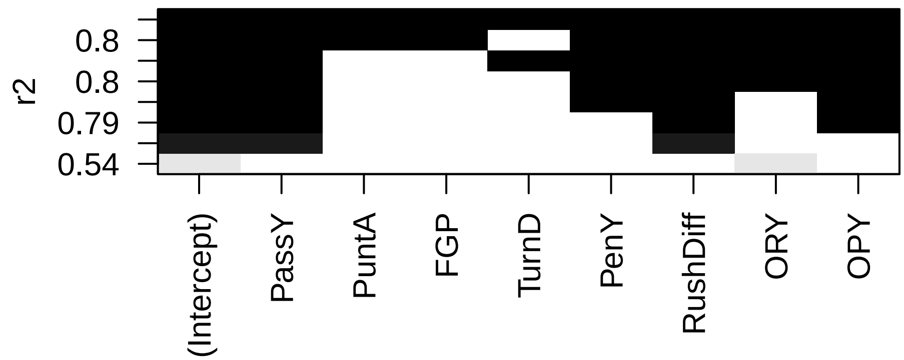
```
Polynomial Model MSE: 50.40172
```

```r
# Notice how the generalized error is lower for the linear model?
```

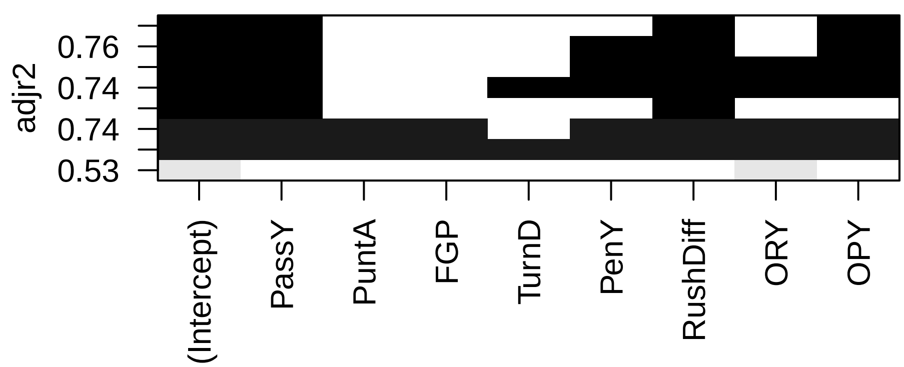
Above, we fit two models to the data. One that allows for polynomial functions of degree at most 9 (not just linear functions), and one that only allows linear functions. It is clear that the polynomial model has a lower error on the data set at hand. However, when we generate new data from the same process, the polynomial model has a much higher error. This is an example of overfitting - the complex model does not generalize well beyond the current data set.

Example 9.6. Use the NFL data from the textbook - Perform all subsets regression with each of the metrics described above.

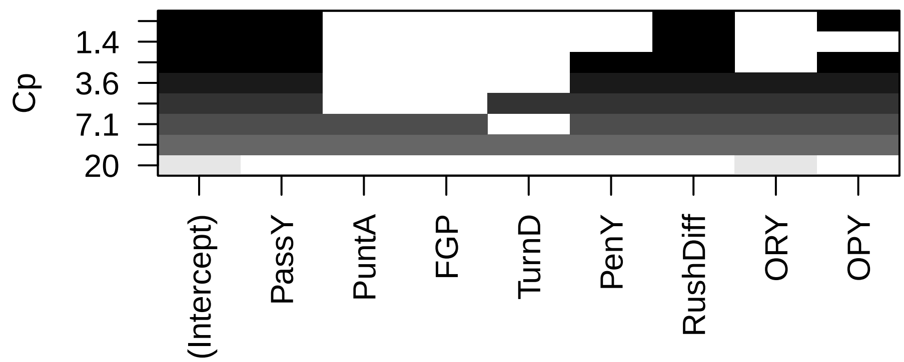
```r
# install.packages('leaps')

all=leaps::regsubsets(Wins~PassY+PuntA+FGP+TurnD+PenY+RushDiff+ORY+OPY,data=df,nvmax=10,me

# 2^8

plot(all,scale='r2')
```



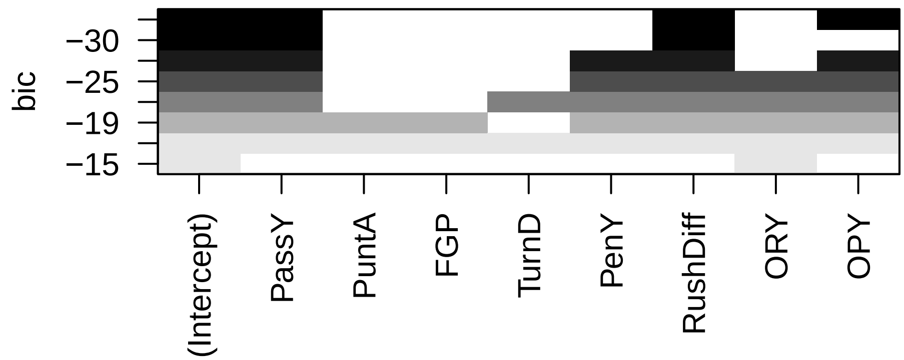

```r
plot(all,scale='adjr2')
```



```r
plot(all,scale='Cp')
```




```r
plot(all,scale='bic')
```

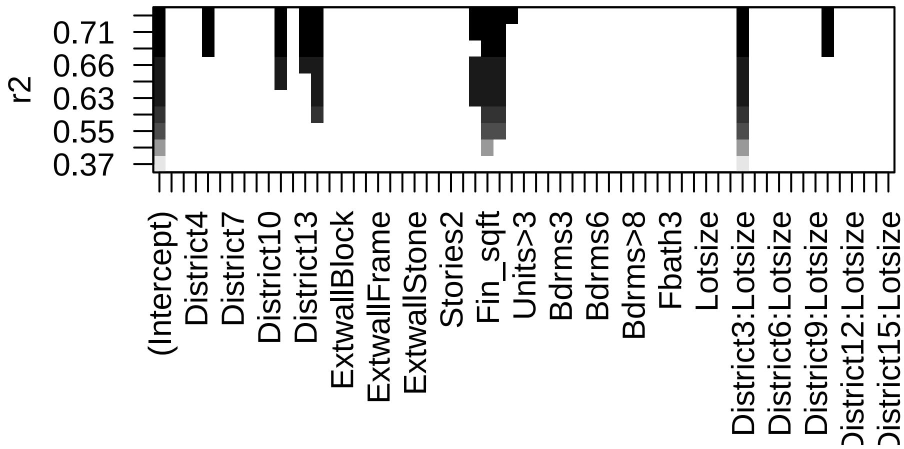


The `leaps` package allows us to do all subsets regression. We see that (with the exception of the R-squared) the best model under all criterion contains passing yards, rushing yard differential, and opponent passing yards.

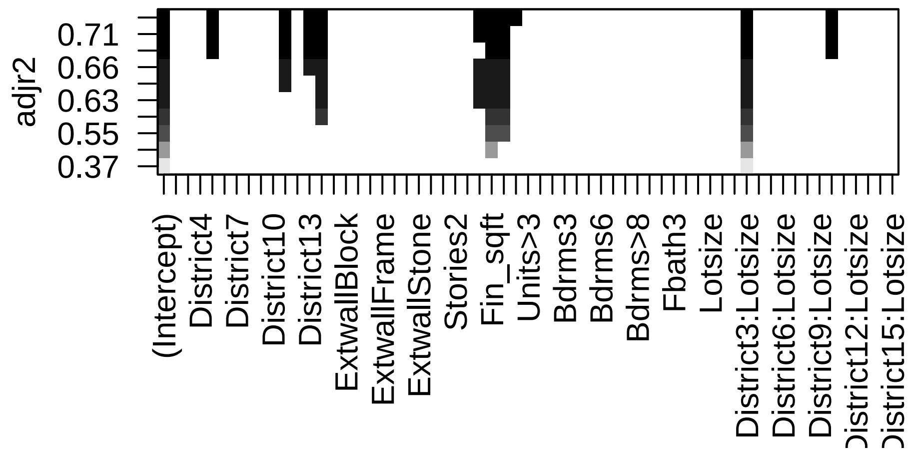
We can do the same thing on the real estate data:

```
# install.packages('leaps')
# If you run this, you will get the following error:
# all=leaps::regsubsets(Sale_price~ District +  Extwall +
#             Stories  +    Year_Built  + Fin_sqft  +
#             Units   +     Bdrms   +
#             Fbath  +    Lotsize   + Sale_date +District*Lotsize,data=df2,nvmax=10,method
# "Error in leaps.exhaustive(a, really.big) : Exhaustive search will be S L O W, must spec
# This shows how this can quickly become computationally intensive

# You will see that running the following code takes very long!
all=leaps::regsubsets(Sale_price~ District +  Extwall +
            Stories  +    Year_Built  + Fin_sqft  +
            Units   +     Bdrms   +
            Fbath  +    Lotsize   + Sale_date+District*Lotsize,data=df2,nvmax=10,really.bi
```

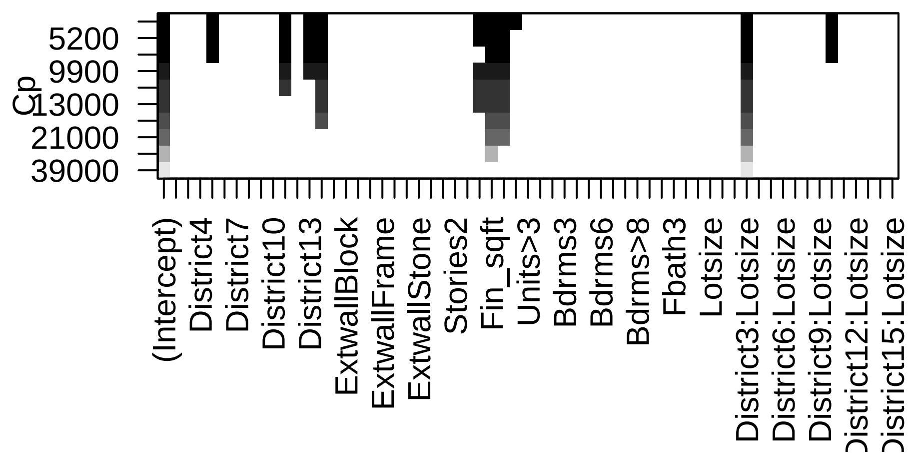
```r
plot(all,scale='r2')
```



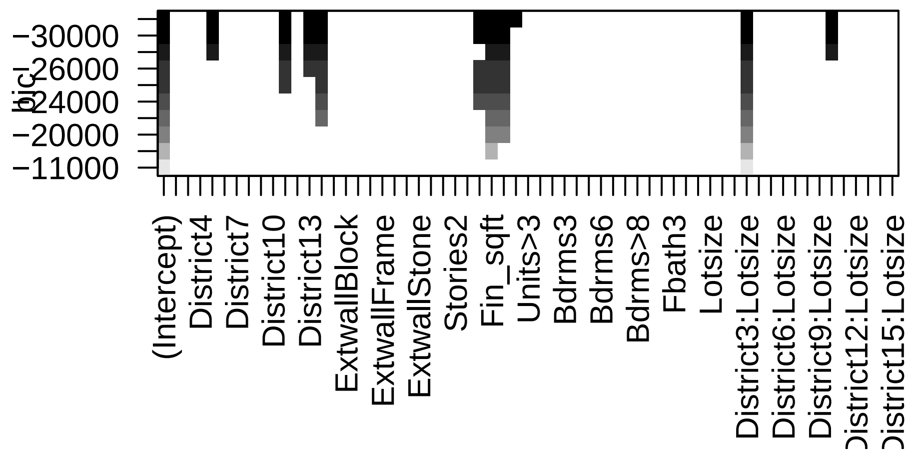

```r
plot(all,scale='adjr2')
```




```
plot(all,scale='Cp')
```



```
plot(all,scale='bic')
```



The selected models appear fairly sparse!

9.5.2 Forward selection

Forward selection is another common algorithm used to select the best model. It involves adding regressors one by one until adding more regressors becomes unhelpful. Essentially, we find the most "significant" regressor to be added to the model, and add it if it is "significant enough". We keep adding regressors, until none are significant enough. However, forward selection is "greedy", in the sense that it does not consider all possible subsets of regressors. This means that it is possible that the "best" model is missed by the algorithm.

The algorithm proceeds as follows:

1. Choose α_{entry}, which is the significance level for a regressor to enter the model. Set the current model to be $Y = \beta_0 + \epsilon$.
2. Among all regressors X_i not in the current model, test $H_0 : X_i$ not entered versus $H_a : X_i$ entered.
3. Choose the covariate with smallest p-value, i.e. the most likely X_i to be entered. Say the p-value for testing $H_0 : \beta_1 = 0$ is the smallest.

4. If the chosen p-value is $\geq \alpha_{entry}$, then X_1 is not entered, and the current model is chosen as the final model and the process terminates. Otherwise, X_1 is entered and the new current model is set to the current model, with the addition of X_1.
5. Return to step 2.

Note that once an explanatory variable is entered, it will never leave the model.

Example 9.7. Suppose that we have 3 explanatory variables and α_{entry} is chosen to be 0.05. The following is how we would proceed.

Iteration 1:

Model	$H_a:$	p-value
$Y = \beta_0 + \beta_1 X_1 + \epsilon$	$\beta_1 \neq 0$	0.00033
$Y = \beta_0 + \beta_2 X_2 + \epsilon$	$\beta_2 \neq 0$	0.00490
$Y = \beta_0 + \beta_3 X_3 + \epsilon$	$\beta_3 \neq 0$	0.00101

Since 0.00033 is the smallest p-value and it is less than α_{entry}, the chosen model is $Y = \beta_0 + \beta_1 X_1 + \epsilon$, and we continue to the next iteration.

Iteration 2:

Model	$H_a:$	p-value
$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$	$\beta_2 \neq 0$	0.01330
$Y = \beta_0 + \beta_1 X_1 + \beta_3 X_3 + \epsilon$	$\beta_3 \neq 0$	0.1125

Since 0.01330 is the smallest p-value and it is less than α_{entry}, the chosen model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$, and we continue to the next iteration.

Iteration 3:

Model	$H_a:$	p-value
$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$	$\beta_3 \neq 0$	0.36145

Since 0.36145 is the smallest p-value and it is greater than α_{entry}, the chosen model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$, and STOP.

If α_{entry} is chosen to be 0.01, observe that we will stop at Step 2 and the chosen model is $Y = \beta_0 + \beta_1 X_1 + \epsilon$. As you can see, the chosen model is easily affected by the chosen α_{entry} value.

9.5.3 Backward elimination

In Backward elimination, we start with the largest model, i.e., the one with all of the regressors, and eliminate regressors one by one until nothing can be eliminated. Like forward selection, backward selection is also greedy, and therefore, suffers from the same drawbacks.

For $p-1$ regressors, the algorithm proceeds as follows:

1. Choose: α_{stay}, which is the significance level for an explanatory variable to **stay** in the model. Set the current model to be $Y = \beta_0 + \beta_1 X_1 + ... + \beta_{p-1} X_{p-1} + \epsilon$.
2. Test $H_0 : X_i$ eliminated versus $H_a : X_i$ not eliminated, i.e. $H_0 : \beta_i = 0$ versus $H_a : \beta_i \neq 0$ with respect to the current model.
3. Choose the regressor with the largest p-value – i.e., the most likely X_i to be eliminated. Say p-value for testing $H_0 : \beta_p = 0$ is the largest.
4. If the chosen p-value is $< \alpha_{stay}$, then X_p is not eliminated, and the chosen model is the current model and the process terminates. Otherwise, X_p is eliminated and the current model is set to be the old current model, with X_p removed.
5. If there are no regressors in the current model, terminate. Otherwise, go back to step 2.

Example 9.8. Suppose that we have 3 explanatory variables and α_{stay} is chosen to be 0.05. The following is how we would proceed.

The current model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$.

Iteration 1:

H_a :	p-value
$\beta_1 \neq 0$	0.0176
$\beta_2 \neq 0$	0.05627
$\beta_3 \neq 0$	0.36145

Since 0.36145 is the largest p-value and it is greater than α_{stay}, the current model is set to be $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$, and we continue to the next iteration.

Iteration 2:

Model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$.

H_a :	p-value
$\beta_1 \neq 0$	0.00139
$\beta_2 \neq 0$	0.01330

Since 0.01330 is the smallest p-value and it is less than α_{stay}, the final model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$, and the process terminates.

If α_{stay} is chosen to be 0.01, then we will continue the process and eventually, the chosen model is $Y = \beta_0 + \beta_1 X_1 + \epsilon$. Again, the chosen model is easily affected by the chosen α_{stay} value.

In forward selection, once a regressor is entered, it will be in the final model. In backward elimination, once a regressor is eliminated, it will not be in the final model. A combination of these two processes, that allows for variables to enter and exit the model is called **stepwise Regression**.

9.5.4 Stepwise regression

In stepwise regression, we start the model from smallest model, $Y = \beta_0 + \epsilon$, and we keep adding and eliminating regressors one by one such that added regressors can be eliminated later and eliminated regressors can be added later. Like forward selection and backward selection, stepwise regression is also greedy, and therefore, suffers from the same drawbacks.

For $p-1$ regressors, the algorithm proceeds as follows:

1. Choose: α_{enter} and α_{stay}. Set the current model to be $Y = \beta_0 + \epsilon$.
2. Set the old model to be the current model. Perform steps 2-4 of forward selection.
3. Suppose in the previous step, X_1 was added to the current model. Next, with the current model, apply steps 2-5 of backward elimination.
4. If X_1 is eliminated in backward elimination, then do not remove X_1 from the current model and set the current model to be the final model. The process then terminates. If the current model equals the old model, terminate. Otherwise, go back to step 2.

The stopping rule for stepwise regression is when we run into infinite loop (the same variable being added and eliminated) OR when we have nothing to add **and** nothing to eliminate.

> **Note**
>
> 1. One way to avoid the infinite loop is to choose $\alpha_{stay} \neq \alpha_{entry}$.
> 2. Some books suggest "easy to enter" and "tough to eliminate" strategy for selecting $\alpha_{stay}, \alpha_{entry}$ but there is no strict rule.
> 3. `R` is using the AIC criterion for entering and eliminating, rather than the F test. This is also acceptable.

Example 9.9. Recall the real estate example - Example 6.6. Run forward selection, backward selection and stepwise regression using AIC as the criterion, including all regressors. (AIC is done automatically in R.)

```r
# ########### Automated methods ###########

############ Forward selection with AIC ###########

#define intercept-only model
intercept_only = lm(Sale_price~ 1, data=df2)

# define model with all predictors
all = lm(Sale_price~ ., data=df2)

# perform forward stepwise regression
forward = step(intercept_only, direction='forward', scope=formula(all), trace=0)

#view results of forward stepwise regression
forward$anova
```

```
           Step  Df     Deviance Resid. Df   Resid. Dev      AIC
1                NA           NA     24441 1.710695e+14 554078.5
2    + District -14 6.175224e+13     24427 1.093172e+14 543161.0
3    + Fin_sqft  -1 3.824600e+13     24426 7.107123e+13 532639.0
4       + Units  -3 1.789793e+13     24423 5.317331e+13 525553.7
5       + Fbath  -5 2.331394e+12     24418 5.084191e+13 524467.8
6       + Bdrms  -9 1.243638e+12     24409 4.959827e+13 523880.5
7   + Sale_date  -1 7.291417e+11     24408 4.886913e+13 523520.5
8  + Year_Built  -1 7.023471e+11     24407 4.816678e+13 523168.7
9     + Lotsize  -1 3.364533e+11     24406 4.783033e+13 522999.4
10    + Extwall  -8 2.479395e+11     24398 4.758239e+13 522888.3
11    + Stories  -3 2.068265e+11     24395 4.737557e+13 522787.9
```

```r
#view final model
names(forward$coefficients)
```

```
 [1] "(Intercept)"            "District2"              "District3"
 [4] "District4"              "District5"              "District6"
 [7] "District7"              "District8"              "District9"
[10] "District10"             "District11"             "District12"
[13] "District13"             "District14"             "District15"
[16] "Fin_sqft"               "Units2"                 "Units3"
[19] "Units>3"                "Fbath1"                 "Fbath2"
[22] "Fbath3"                 "Fbath4"                 "Fbath>4"
[25] "Bdrms1"                 "Bdrms2"                 "Bdrms3"
```

```
[28] "Bdrms4"                  "Bdrms5"                  "Bdrms6"
[31] "Bdrms7"                  "Bdrms8"                  "Bdrms>8"
[34] "Sale_date"               "Year_Built"              "Lotsize"
[37] "ExtwallBlock"            "ExtwallBrick"            "ExtwallFiber-Cement"
[40] "ExtwallFrame"            "ExtwallMasonry / Frame" "ExtwallPrem Wood"
[43] "ExtwallStone"            "ExtwallStucco"           "Stories1.5"
[46] "Stories2"                "Stories>2"
```

```r
# ###################### Backward selection with AIC ######################
#define intercept-only model
intercept_only = lm(Sale_price~ 1, data=df_clean4)

#define model with all predictors
all = lm(Sale_price~ ., data=df_clean4)

#perform backward stepwise regression
backward = step(all, direction='backward', scope=formula(all), trace=0)

#view results of backward stepwise regression
backward$anova
```

```
   Step Df Deviance Resid. Df   Resid. Dev      AIC
1        NA       NA     24396 4.757525e+13 522911.1
2 - d_3  0        0     24396 4.757525e+13 522911.1
```

```r
#view final model

names(backward$coefficients)
```

```
 [1] "(Intercept)"             "District2"               "District3"
 [4] "District4"               "District5"               "District6"
 [7] "District7"               "District8"               "District9"
[10] "District10"              "District11"              "District12"
[13] "District13"              "District14"              "District15"
[16] "ExtwallBlock"            "ExtwallBrick"            "ExtwallFiber-Cement"
[19] "ExtwallFrame"            "ExtwallMasonry / Frame" "ExtwallPrem Wood"
[22] "ExtwallStone"            "ExtwallStucco"           "Stories1.5"
[25] "Stories2"                "Stories>2"               "Year_Built"
[28] "Fin_sqft"                "Units2"                  "Units3"
[31] "Units>3"                 "Bdrms1"                  "Bdrms2"
```

```
[34] "Bdrms3"                "Bdrms4"                "Bdrms5"
[37] "Bdrms6"                "Bdrms7"                "Bdrms8"
[40] "Bdrms>8"               "Fbath1"                "Fbath2"
[43] "Fbath3"                "Fbath4"                "Fbath>4"
[46] "Lotsize"               "Sale_date"
```

```r
########### Stepwise Regression with AIC ###########
# Define intercept-only model
intercept_only = lm(Sale_price~ 1, data=df_clean4)

# Define model with all predictors
all = lm(Sale_price~ ., data=df_clean4)

#perform stepwise regression
both = step(intercept_only, direction='both', scope=formula(all), trace=0)

#view results of Stepwise regression
both$anova
```

```
           Step  Df     Deviance Resid. Df   Resid. Dev      AIC
1                NA           NA     24442 1.723269e+14 554279.2
2    + District -14 6.207601e+13     24428 1.102509e+14 543390.1
3    + Fin_sqft  -1 3.865374e+13     24427 7.159711e+13 532839.9
4       + Units  -3 1.806837e+13     24424 5.352874e+13 525737.0
5       + Fbath  -5 2.390742e+12     24419 5.113800e+13 524630.2
6       + Bdrms  -9 1.268332e+12     24410 4.986967e+13 524034.3
7   + Sale_date  -1 7.299962e+11     24409 4.913967e+13 523675.9
8  + Year_Built  -1 7.284709e+11     24408 4.841120e+13 523312.8
9     + Lotsize  -1 3.850140e+11     24407 4.802619e+13 523119.7
10    + Extwall  -8 2.445413e+11     24399 4.778164e+13 523010.9
11    + Stories  -3 2.063959e+11     24396 4.757525e+13 522911.1
```

```r
#view final model
names(both$coefficients)
```

```
 [1] "(Intercept)"           "District2"             "District3"
 [4] "District4"             "District5"             "District6"
 [7] "District7"             "District8"             "District9"
[10] "District10"            "District11"            "District12"
[13] "District13"            "District14"            "District15"
```

```
[16] "Fin_sqft"                "Units2"                  "Units3"
[19] "Units>3"                 "Fbath1"                  "Fbath2"
[22] "Fbath3"                  "Fbath4"                  "Fbath>4"
[25] "Bdrms1"                  "Bdrms2"                  "Bdrms3"
[28] "Bdrms4"                  "Bdrms5"                  "Bdrms6"
[31] "Bdrms7"                  "Bdrms8"                  "Bdrms>8"
[34] "Sale_date"               "Year_Built"              "Lotsize"
[37] "ExtwallBlock"            "ExtwallBrick"            "ExtwallFiber-Cement"
[40] "ExtwallFrame"            "ExtwallMasonry / Frame" "ExtwallPrem Wood"
[43] "ExtwallStone"            "ExtwallStucco"           "Stories1.5"
[46] "Stories2"                "Stories>2"
```

These methods retain many more variables. It is interesting, considering all subsets contained very few.

Example 9.10. Use the NFL data from the textbook - Run forward selection and backward selection including all regressors using the F statistic criterion.

```
############ Forward selection by F value
# Threshold for p-value to determine variable inclusion
thresh = 0.05

# Initialize an empty vector to store currently selected variables
curr_vars = c()

# Create a vector of variable names excluding the dependent variable 'Wins' and 'PerR'
vars_left = names(df)[2:10]
vars_left = vars_left[vars_left != 'PerR']

# Initialize the passed flag to TRUE
passed = TRUE

# Loop until no more variables can be added (passed is FALSE) or there are no more variabl
while(passed && length(vars_left) > 0) {

  # Initialize an empty vector to store p-values of models
  pvals = c()

  # Loop through each remaining variable
  for(var in vars_left) {
    # Create a temporary dataframe with current variables and the new candidate variable
    df_tmp = df[, c(curr_vars, var, 'Wins')]
```

```r
    # Fit a linear model with 'Wins' as the dependent variable
    model = lm(Wins ~ ., df_tmp)

    # Get the summary of the model
    s = summary(model)

    # Calculate the p-value of the F-statistic for the model
    pval = 1 - pf(s$fstatistic[1], s$fstatistic[2], s$fstatistic[3])

    # Append the p-value to the pvals vector
    pvals = c(pvals, pval)
  }

  # Find the index of the minimum p-value
  min_index = which.min(pvals)

  # Get the minimum p-value
  mp = pvals[min_index]

  # Check if the minimum p-value is less than the threshold
  passed = mp < thresh

  if(passed) {
    # If passed, get the corresponding variable name
    new_var = vars_left[min_index]

    # Add the new variable to the current variables list
    curr_vars = c(curr_vars, new_var)

    # Remove the new variable from the remaining variables list
    vars_left = vars_left[vars_left != new_var]

    # Print the p-value and the variable being added
    print('pvalue')
    print(mp)
    print('adding')
    print(new_var)
  }
}
```

```
[1] "pvalue"
```

```
       value
7.380709e-06
[1] "adding"
[1] "ORY"
[1] "pvalue"
       value
4.151848e-08
[1] "adding"
[1] "PassY"
[1] "pvalue"
       value
5.286139e-08
[1] "adding"
[1] "RushY"
[1] "pvalue"
       value
1.237473e-07
[1] "adding"
[1] "OPY"
[1] "pvalue"
       value
5.151486e-07
[1] "adding"
[1] "PenY"
[1] "pvalue"
       value
2.098536e-06
[1] "adding"
[1] "TurnD"
[1] "pvalue"
       value
7.894081e-06
[1] "adding"
[1] "PuntA"
[1] "pvalue"
      value
2.52811e-05
[1] "adding"
[1] "FGP"
```

```
# Print the final list of selected variables
print(curr_vars)
```

```
[1] "ORY"   "PassY" "RushY" "OPY"   "PenY"  "TurnD" "PuntA" "FGP"
```

```
# Threshold for p-value to determine variable exclusion
thresh = 0.05

# Initialize a vector of variable names excluding the dependent variable 'Wins' and 'PerR'
vars_left = names(df)[2:10]
vars_left = vars_left[vars_left != 'PerR']

# Initialize the passed flag to TRUE
passed = TRUE

# Loop until no more variables need to be removed (passed is FALSE) or there are no more v
while(passed && length(vars_left) > 0) {
  # Create a temporary dataframe with remaining variables and the dependent variable 'Wins
  df_tmp = df[, c(vars_left, 'Wins')]

  # Fit a linear model with 'Wins' as the dependent variable
  model = lm(Wins ~ ., df_tmp)

  # Get the summary of the model
  s = summary(model)

  # Extract the p-values of the t-statistics for the coefficients
  tv = coef(s)[, "Pr(>|t|)"]

  # Remove the intercept p-value
  tv = tv[-1]

  # Find the index of the maximum p-value
  max_index = which.max(tv)

  # Get the maximum p-value
  mp = tv[max_index]

  # Print the maximum p-value
  print(mp)

  # Check if the maximum p-value is greater than the threshold
  passed = mp > thresh

  if(passed) {
```

```r
      # If passed, get the corresponding variable name to be removed
      rem_var = names(max_index)

      # Remove the variable from the remaining variables list
      vars_left = vars_left[vars_left != rem_var]

      # Print the p-value and the variable being removed
      print('pvalue')
      print(mp)
      print('removing')
      print(rem_var)
    }
  }
```

```
    TurnD
0.7312364
[1] "pvalue"
    TurnD
0.7312364
[1] "removing"
[1] "TurnD"
      FGP
0.5679556
[1] "pvalue"
      FGP
0.5679556
[1] "removing"
[1] "FGP"
    PuntA
0.6919653
[1] "pvalue"
    PuntA
0.6919653
[1] "removing"
[1] "PuntA"
     PenY
0.5147875
[1] "pvalue"
     PenY
0.5147875
[1] "removing"
[1] "PenY"
```

```
      OPY
0.1752231
[1] "pvalue"
      OPY
0.1752231
[1] "removing"
[1] "OPY"
     RushY
0.06663316
[1] "pvalue"
     RushY
0.06663316
[1] "removing"
[1] "RushY"
       PassY
0.0001775241
```

```
# Print the final list of remaining variables
print(vars_left)
```

```
[1] "PassY" "ORY"
```

Exercise 9.1. Use the NFL data from the textbook - modify the above code to perform stepwise regression, including all regressors using the F statistic criterion.

9.6 Cross Validation

We may be mainly interested in predictive performance of our model. To assess the performance of our model's predictive ability, it would be ideal to test its ability to predict the response. We could test our model out on our dataset, but our model has already 'seen' this dataset. Previously, we saw that the model's predictive ability on the dataset at hand does not neccessarily mean it will output good predictions for new data from the same population. This means that the predictive ability of the model on the current observations won't reflect the model's ability to predict observations it hasn't seen. It would be ideal to have some new data in order to measure predictive performance, but it is also ideal to use all data in building the model. **Cross validation** allows us to obtain an estimate of a given model's out of sample predictive performance.

Cross-validation is a technique where we repeatedly fit our model using a subset of the dataset and then compute its predictive performance on the complementary subset of the dataset.

We then average these predictive performances and use this to estimate our model's overall predictive performance.

Here is an overview of the algorithm:

1. Partition the data $\mathbb{Z}$ into k groups of size about n/k uniformly at random. Denoted $\mathbb{Z}_1, \ldots, \mathbb{Z}_k$.
2. For each; $\mathbb{Z}_j$, fit the regression model on $\mathbb{Z} \backslash \mathbb{Z}_j$.
3. For each observation in the left out group $\mathbb{Z}_j$, use the previously computed model to produce a prediction $\hat{Y}_{j,1}, \ldots, \hat{Y}_{j,n/k}$, resulting in n/k predictions.
4. For each prediction, compute the predictive error $(y_{j,i} - \hat{y}_{j,i})^2$ (Can also use absolute error).
5. Repeat steps 2-3 for each group $1, \ldots, k$.
6. Average the n predictive errors.
7. Repeat for all candidate models and choose the model with the lowest prediction error.

To highlight the choice of k, we say k-fold Cross Validation – Each of the k subsets are known as folds. What do we choose for k? LOOCV (Leave One Out Cross Validation) is a popular technique. Here, $k = n$. This is where each run, one observation is left out of the model. This gives a low bias, but high variation estimate of the prediction error. It also has a high run time. As a result, often, k is taken to be 5 or 10. Taking lower values of k gives a result similar to that of splitting the data into training and test sets, while higher values of k gives a result similar to LOOCV. The choice of the number of folds in cross validation is a bias-variance tradeoff: too few folds may result in high bias, while too many folds may result in high variance. (This is why k is taken between 1 and n.)

Example 9.11. Using the real estate data, run cross validation in a forward selection manner. That is, run the cross validation procedure for the model with the first variable, the first and the second variable, the first three variables etc. Which model has the lowest error?

The `caret` package can be used to perform 5-fold CV.

```r
########### Cross Validation Regression ###########
# install.packages('caret')
library(caret)
```

```
Warning: package 'caret' was built under R version 4.2.3
```

```
Loading required package: lattice
```

```r
#specify the cross-validation method 5 fold in this case.
ctrl = trainControl(method = "cv", number = 5)

#fit a regression model and use k-fold CV to evaluate performance
model = train(Sale_price~., data = df2, method = "lm", trControl = ctrl)
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```r
for(i in 1:length(var_list)){
  vars=c(var_list[1:i],"Sale_price")
  model = train(Sale_price~., data = df2[,vars], method = "lm", trControl = ctrl)
  print(paste0("k" ,i," CV Result ", model$results$RMSE,sep=""))
}
```

```
[1] "k1 CV Result 66912.7355323838"
[1] "k2 CV Result 64288.4426359389"
[1] "k3 CV Result 61819.6176427859"
[1] "k4 CV Result 61336.9593108528"
[1] "k5 CV Result 52061.7475798042"
[1] "k6 CV Result 46100.1776666706"
[1] "k7 CV Result 45683.6103871796"
[1] "k8 CV Result 44988.2437790465"
[1] "k9 CV Result 44668.105850592"
[1] "k10 CV Result 44282.8184779538"
```

```r
#view summary of k-fold CV
print(model)
```

```
Linear Regression

24442 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 19554, 19554, 19555, 19552, 19553
Resampling results:

  RMSE      Rsquared   MAE
  44282.82  0.7199015  29860.22

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
# We see that all of the variables minimize the prediction error.
```

Example 9.12. Use the NFL data from the textbook. 1. Compare the model `Wins~.` to the model `Wins~PassY+PuntA+FGP+TurnD+PenY+OPY+RushDiff`. 2. Run cross validation in a forward selection manner. That is, run the cross validation procedure for the model with the first variable, the first and the second variable, the first three variables etc. Which model has the lowest error? 3. Run cross validation for all subsets. Which model has the lowest error? Does it match the result in step 2?

```r
########### Cross Validation Regression ###########
set.seed(1251)


#specify the cross-validation method
ctrl = trainControl(method = "cv", number = 5)

#fit a regression model and use k-fold CV to evaluate performance
model_cv_1 = train(Wins~., data = df, method = "lm", trControl = ctrl)
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
model_cv_1
```

```
Linear Regression

28 samples
10 predictors

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 21, 22, 22, 24, 23
Resampling results:

  RMSE      Rsquared   MAE
  1.962945  0.5925805  1.686354

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
model_cv_2 = train(Wins~PassY+PuntA+FGP+TurnD+PenY+OPY+RushDiff, data = df, method = "lm",
model_cv_2
```

```
Linear Regression

28 samples
 7 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
```

```
Summary of sample sizes: 23, 21, 23, 23, 22
Resampling results:

  RMSE      Rsquared   MAE
  2.418524  0.6516294  1.887159

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
for(i in 2:ncol(df)){
  # reg=sample(2:10,sample(1:9,1))
  vars=c(names(df)[2:i],'Wins')
  model = train(Wins~., data = df[,vars], method = "lm", trControl = ctrl)
  print(paste0("k=" ,i-1," vars ",paste(vars,collapse=' ')," CV Result ",
               round(model$results$RMSE,1),sep=""))
}
```

```
[1] "k=1 vars RushY Wins CV Result 2.9"
[1] "k=2 vars RushY PassY Wins CV Result 2.3"
[1] "k=3 vars RushY PassY PuntA Wins CV Result 2.2"
[1] "k=4 vars RushY PassY PuntA FGP Wins CV Result 2.4"
[1] "k=5 vars RushY PassY PuntA FGP TurnD Wins CV Result 2.6"
[1] "k=6 vars RushY PassY PuntA FGP TurnD PenY Wins CV Result 2.5"
[1] "k=7 vars RushY PassY PuntA FGP TurnD PenY PerR Wins CV Result 2.5"
[1] "k=8 vars RushY PassY PuntA FGP TurnD PenY PerR ORY Wins CV Result 2.5"
[1] "k=9 vars RushY PassY PuntA FGP TurnD PenY PerR ORY OPY Wins CV Result 2"

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
[1] "k=10 vars RushY PassY PuntA FGP TurnD PenY PerR ORY OPY RushDiff Wins CV Result 2.5"
```

```r
# Select model k=9 vars

#
p = ncol(df)-1
l = rep(list(0:1), p)

models=expand.grid(l); dim(models)
```

```
[1] 1024   10
```

```r
cv=RMSE=c()
for(i in 2:nrow(models)){
  reg=(2:(p+1))[models[i,]==1]
  vars=c(names(df)[reg],'Wins')
  model = train(Wins~., data = df[,vars], method = "lm", trControl = ctrl)
  RMSE=c(RMSE,model$results$RMSE)
  cv=c(cv,list(model))
  # print(paste0("k=" ,k," vars ",paste(vars,collapse=' ')," CV Result ",
               # round(model$results$RMSE,1),sep=""))
}
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

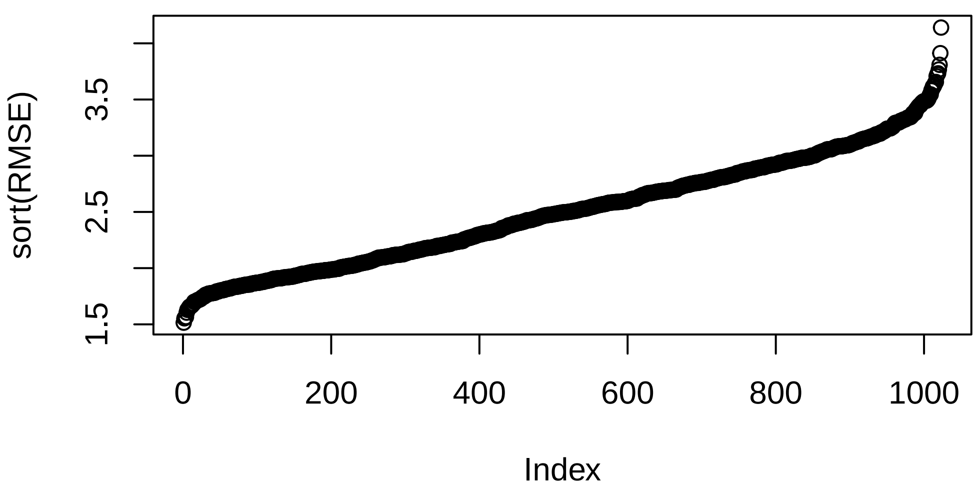
```
Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
```

```
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading

Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
may be misleading
```

```
bottom_10=order(RMSE)[1:10]
RMSE[bottom_10]
```

```
 [1] 1.514831 1.553680 1.561540 1.565194 1.604034 1.629024 1.634625 1.647743
 [9] 1.657872 1.658021
```

```
plot(sort(RMSE))
```



```r
m=which.min(RMSE)
cv[which.min(RMSE)]
```

```
[[1]]
Linear Regression

28 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 22, 22, 23, 23, 22
Resampling results:

  RMSE      Rsquared   MAE
  1.514831  0.7945392  1.255977

Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
df[1,(2:(p+1))[unlist(models[m,])==1]]
```

```
  RushY PuntA PerR  OPY RushDiff
1  2113  38.9 59.7 1917      -92
```

```r
model = train(Wins~ PassY+PuntA+TurnD+PerR++OPY+RushDiff, data = df, method = "lm", trCont
RMSE=c(RMSE,model$results$RMSE)
```

9.7 Homework questions

Complete the Chapter 10 questions from the textbook.

Exercise 9.2. Load car mileage data from Example 3.8 and run 10-fold cross validation on all subset models. Which model is the best?

10 Robust Regression

A serious problem that may dramatically impact the usefulness of a regression model is that of outlying observations. We have seen trying to locate and remove outliers, but an alternative technique is to use robust statistics. Robust statistics is a field of statistics that develops estimators that are not corrupted by outlying observations.

Recall that we minimize the squared error to obtain the least squares estimator. The best estimator was defined as the minimizer of

$$C(\beta) = \sum_{i=1}^{n} (Y_i - \beta^\top X_i)^2.$$

The squared error is an example of a **cost function**. The fact that the errors are squared implies that unusually large residuals will contribute significantly to the cost function. We may instead define the best estimator to minimize another cost function, say

$$C(\beta) = \sum_{i=1}^{n} |Y_i - \beta^\top X_i|.$$

The absolute value function grows linearly, thus, it is less affected by unusually large residuals. We can go a step further and design a cost function that is even less affected by unusually large residuals.

10.1 M-estimators

M-estimators are a group of estimators which can be defined as minimizers of cost functions. (The M stands for minimizers.) We can define an error function ρ as a function such that $\rho(|x|)$ is non-negative, non-decreasing and $\rho(0) = 0$. We can then define the **M-estimator** with error function ρ as

$$\hat{\beta} = \operatorname*{argmin}_{\beta} C_\rho(\beta) = \operatorname*{argmin}_{\beta} \sum_{i=1}^{n} \rho(Y_i - \beta^\top X_i) = \operatorname*{argmin}_{\beta} \sum_{i=1}^{n} \rho(\hat{\epsilon}_i(\beta)),$$

where $\hat{\epsilon}_i(\beta) = Y_i - \beta^\top X_i$. Note that for arbitrary ρ, the above may not have a unique minimizer, which may be problematic for computational algorithms and theoretical analysis. An M-estimator is an alternative to the OLS estimator. The idea is to define ρ so that large residuals $\hat{\epsilon}_i(\beta)$ do not contribute more than they should to the cost function. To elaborate, we would like $\rho(\hat{\epsilon}_i(\beta))$ to be somewhat large if $\hat{\epsilon}_i(\beta)$ is large, because that signals that the model defined by β does not fit the data well at that point. On the other hand, if observation i is an outlier, than we don't want to try too hard to make the model fit that point.

One drawback of M-estimators is that they are not necessarily scale invariant. For example, we have that

$$Y = X\beta + \epsilon \implies aY = aX\beta + a\epsilon.$$

Therefore, it is natural that we would expect the estimation procedure applied to (X, Y) to produce the same estimator for β as the estimation procedure applied to (aX, aY). Unfortunately, this is not always the case for M-estimators. One way to remedy this is to obtain a (robust) scale estimate s, and then define the estimate as

$$\operatorname*{argmin}_{\beta} \sum_{i=1}^{n} \rho(\frac{Y_i - \beta^\top X_i}{s}).$$

For example, the median absolute deviation divided by 0.6745 is often used.

10.2 Different ρ functions

Let ψ be the derivative of ρ. Some common choices for ρ include the following:

10.2.1 Least Squares

Least squares loss, also known as quadratic loss, is the most common loss function in regression analysis. It is defined as the square of the residuals (the differences between observed and predicted values).

$$\rho(u) = u^2, \qquad\qquad \psi(u) = 2u.$$

It is highly sensitive to outliers because the squared term amplifies large residuals, making it less robust in the presence of outliers.

10.2.2 Huber's Loss

Huber's loss is a piecewise loss function that is quadratic for small residuals and linear for large residuals. It is less sensitive to outliers compared to least squares loss. This loss provides a compromise between least squares and absolute loss, offering robustness to outliers while retaining efficiency for small residuals.

$$\rho(u) = \begin{cases} u^2/2 & \text{if } |u| \le \delta \\ \delta(|u| - \delta/2) & \text{if } |u| > \delta \end{cases}, \qquad \psi(u) = \begin{cases} u & \text{if } |u| \le \delta \\ \delta \times sign(u) & \text{if } |u| > \delta \end{cases}.$$

The next three losses offer even more robustness to outliers, potentially at the cost of efficiency.

10.2.3 Ramsay's E Function

Ramsay's E function is used to give higher influence to residuals near zero and less influence to large residuals.

$$\rho(u) = \frac{1 - (1 + \delta|u|)\exp(-\delta|u|)}{\delta^2}, \qquad \psi(u) = sign(u)\exp(-\delta|u|)/\delta.$$

10.2.4 Andrews' Wave Function

Andrews' wave function is a loss function that limits the influence of large residuals more than Huber's loss.

$$\rho(u) = \delta\left(1 - \cos\left(\frac{u}{\delta}\right)\right), \qquad \psi(u) = \frac{\sin(u/\delta)}{u/\delta}.$$

10.2.5 Tukey's Loss

Tukey's loss function, also known as the biweight loss function, completely cuts off the influence of residuals beyond a certain point.

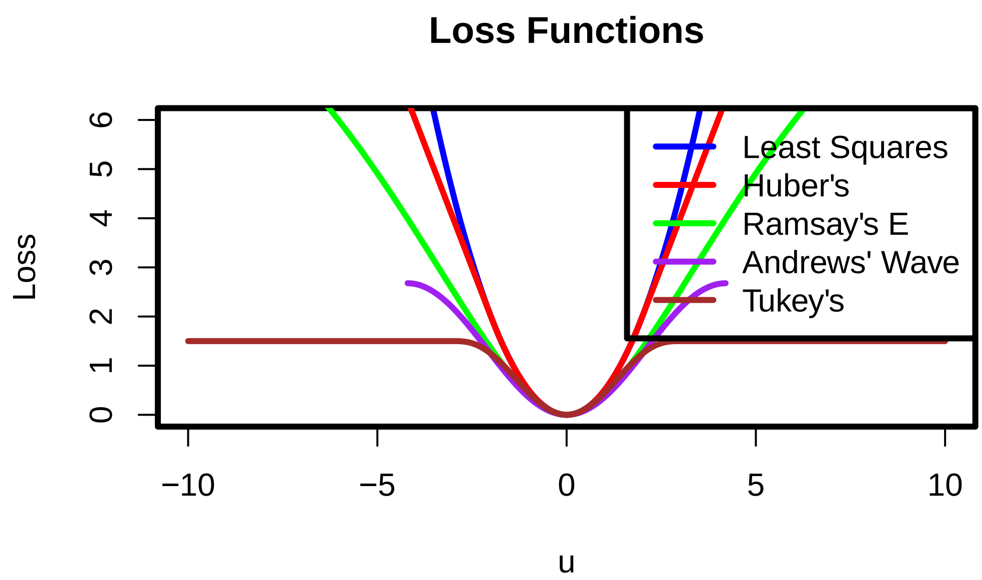
$$\rho(u) = \begin{cases} \frac{c^2}{6}\left[1 - \left(1 - \left(\frac{u}{c}\right)^2\right)^3\right] & \text{if } |u| \le c \\ \frac{c^2}{6} & \text{if } |u| > c \end{cases}, \qquad \psi(u) = \begin{cases} u\left(1 - \left(\frac{u}{c}\right)^2\right)^2 & \text{if } |u| \le c \\ 0 & \text{if } |u| > c \end{cases}.$$

Let's graph each of these functions:

```r
# Define the least squares loss function
least_squares_loss <- function(u) {
  return(u^2/2)
}
```

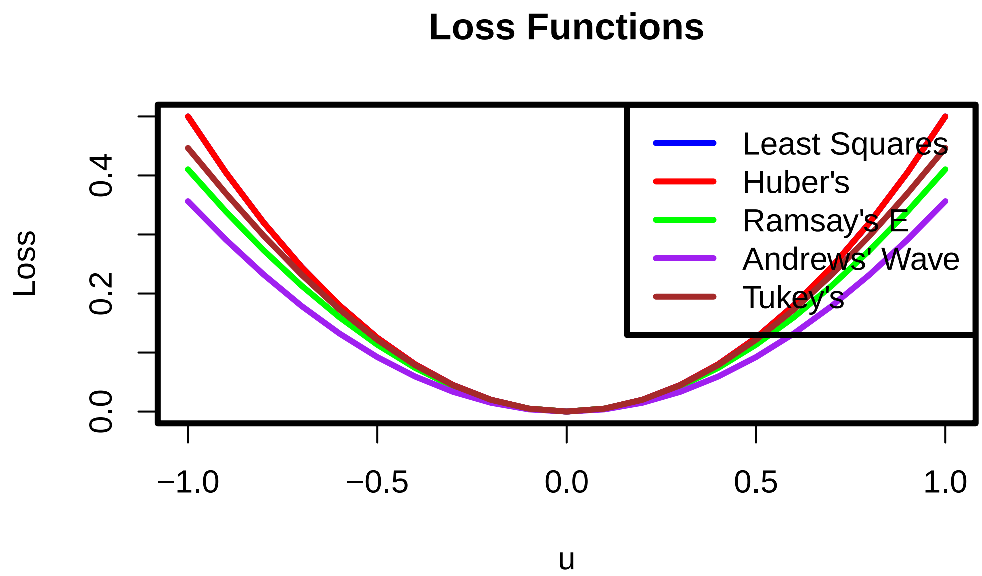
```r
# Define Huber's loss function
hubers_loss <- function(u, delta=2) {
  return(ifelse(abs(u) <= delta, 0.5 * u^2, delta * (abs(u) - 0.5 * delta)))
}

# Define Ramsay's E function
ramsays_e_function <- function(u, delta=0.3) {
  return((1-(1+delta*abs(u))*exp(-delta*abs(u)))/(delta^2))
}

# Define Andrews' wave function
andrews_wave_function <- function(u, delta=1.339) {
  if(abs(u / delta)<pi)
    return(delta * (1 - cos(u / delta)))
  else
    return(NA)
}


# Define Tukey's loss function
tukeys_loss <- function(u, c=3) {
  abs_u <- abs(u)
  return(ifelse(abs_u <= c, (c^2 / 6) * (1 - (1 - (u / c)^2)^3), c^2 / 6))
}


u <- seq(-10,10, by = 0.1)

ls_loss <- sapply(u,least_squares_loss)
h_loss <- sapply(u,hubers_loss)
r_loss <- sapply(u,ramsays_e_function)
a_loss <- sapply(u,andrews_wave_function)
tukey_loss <- sapply(u,tukeys_loss)

par(lwd=3)
# Plotting example losses for visualization
plot(u, ls_loss, type = "l", col = "blue", ylim = c(0, 6), ylab = "Loss", xlab = "u", main
lines(u, h_loss, col = "red")
lines(u, r_loss, col = "green")
lines(u, a_loss, col = "purple")
lines(u, tukey_loss, col = "brown")
```

```
legend("topright", legend = c("Least Squares", "Huber's", "Ramsay's E", "Andrews' Wave",
       col = c("blue", "red", "green", "purple", "brown"), lty = 1)
```




```
#Zooming in

u <- seq(-1, 1, by = 0.1)

ls_loss <- sapply(u,least_squares_loss)
h_loss <- sapply(u,hubers_loss)
r_loss <- sapply(u,ramsays_e_function)
a_loss <- sapply(u,andrews_wave_function)
tukey_loss <- sapply(u,tukeys_loss)

par(lwd=3)
# Plotting example losses for visualization
plot(u, ls_loss, type = "l", col = "blue", ylab = "Loss", xlab = "u", main = "Loss Functio
lines(u, h_loss, col = "red")
lines(u, r_loss, col = "green")
lines(u, a_loss, col = "purple")
lines(u, tukey_loss, col = "brown")
legend("topright", legend = c("Least Squares", "Huber's", "Ramsay's E", "Andrews' Wave",
```

```r
       col = c("blue", "red", "green", "purple", "brown"), lty = 1)
```



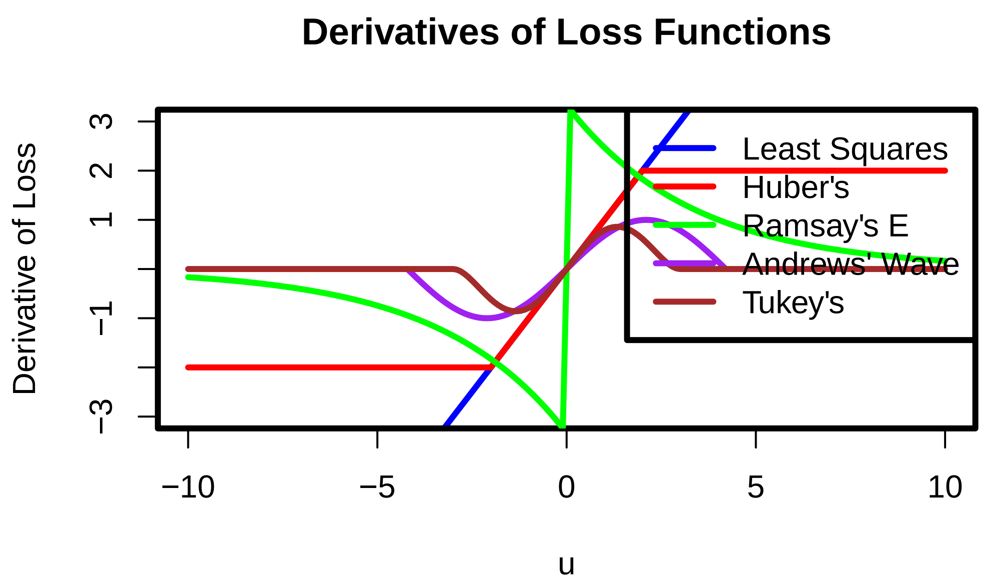
We see that the cost attributed to large residuals varies based on the chosen function. Most of the functions appear quadratic near 0, but have quite different behaviours at large values. Tukey and Andrew's Wave function completely cut off the influence of large residuals, which the remaining functions just dampen the influence (except of course, the lease squares loss). Huber's function and the least squares function are convex functions, which make computation of the minimzer considerably easier. Huber's function is about as robust as we can be, while still maintaining a convex loss function. Data drawn from distributions with heavy tails (data with many "large" observations) require more robust loss functions, such as the Tukey biweight function.

The robustness of a regression procedure can be classified by the behavior of ψ, the derivative of ρ. The ψ function controls the weight given to each residual and is proportional to a central concept in robustness called the **influence function.** Unbounded influence functions are not desirable from a robustness perspective, as this means that one corrupted point is able to drag the estimated hyperplane arbitrarily far. The ψ function for least squares ρ is unbounded, and thus least squares tends to be "nonrobust" when used with data arising from a heavy - tailed distribution. On the other hand. the Huber loss function has a monotone ψ function and is not unbounded, making it more robust.

The other influence functions actually redescend as the residual becomes larger. Ramsay's E function is a soft redescender, that is, the ψ function is approaches zero as $|z| \to \infty$. Andrew's wave function and Tukey loss are hard redescenders. That is, the ψ function equals zero for sufficiently large $|z|$. Note that the ρ functions associated with the redescending ψ functions are nonconvex, and this in theory can cause convergence problems in the iterative estimation procedure. We graph the functions below with their common default tuning parameters:

```
# Define the derivative of the least squares loss function
d_least_squares_loss <- function(u) {
  return(u)
}

# Define the derivative of Huber's loss function
d_hubers_loss <- function(u, delta = 2) {
  return(ifelse(abs(u) <= delta, u, delta * sign(u)))
}

# Define the derivative of Ramsay's E function
d_ramsays_e_function <- function(u, delta = 0.3) {
  return(sign(u)*exp(-delta*abs(u))/delta)
}

# Define the derivative of Andrews' wave function
d_andrews_wave_function <- function(u, delta = 1.339) {
  if(abs(u / delta) < pi)
    return(sin(u / delta))
  else
    return(NA)
}

# Define the derivative of Tukey's loss function
d_tukeys_loss <- function(u, c = 3) {
  abs_u <- abs(u)
  return(ifelse(abs_u <= c, u * (1 - (u / c)^2)^2, 0))
}


u <- seq(-10, 10, by = 0.1)

d_ls_loss <- sapply(u, d_least_squares_loss)
d_h_loss <- sapply(u, d_hubers_loss)
d_r_loss <- sapply(u, d_ramsays_e_function)
d_a_loss <- sapply(u, d_andrews_wave_function)
d_tukey_loss <- sapply(u, d_tukeys_loss)
```

```r
par(lwd = 3)
# Plotting example derivatives for visualization
plot(u, d_ls_loss, type = "l", col = "blue", ylim = c(-3, 3), ylab = "Derivative of Loss",
lines(u, d_h_loss, col = "red")
lines(u, d_r_loss, col = "green")
lines(u, d_a_loss, col = "purple")
lines(u, d_tukey_loss, col = "brown")
legend("topright", legend = c("Least Squares", "Huber's", "Ramsay's E", "Andrews' Wave",
       col = c("blue", "red", "green", "purple", "brown"), lty = 1)
```



10.3 Computing M-estimators

10.3.1 Iterated re-weighted least squares

Let ψ be the derivative of ρ. A common way to compute the minimizer, of $C_\rho(\beta)$ is to use iterated re-weighted least squares. First, note that to find the minimizer, of $C_\rho(\beta)$, we solve the following system of equations:

$$\sum_{i=1}^{n} X_{ij}\psi\left(\frac{Y_i - X_i^\top \beta}{s}\right) = 0, \quad j = 0, 1, \dots, k.$$

To do this, rewrite

$$\sum_{i=1}^{n} X_{ij}\psi\left(\frac{Y_i - X_i^\top \beta}{s}\right)\frac{Y_i - X_i^\top \beta}{s}\Big/\frac{Y_i - X_i^\top \beta}{s} = 0$$

$$\text{or,}\ \sum_{i=1}^{n} X_{ij}w_{i,\beta}\frac{Y_i - X_i^\top \beta}{s} = 0,$$

where

$$w_{i,\beta} = \psi\left(\frac{Y_i - X_i^\top \beta}{s}\right)\Big/\frac{Y_i - X_i^\top \beta}{s}.$$

Next, one will propose an initial estimate of the parameters α_0 and consider

$$\sum_{i=1}^{n} X_{ij}w_{i,\alpha_0}(Y_i - X_i^\top \beta) = 0.$$

Equivalently, we have $X^\top W_{\alpha_0} X\beta = X^\top W_{\alpha_0} Y$, where W_{α_0} is an $n \times n$ diagonal matrix of "weights" with diagonal elements $w_{1,\alpha_0}, \dots, w_{n,\alpha_0}$. The algorithm proceeds as follows: iteratively compute $\alpha_i = (X^\top W_{\alpha_{i-1}} X)^{-1} X^\top W_{\alpha_{i-1}} Y$ until $\|\alpha_i - \alpha_{i-1}\| < \epsilon$ for small ϵ.

10.3.2 Gradient descent

We can also use gradient descent to compute the minimizer, of $C_\rho(\beta)$. To do this given a step size $\eta > 0$, iteratively compute

$$\alpha_i = \alpha_{i-1} - \eta \times \sum_{i=1}^{n} X_i \psi\left(\frac{Y_i - X_i^\top \alpha_{i-1}}{s}\right),$$

until $\|\alpha_i - \alpha_{i-1}\| < \epsilon$ for small ϵ.

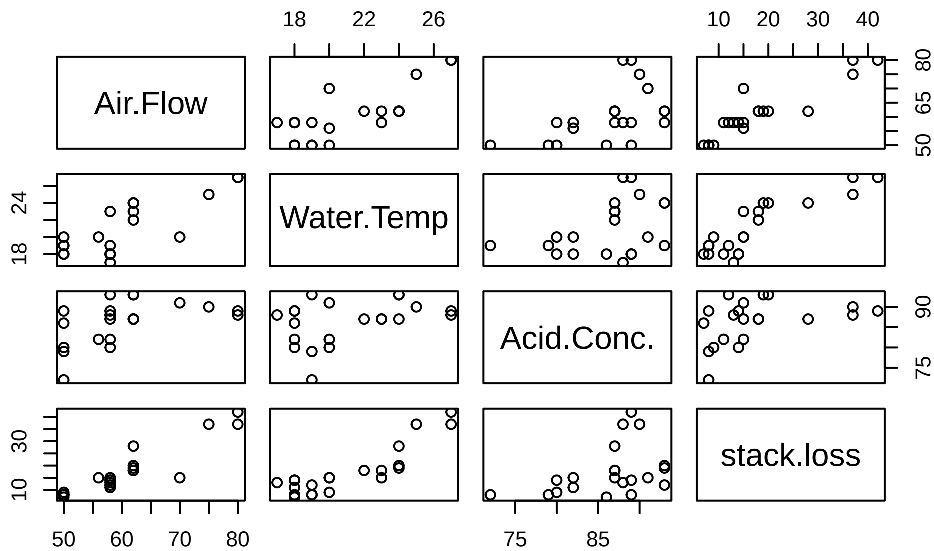
Example 10.1. Consider the stack loss data, it records the percentage of stack loss in the operation of a plant that uses the oxidation of ammonia to produce nitric acid. The data set contains four variables and 21 observations.

Variables - **Air.Flow**: Flow rate of cooling air (in cubic meters per hour) - **Water.Temp**: Cooling water inlet temperature (in degrees Celsius) - **Acid.Conc.**: Acid concentration (percentage) - **stack.loss**: Stack loss (percentage of the ammonia lost)

Regress stack loss on the remaining variables. Compare various robust regression estimates to the OLS estimates. Add a large outlier and repeat the process. What do you observe?

The `rlm` function in the `MASS` package allows us to run robust regression. Huber's loss is used by default.

```
# Computes how many standard errors coeficients are different by between models m1 and m2
compute_movement=function(m1,m2){
  vb=summary(m2)
  return(abs(coef(m2)-coef(m1))/vb$coef[,2])
}

library(MASS)

plot(stackloss)
```




```
#fit robust regression model
# psi argument specifies psi...
# run ?rlm for more details.
OLS1=lm(stack.loss ~ ., stackloss)
RR=rlm(stack.loss ~ ., stackloss)
summary(OLS1)
```

```
Call:
lm(formula = stack.loss ~ ., data = stackloss)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-7.2377 -1.7117 -0.4551  2.3614  5.6978 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -39.9197    11.8960  -3.356  0.00375 ** 
Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
Water.Temp    1.2953     0.3680   3.520  0.00263 ** 
Acid.Conc.   -0.1521     0.1563  -0.973  0.34405    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.243 on 17 degrees of freedom
Multiple R-squared:  0.9136,	Adjusted R-squared:  0.8983 
F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

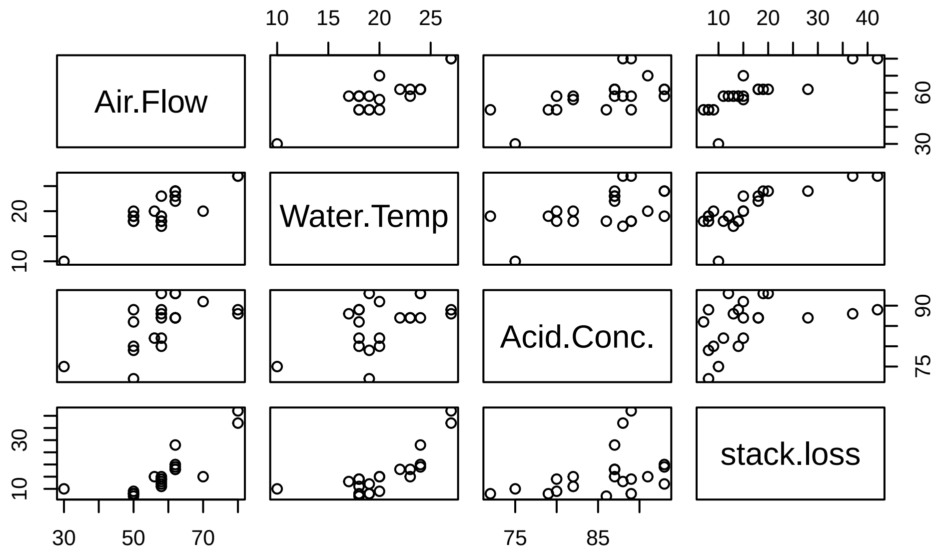
```
summary(RR)
```

```
Call: rlm(formula = stack.loss ~ ., data = stackloss)
Residuals:
     Min       1Q   Median       3Q      Max 
-8.91753 -1.73127  0.06187  1.54306  6.50163 

Coefficients:
            Value    Std. Error t value
(Intercept) -41.0265   9.8073    -4.1832
Air.Flow      0.8294   0.1112     7.4597
Water.Temp    0.9261   0.3034     3.0524
Acid.Conc.   -0.1278   0.1289    -0.9922

Residual standard error: 2.441 on 17 degrees of freedom
```

```
compute_movement(RR,OLS1)
```

```
(Intercept)    Air.Flow  Water.Temp  Acid.Conc. 
 0.09304446  0.84335755  1.00313478  0.15530572 
```

```r
# Air flow and water temp moved one standard error.


# We can compare the estimates to their standard errors.

# Adding the outlier
stackloss2=stackloss
n=nrow(stackloss)
stackloss2[3,]=c(30,10,75,10)
plot(stackloss2)
```




```r
OLS=lm(stack.loss ~ ., stackloss2)
RR_Hu=rlm(stack.loss ~ ., stackloss2, maxit = 100)
RR_Ha=rlm(stack.loss ~ ., stackloss2, psi = psi.hampel, maxit = 100)
RR_Tu=rlm(stack.loss ~ ., stackloss2, psi = psi.bisquare, maxit = 100)
```

```
Warning in rlm.default(x, y, weights, method = method, wt.method = wt.method, :
'rlm' failed to converge in 100 steps
```

```r
# Error from uncorrupted estimates
error=rbind(compute_movement(OLS,OLS1),
compute_movement(RR_Hu,OLS1),
compute_movement(RR_Ha,OLS1),
compute_movement(RR_Tu,OLS1))
rownames(error)=c("OLS","Huber","Hampel","Tukey")
error
```

```
       (Intercept)    Air.Flow Water.Temp Acid.Conc.
OLS      2.3766707 1.317436984  1.4611311  0.4406644
Huber    0.5905931 0.006153154  1.4286807  0.2686428
Hampel   0.1180260 0.047197074  0.4429042  0.1073847
Tukey    0.2180443 0.716273230  2.0518634  0.5167066
```

```r
rowSums(error)
```

```
     OLS    Huber   Hampel    Tukey
5.595903 2.294070 0.715512 3.502887
```

```r
# summary(lm(stack.loss ~ ., stackloss2))
# summary(rlm(stack.loss ~ ., stackloss2))
# summary(rlm(stack.loss ~ ., stackloss2, psi = psi.hampel))
# summary(rlm(stack.loss ~ ., stackloss2, psi = psi.bisquare))


res=rbind(coef(OLS),
coef(RR_Hu),
coef(RR_Ha),
coef(RR_Tu),
coef(OLS1))
rownames(res)=c("OLS","Huber","Hampel","Tukey","Before Corruption")
res
```

```
                  (Intercept)  Air.Flow Water.Temp  Acid.Conc.
OLS                 -11.64681 0.5379730  0.7575544 -0.22099574
Huber               -32.89398 0.7164700  0.7694970 -0.11013524
Hampel              -38.51564 0.7220051  1.1322866 -0.13533894
Tukey               -37.32582 0.8122355  0.5401506 -0.07136436
Before Corruption   -39.91967 0.7156402  1.2952861 -0.15212252
```

10.4 Homework questions

Complete question 15.5 in the textbook.

Exercise 10.1. In the previous examples given in class, add a large outlier and run both least squares and robust regression. How many standard errors did the coefficients move? How large does the outlier need to be for the regression to become corrupted? Repeat the process with a cluster of small outliers.

Exercise 10.2. Why are the OLS estimators susceptible to outliers?

Exercise 10.3. Implement gradient descent and IRLS in R for Huber's loss function.

References

Fox, John, and Georges Monette. 1992. "Generalized Collinearity Diagnostics." *Journal of the American Statistical Association* 87 (417): 178–83. https://doi.org/10.1080/01621459.1992.10475190.

Miller, Don M. 1984. "Reducing Transformation Bias in Curve Fitting." *The American Statistician* 38 (2): 124–26. http://www.jstor.org/stable/2683247.

A Introduction to R software

A.1 Some Basics

R is a Statistical Programming language, it consists of 2 types of objects: data and functions.

```
##Data
x<-2
print(x)
```

```
[1] 2
```

```
##function
log(2)
```

```
[1] 0.6931472
```

Data is stored in variables and can take many forms. To store a value in a variable use "<-", above we set the variable x equal to 2. There are many data types in R, we will go through some of them.

```
#real numbers
num=29.333
num
```

```
[1] 29.333
```

```
#Some math
#adding and subtraction
2+3-2
```

```
[1] 3
```

```
#multiplying and dividing
num<-5*(10/25)
num
```

```
[1] 2
```

```
#Strings
word<-"hello"
word
```

```
[1] "hello"
```

```
word='hello'
```

A.2 Booleans

Booleans take on either TRUE or FALSE values, and can be very useful in R. You can set booleans to the result of a comparison of two data types, some of the syntax is below:

- <,>,<=,>= corresponds to less than, greater than, less than or equal, greater than or equal
- ==, != equals, not equals
- && , written like a&&b where a and b are booleans, it is TRUE is *both* a and b are TRUE
- || , written like a||b where a and b are booleans, it is TRUE if at least *one of* a and b are TRUE

```
#booleans can be initialize in a variety of ways, for example
#must capitalize the true or false
FALSE
```

```
[1] FALSE
```

```
F
```

```
[1] FALSE
```

```
T
```

```
[1] TRUE
```

```
myBoolean<-TRUE
myBoolean
```

```
[1] TRUE
```

```
myBoolean2<- 3<4
myBoolean2
```

```
[1] TRUE
```

```
myBoolean3<-"this"=="that"
myBoolean3
```

```
[1] FALSE
```

```
## && (and) is TRUE if BOTH input booleans are true
## || (or) is TRUE if AT LEAST one input boolean is true
myBoolean4<-myBoolean2&&myBoolean
myBoolean4
```

```
[1] TRUE
```

A.3 Vectors

Vectors in R are used frequently, they are "lists" or "arrays" of all the same data type.

```
##vectors are created with c(data,data,data)
myVector<-c(2,3,4,5,6,7,8,9,10)
myVector
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```r
#a:b is a shortcut for a sequence from a to b adding 1
#you can create vectors of sequences using seq(), for more type ?seq in the console
myVector2<-2:10
myVector2
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```r
as.numeric(2:10)
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```r
as.double(2:10)
```

```
[1]  2  3  4  5  6  7  8  9 10
```

```r
myVector2<-rep(NA,l=20)


#These do not have to be numbers, they can be vectors, Strings, booleans...
myVector<-c(myVector,myVector)
myVector
```

```
 [1]  2  3  4  5  6  7  8  9 10  2  3  4  5  6  7  8  9 10
```

```r
myVector3<-c("this","is","a","vector","of","strings")
myVector3
```

```
[1] "this"    "is"      "a"       "vector"  "of"      "strings"
```

```r
#access elements with square brackets []
myVector[1]
```

```
[1] 2
```

```r
#more advanced accesssing
#access elements 1 to 5
myVector[1:5]
```

```
[1] 2 3 4 5 6
```

```r
#access elements 1, 4 and 6
myVector[c(1,4,6)]
```

```
[1] 2 5 7
```

```r
#access elements that are greater than 2
myVector[myVector>2]
```

```
 [1]  3  4  5  6  7  8  9 10  3  4  5  6  7  8  9 10
```

```r
myVector[-c(1,4,6)]
```

```
 [1]  3  4  6  8  9 10  2  3  4  5  6  7  8  9 10
```

We can perform mathematical operations and comparisons on vectors

```r
x<-1:10
x
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```r
#adds 1 to every element
x+1
```

```
 [1]  2  3  4  5  6  7  8  9 10 11
```

```r
#this works for comparisons
x<4
```

```
 [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
x[x<4]
```

```
[1] 1 2 3
```

```r
#multiplies element 1 to element 1 of second vectors
x*-(1:10)
```

```
 [1]   -1   -4   -9  -16  -25  -36  -49  -64  -81 -100
```

```r
#beware repetition
x-c(1,2)
```

```
 [1] 0 0 2 2 4 4 6 6 8 8
```

```r
# mathematical operations on the vector apply to each element

#squares each element
x^2
```

```
 [1]   1   4   9  16  25  36  49  64  81 100
```

```r
#log each element
log(x)
```

```
 [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595 1.9459101
 [8] 2.0794415 2.1972246 2.3025851
```

```r
#Example: Dot Product
x<-c(1,2,3)
y<-c(2,5,8)
#sum adds the elements of the vector together
sum(x*y)
```

```
[1] 36
```

A.4 Matrices

You can also use matrices in R.

```r
#you can create a matrix with matrix(vector of data,nrow=number of rows,ncol=number of col
#You can see it will fill in the data down the columns first
myMatrix<-matrix(1:9,nrow=3,ncol=3); myMatrix
```

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

```r
myMatrix
```

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

```r
#rbind and cbind add a row or column repectively to the matrix
#you can create matrices with rbind(rowvector1,rowvector2,...), or with cbind(column vecto

myMatrix<-rbind(c(2,3,4),c(3,4,5),c(1,2,3))
myMatrix
```

```
     [,1] [,2] [,3]
[1,]    2    3    4
[2,]    3    4    5
[3,]    1    2    3
```

```r
myMatrix2<-cbind(c(1,2,3),c(4,5,6),c(7,8,9))
myMatrix2
```

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

```
myMatrix3<-cbind(myMatrix2,c(10,11,12))
myMatrix3
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```
myMatrix3<-cbind(c(10,11,12),myMatrix2)
```

We can also do Matrix math:

```
#again math functions apply to every element
myMatrix^2
```

```
     [,1] [,2] [,3]
[1,]    4    9   16
[2,]    9   16   25
[3,]    1    4    9
```

```
#multiply with '%*%
myMatrix2%*%myMatrix
```

```
     [,1] [,2] [,3]
[1,]   21   33   45
[2,]   27   42   57
[3,]   33   51   69
```

```
#we can find the inverse with 'solve()
X<-matrix(c(1,0,1,-2,3,0,1,4,2),nrow=3)
X
```

```
     [,1] [,2] [,3]
[1,]    1   -2    1
[2,]    0    3    4
[3,]    1    0    2
```

```
solve(X)
```

```
     [,1] [,2] [,3]
[1,] -1.2 -0.8  2.2
[2,] -0.8 -0.2  0.8
[3,]  0.6  0.4 -0.6
```

```
#check dimension
dim(X)
```

```
[1] 3 3
```

```
#We can also transpose with t()
t(X)
```

```
     [,1] [,2] [,3]
[1,]    1    0    1
[2,]   -2    3    0
[3,]    1    4    2
```

```
#Some times to multiply vectors we have to trun them into matrix types
myVector<-c(1,2,3)
newM<-matrix(myVector,ncol=1)
```

A.5 Functions

Functions are objects that take an input and transform it into some output, just like in mathematics. We have already seen some, such as `log()`.

They are called with this format `output<-functionName(input)`.

- The input is called *parameters*, and there can be many parameters
- parameters are usually described in the documentation
- the output is what the function *returns*
- functions can only return 1 object, but this includes a list… so it could return many objects in the form of a list object

R has many, many functions, to learn more about a function type `?functionName` and the documentation will come up.

```
#A simple function
#here the function log is called, with the parameter 2, and the output is stored in the va
x<-log(2)
x
```

```
[1] 0.6931472
```

```
#A more complicated function
#What are the parameters?
#not rep(a,n) gives a vector of size n where all elements are a
s<-sample(x=1:10,size=4,replace=TRUE,prob=rep(1/10,10))
s
```

```
[1] 9 3 6 1
```

We have seen other people's functions but we can also make our own! Let's see an example first:

```
#recall the dot product example...
dotProd=function(a,b){
  value<-sum(a*b)
  return(value)
}
#calling our function
dotProd(x,y)
```

```
[1] 10.39721
```

What exactly does this code say?

- We stored the function in the variable `dotProd`
- to tell the compiler we are creating a function, we use the keyword function
- we specify the parameters in round brackets ()
- we put the names of the parameters in the () only, not what data type we expect them to be
- inside curly brackets, we put the code that the function will run when it is called
- `return()` ends the function, and sends back the variable in the brackets

Back to built in functions… `R` is a statistical software, what does that mean? It already includes many common statistical functions! For most common distributions there are functions for the pdf, cdf, inverse cdf as well as one to get a sample from that distribution. The syntax is in the format: `dDistName(x,parameters)`, `pDistName(x,parameters)`, `qDistName(x,parameters)` and `rDistName(x,parameters)` respectively. This will make more sense in the example below…

```
#The normal distribution, sd is the standard deviation
#pdf
dnorm(c(2,3,5),mean=0,sd=1)
```

```
[1] 5.399097e-02 4.431848e-03 1.486720e-06
```

```
#cdf
pnorm(c(2,3,5),mean=0,sd=1)
```

```
[1] 0.9772499 0.9986501 0.9999997
```

```
#inverse cdf
qnorm(c(0.2,.5,.3),mean=0,sd=1)
```

```
[1] -0.8416212  0.0000000 -0.5244005
```

```
#random sample of size 10
rnorm(10,mean=0,sd=1)
```

```
 [1]  0.1144319  1.4437119  1.2011116 -0.4250923  0.4535709 -1.0646861
 [7] -0.2045586 -0.8419064  0.8100511 -0.5563228
```

A.6 Plotting

R is very good for plotting! There are many types of plots in R, here are some useful plotting functions, this list is not exhaustive…

- `plot(x,y,...)` produces a scatter plot.
- `abline(a=intercept,b=slope,...)`
- `curve(expr,...)` evaluates an expression along a grid to create a curve

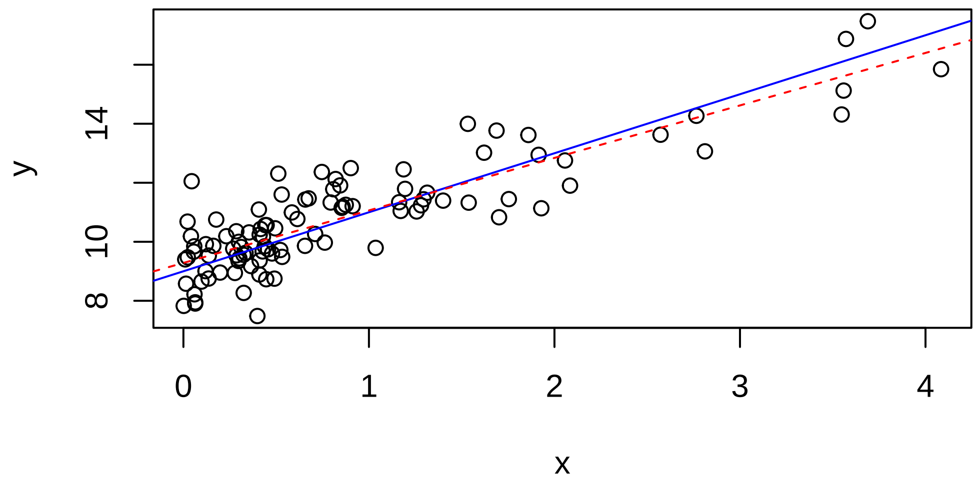
- `hist(data)` creates a histogram

Plot functions have many parameters, some include `col` which changes the color and add which should be set to `TRUE` if the plot should be added to the existing plot. The best way to learn plots is with examples, I have included a regression example below.

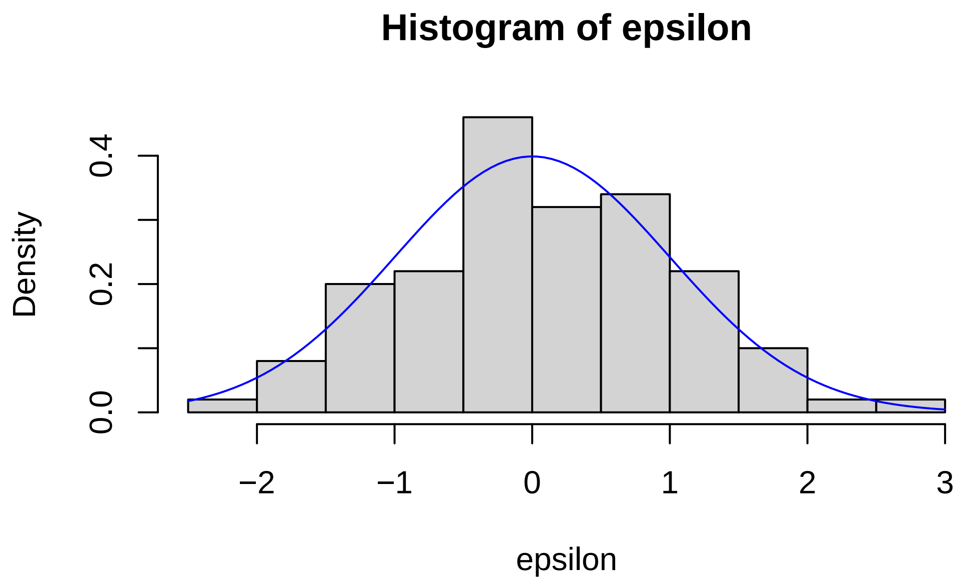
```r
#simulate errors
epsilon<-rnorm(100)
x<-rexp(100)
y<-9+2*x+epsilon

#scatter plot with true line
plot(x,y)
abline(a=9,b=2,col="blue")

#least squares line
lmm<-lm(y~x)
summary(lmm)
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.50910 -0.67030 -0.00423  0.49445  2.68962 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   9.2831     0.1325   70.04   <2e-16 ***
x             1.7791     0.1057   16.83   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9582 on 98 degrees of freedom
Multiple R-squared:  0.7429,	Adjusted R-squared:  0.7403 
F-statistic: 283.2 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
abline(lmm$coefficients[1],lmm$coefficients[2],col="red",lty=2)
```




```
#histogram of residuals
hist(epsilon,freq = F)
#x is what you want to evaluate the grid along
curve(dnorm(x),add=T,col="blue")
```



A.7 If Statements

If statements are essential in programming, and they are a form of 'Control Structure'. They take the form `if(boolean variable){some task}`.
When the computer runs through the code, it checks if the boolean value is `TRUE`, and if it is, it executes the code in the curly brackets, code in curly brackets is called a *block*. A simple example...

```
jim<-"nice"

if(jim=="nice"){
  alice="nice"
}
```

Placing an `else{some code}` after the if statement will execute the code in it's block if the code in the above if statement *was not* executed. The if and else must be in the same block so I have surrounded them in curly brackets.

```
jim<-"nice"
##same block
{
```

```r
if(jim=="nice"){
  alice="nice"
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "nice"
```

```r
jim<-"mean"
##same block
{
if(jim=="nice"){
  alice="nice"
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "not nice"
```

You may also use `else if(boolean){block}`, which executes it's block if the above (else) if statement(s) did not execute. See below:

```r
jim<-"okay"
##same block
{
if(jim=="nice"){
  alice="nice"
}
else if(jim=="okay"){
  alice="okay"
}
  #Here if jim is not okay or nice, then we check if he is neutral.
else if(jim=="neutral"){
  alice="neutral"
```

```
}
else{
  alice="not nice"
}
  alice
}
```

```
[1] "okay"
```

Lastly you may put if statements inside of other if statements, called 'nested ifs'.

```
jim<-"nice"
##same block

if(jim=="nice"){
  alice=sample(c("nice","not nice"),1)
  if(alice=="nice"){
    print(alice)
  }
  else{
    print(alice)
  }
}
```

```
[1] "nice"
```

A.8 Loops

Loops execute operations within their blocks repeatedly. There are 2 types of loops you will generally use, for loops and while loops. For loops repeat the block a set number of times, while while loops repeat until a condition is satisfied. You can also nest loops, like if statements.

```
#calculate 2 to the power of ten
x<-1
#this reads for i in 1 to 10, this can be any vector that i loops through, not just a sequ
for(i in 1:10){
  x<-x*2
}
```

```r
x
```

```
[1] 1024
```

```r
for(i in 1:10){
  x<-x+i
}

vec=2:5

for(i in vec){
  x<-x+i
}


#calculate power of 2 less that 1000
x<-1
while(2*x<1000){
  x<-x*2
}
x
```

```
[1] 512
```

```r
#nested loop
for(i in c(10,9,8,7,6,5,4,3,2,1)){
  v<-NULL
  for(j in 1:i){
    v<-c(v,"*")
  }
  print(v)
}
```

```
 [1] "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*" "*"
[1] "*" "*" "*" "*" "*"
```

```
[1] "*" "*" "*" "*"
[1] "*" "*" "*"
[1] "*" "*"
[1] "*"
```

You can also use the replicate function, which replicates a line of code a specified number of times. This gives a 10 by 5 matrix.

```r
replicate(5,rnorm(10))
```

```
             [,1]       [,2]       [,3]       [,4]       [,5]
 [1,] -0.87792227 -1.3207310  0.6672619  0.6172735 -0.6970302
 [2,]  1.31751836  0.1139693  0.4894577 -0.5204016  0.8965531
 [3,] -0.38479214  0.6237071  0.4576051 -0.3716644  0.2474148
 [4,] -0.57802144  0.3996335  0.6288974 -0.1433156  0.4785685
 [5,]  0.82808340 -0.9504758  0.3665279 -0.9369322  0.9818202
 [6,] -1.10535449  1.1116529  0.6306810  2.1061986  0.1538709
 [7,]  0.84640052  1.5537925  1.5486722 -0.2716292 -0.6203708
 [8,] -0.06170204  1.0483459 -0.5265191 -2.0115232 -1.1135819
 [9,] -1.64816742  0.6931981  0.5275307 -0.3247686 -0.7252148
[10,] -1.06726622  2.4724317 -1.1681384  0.1762106 -1.0338485
```

Similar functions include `sapply()` and `apply()`. `sapply(X,FUN,...)` applies the function that the parameter `FUN` is set to to individual elements of a vector. `apply(X,MARGIN,FUN,...)` applies `FUN` to the rows or columns depending on what `MARGIN` is set to, 1 for rows and 2 for columns.

A.9 Coverage Probability Example

Here we generate 10000 samples of size 100 from the exponential distribution, with $\lambda = 2$. We calculate 10000 confidence intervals for $1/\lambda$ with \$ =\$1%, using the normal approximation:

$$\sqrt{n}(\bar{X} - 1/\lambda) \sim N(0, 1/\lambda^2)$$

and interval:

$$(\bar{X} - t_{99}(0.005) * S/\sqrt{n}, \bar{X} + t_{99}(0.005) * S/\sqrt{n})$$

We then check the proportion of intervals that contain the true value of $1/\lambda$.

```r
#10000 samples, each of size 100 from the exponential distribution
x<-replicate(10000,rexp(100,rate=2))
#x is 100 by 10000, each column is a sample
dim(x)
```

```
[1]   100 10000
```

```r
#calculate sample variances
S_Vector<-apply(x,2,sd);

# S_Vector

#get the t value
tval<-qt(1-0.005,99)
#calculate the means

means<-apply(x,2,mean); length(means)
```

```
[1] 10000
```

```r
# lower and upper bounds
lower<-means-S_Vector*tval/10
upper<-means+S_Vector*tval/10
intervals<-rbind(lower,upper)
#example interval
intervals[,1]
```

```
    lower     upper
0.3135737 0.5448307
```

```r
#we now check each interval to see if it contains the mean
successes<-0
for(i in 1:ncol(intervals)){
  #if 0.5 is in the interval, add 1
  if((intervals[1,i]<0.5)&&(intervals[2,i]>0.5))
    successes<-successes+1
}
#here is the coverage probability...
```

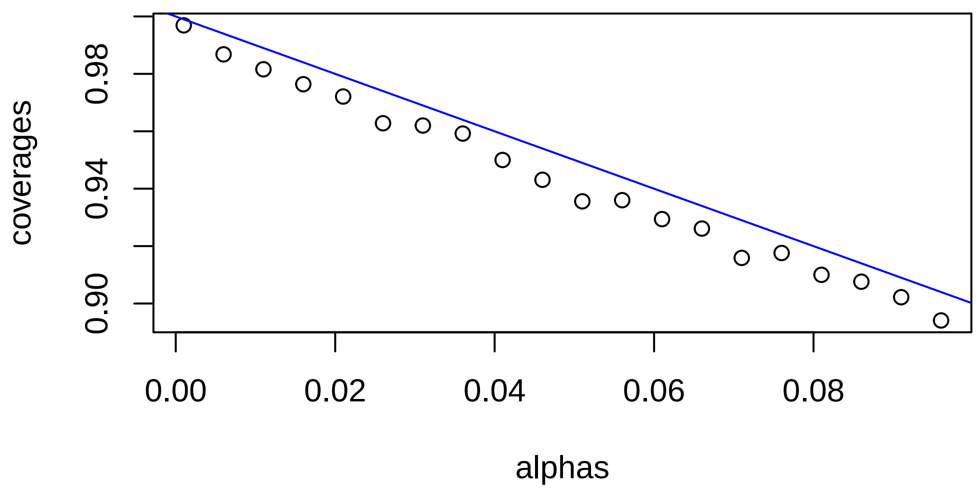
```
coverage.prob<-successes/ncol(intervals)
coverage.prob
```

```
[1] 0.983
```

Something more advanced...

```
#Vectorzing the function changes the way the function calculates when it is a passed a vec
#it will run the function once per element if it is vectorized instead of passing the vect
getCovProb<-Vectorize(function(alpha){
#10000 samples, each of size 100 from the exponential distribution
  x<-replicate(10000,rexp(100,rate=2))
#x is 100 by 10000, each column is a sample
  dim(x)
#calculate sample variances
  S_Vector<-apply(x,2,sd)
#get the t value
  tval<-qt(1-alpha/2,99)
#calculate the means
  means<-apply(x,2,mean)
# lower and upper bounds
  lower<-means-S_Vector*tval/10
  upper<-means+S_Vector*tval/10
  intervals<-rbind(lower,upper)
#example interval
  intervals[,1]

#we now check each interval to see if it contains the mean
successes<-0
for(i in 1:ncol(intervals)){
  #if 0.5 is in the interval, add 1
  if((intervals[1,i]<0.5)&(intervals[2,i]>0.5))
    successes<-successes+1
}
#here is the coverage probability...
coverage.prob<-successes/ncol(intervals)
return(coverage.prob)
})
#here we find the coverage probability for many alphas
alphas<-seq(from=0.001,to=0.1,by=0.005)
coverages<-getCovProb(alphas)
```

```r
#adds a scatter plot
plot(alphas,coverages)
#adds a line
abline(a=1,b=-1,col="blue")
```



For more information you can visit here . It is also very easy to find tutorials on the web (Youtube is good), you could also look at the book by Lafaye, Drouilhet and Liquet (2013).