Assumptions

Ali Thursland and Mary Helen Wood

12/10/2018

== Setup ==

```
#Setup
library("dplyr")
library("ggplot2")
library("broom")
library("knitr")
library("cowplot")
library("readr")
library("dslabs")
library("varhandle")
library("olsrr")
library("car")
```

== Read in Data ==

```
airbnb <- read_csv("https://raw.githubusercontent.com/athursland/STA-210/master/finalairbnb.csv")
```

Separate training and testing

```
#80% of the sample size
smp_size <- floor(0.80 * nrow(airbnb))

#set the seed to make your partition reproducible
set.seed(123456)
train_ind <- sample(seq_len(nrow(airbnb)), size = smp_size)

train.airbnb <- airbnb[train_ind, ]
test.airbnb <- airbnb[-train_ind, ]
```

== Final Model ==

```
stepwise.interactions.model <- lm(logprice ~ cleaning_fee * accommodates + availability_30 * minimum_ni
+ host_is_superhost
+ room_type
+ accommodates
+ cleaning_fee
+ minimum_nights
+ availability_30
```

```r
+ log.reviews_per_month
+ cancellation_policy, data=train.airbnb)

kable(tidy(stepwise.interactions.model), format="markdown", digits = 4)
```

term	estimate	std.error	statistic	p.value
(Intercept)	4.3820	0.0482	90.8472	0.0000
cleaning_fee	0.0038	0.0005	7.8207	0.0000
accommodates	0.0982	0.0072	13.6651	0.0000
availability_30	0.0030	0.0015	1.9919	0.0466
minimum_nights	-0.0189	0.0020	-9.5745	0.0000
host_is_superhostt	0.0590	0.0240	2.4610	0.0140
room_typePrivate room	-0.2604	0.0272	-9.5797	0.0000
room_typeShared room	-1.1917	0.1544	-7.7199	0.0000
log.reviews_per_month	-0.2061	0.0185	-11.1291	0.0000
cancellation_policymoderate	0.0730	0.0280	2.6076	0.0092
cancellation_policystrict_14_with_grace_period	0.1359	0.0306	4.4397	0.0000
cancellation_policysuper_strict_30	0.0727	0.1023	0.7110	0.4772
cancellation_policysuper_strict_60	0.7443	0.0841	8.8522	0.0000
cleaning_fee:accommodates	-0.0002	0.0001	-3.6396	0.0003
availability_30:minimum_nights	0.0003	0.0001	3.2409	0.0012

```r
kable(glance(stepwise.interactions.model))
```

	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.r
value	0.5742094	0.5702486	0.4000556	144.9715	0	15	-756.6985	1545.397	1630.621	240.8669	

== Assumptions ==

There are 5 assumptions for multiple linear regression: 1. Linearity 2. Constant variance 3. Normality 4. Independence

Additionally, we must avoid outliers/multicollinearity in our final model.

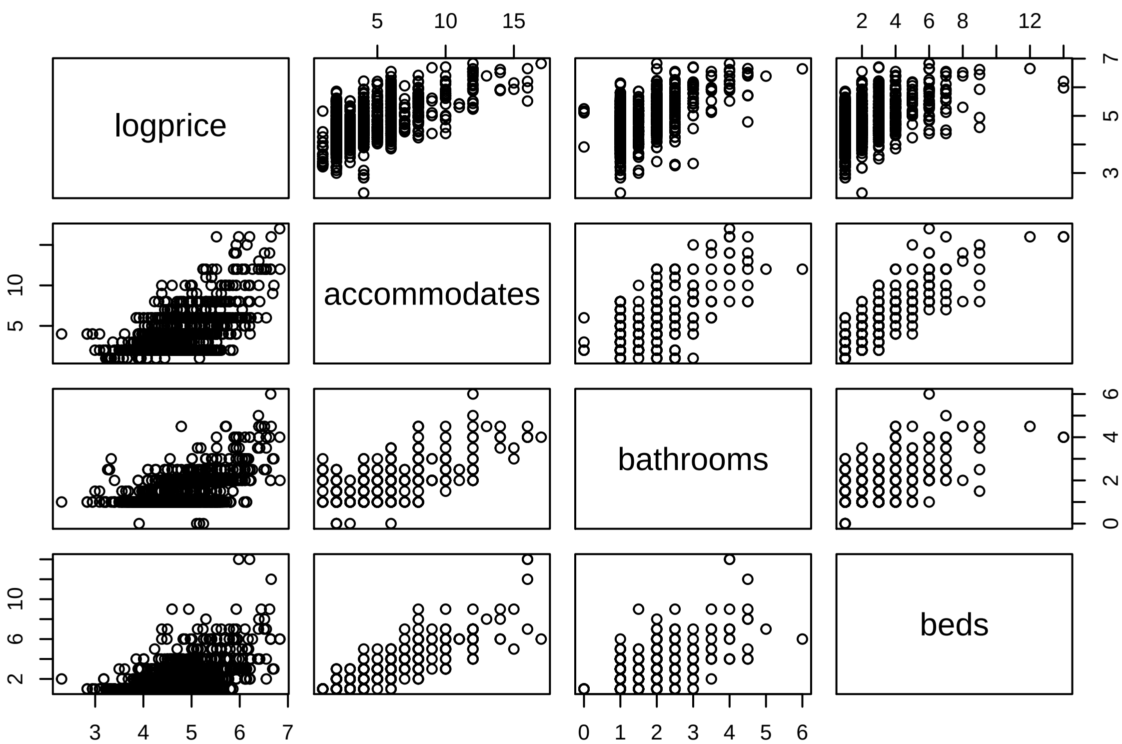
First, we will address multicollinearity. We chose not to include all three of the variables accommodates, bathrooms and beds because there was obvious multicollinearity present between all of them - which makes sense when you think about it. After looking at the p-values of all three variables in a full model, we chose to omit beds and bathrooms for having high p-values, and self-selected accommodates in the model.

```r
#Example model including beds and bathrooms
not.final.model <- lm(logprice ~ cleaning_fee * accommodates + availability_30 * minimum_nights
+ host_is_superhost
+ room_type
+ accommodates
+ cleaning_fee
+ minimum_nights
+ availability_30
+ bathrooms
+ beds
+ log.reviews_per_month
+ cancellation_policy, data=train.airbnb)
```

```r
#Check VIF for accommodates and bathrooms
tidy(vif(not.final.model))
```

```
## # A tibble: 12 x 4
##    .rownames                         GVIF    Df GVIF..1..2.Df..
##    <chr>                            <dbl> <dbl>           <dbl>
##  1 cleaning_fee                      6.48     1            2.55
##  2 accommodates                      6.38     1            2.53
##  3 availability_30                   1.35     1            1.16
##  4 minimum_nights                    2.55     1            1.60
##  5 host_is_superhost                 1.30     1            1.14
##  6 room_type                         1.52     2            1.11
##  7 bathrooms                         2.44     1            1.56
##  8 beds                              4.88     1            2.21
##  9 log.reviews_per_month             1.51     1            1.23
## 10 cancellation_policy               1.61     4            1.06
## 11 cleaning_fee:accommodates         7.78     1            2.79
## 12 availability_30:minimum_nights    2.52     1            1.59
```

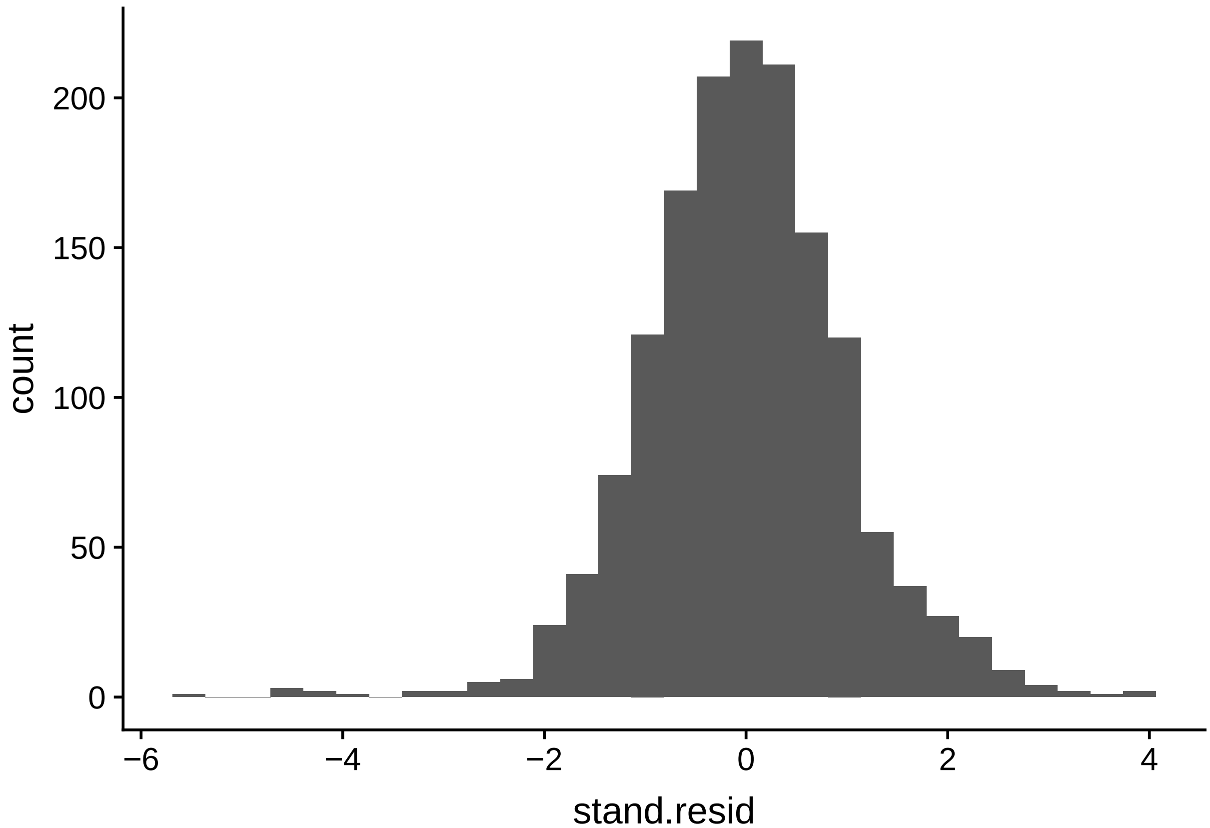
```r
#Pairs plots of accommodates and bathrooms
pairs(logprice ~ accommodates + bathrooms + beds, data = train.airbnb)
```



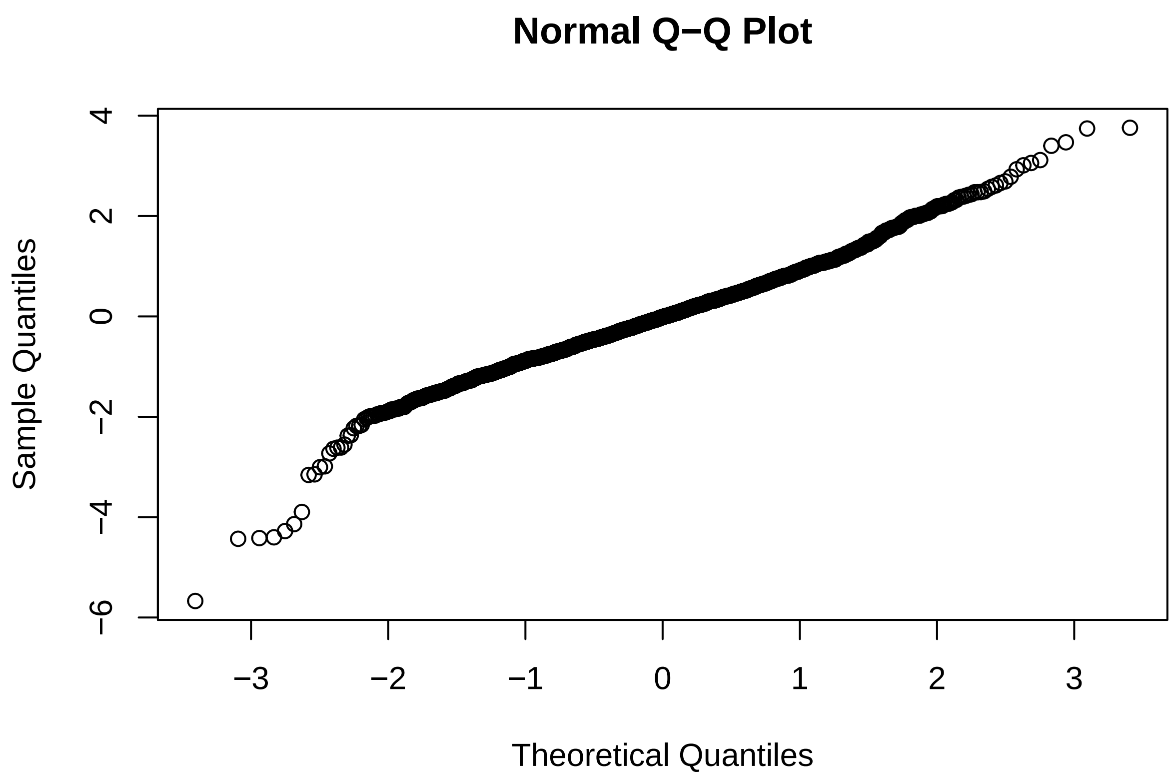

```r
#Standard residuals and predicted values
train.airbnb <- train.airbnb %>% mutate(stand.resid = rstandard(stepwise.interactions.model),
                                        pred = predict(stepwise.interactions.model))
```

```r
#Histogram of the standard residuals
ggplot(data = train.airbnb, aes(x=stand.resid)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#QQ-norm plot
qqnorm(train.airbnb$stand.resid)
```

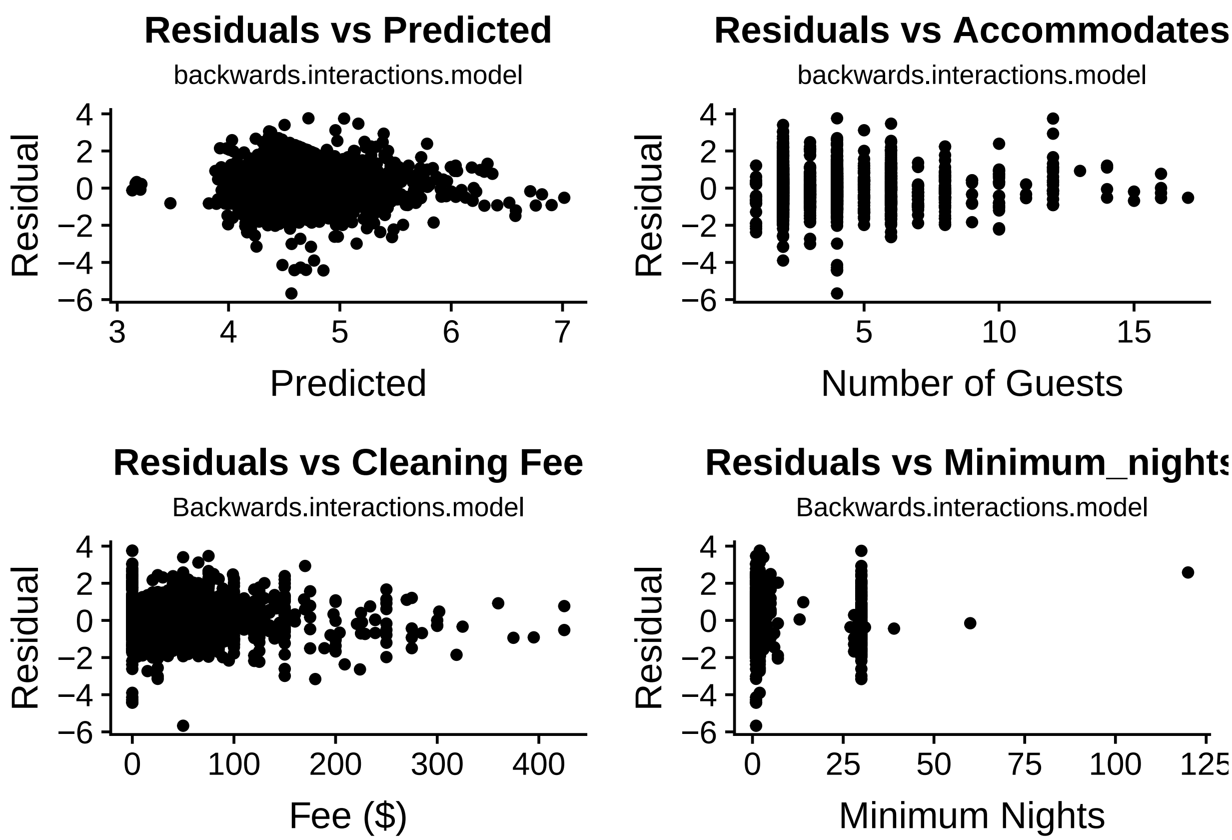


According to the histogram of our standardized residuals and our QQ-norm plot, our standardized residuals appear to be approximately normally distributed. This means that our normality assumption has been satisfied.

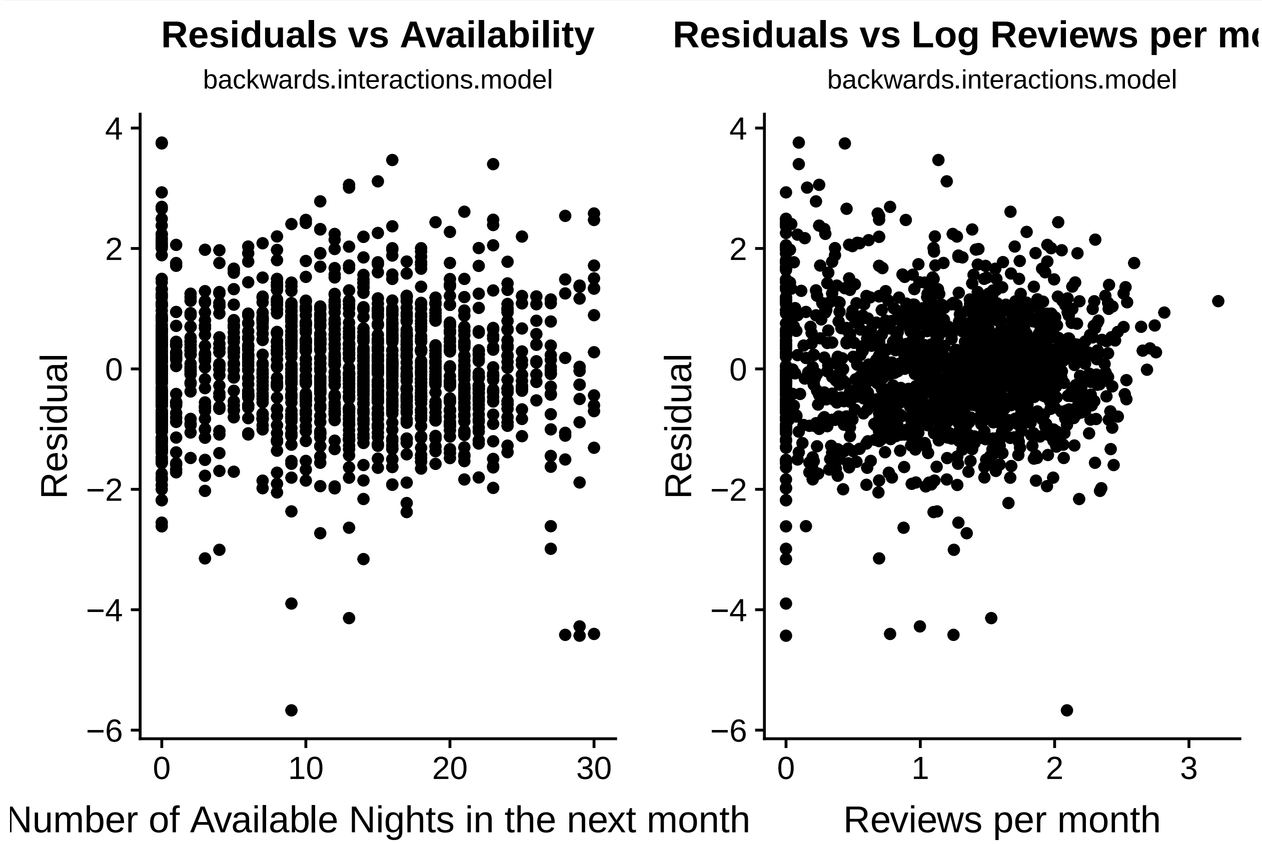
Next, we will plot our residuals against each of the numeric explanatory variables.

```
#Residuals vs. predicted
p1 <- ggplot(data = train.airbnb, aes(x=pred, y=stand.resid)) + geom_point() +
  labs(x="Predicted", y="Residual", title="Residuals vs Predicted",
subtitle=("backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))

#Residuals vs. accommodates
p2 <- ggplot(data = train.airbnb, aes(x=accommodates, y=stand.resid)) + geom_point() +
  labs(x="Number of Guests", y="Residual", title="Residuals vs Accommodates",
subtitle=("backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))

#Residuals vs. cleaning_fee
p3 <- ggplot(data = train.airbnb, aes(x= cleaning_fee, y=stand.resid)) + geom_point() +
  labs(x="Fee ($)", y="Residual", title="Residuals vs Cleaning Fee",
subtitle=("Backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))

#Residuals vs. minimum_nights
p4 <-ggplot(data = train.airbnb, aes(x=minimum_nights, y=stand.resid)) + geom_point() +
  labs(x="Minimum Nights", y="Residual", title="Residuals vs Minimum_nights",
subtitle=("Backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))

#Residuals vs. availability_30
p5 <- ggplot(data = train.airbnb, aes(x=availability_30, y=stand.resid)) + geom_point() +
  labs(x="Number of Available Nights in the next month", y="Residual", title="Residuals vs Availability
subtitle=("backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))

#Residuals vs. log.reviews_per_month
p6 <- ggplot(data = train.airbnb, aes(x=log.reviews_per_month, y=stand.resid)) + geom_point() +
  labs(x="Reviews per month", y="Residual", title="Residuals vs Log Reviews per month",
subtitle=("backwards.interactions.model"))+
theme(plot.title = element_text(hjust = 0.5,size=14),
plot.subtitle=element_text(hjust=0.5,size=10))
```

```
#plot all of the previous graphs
plot_grid(p1,p2,p3,p4)
```

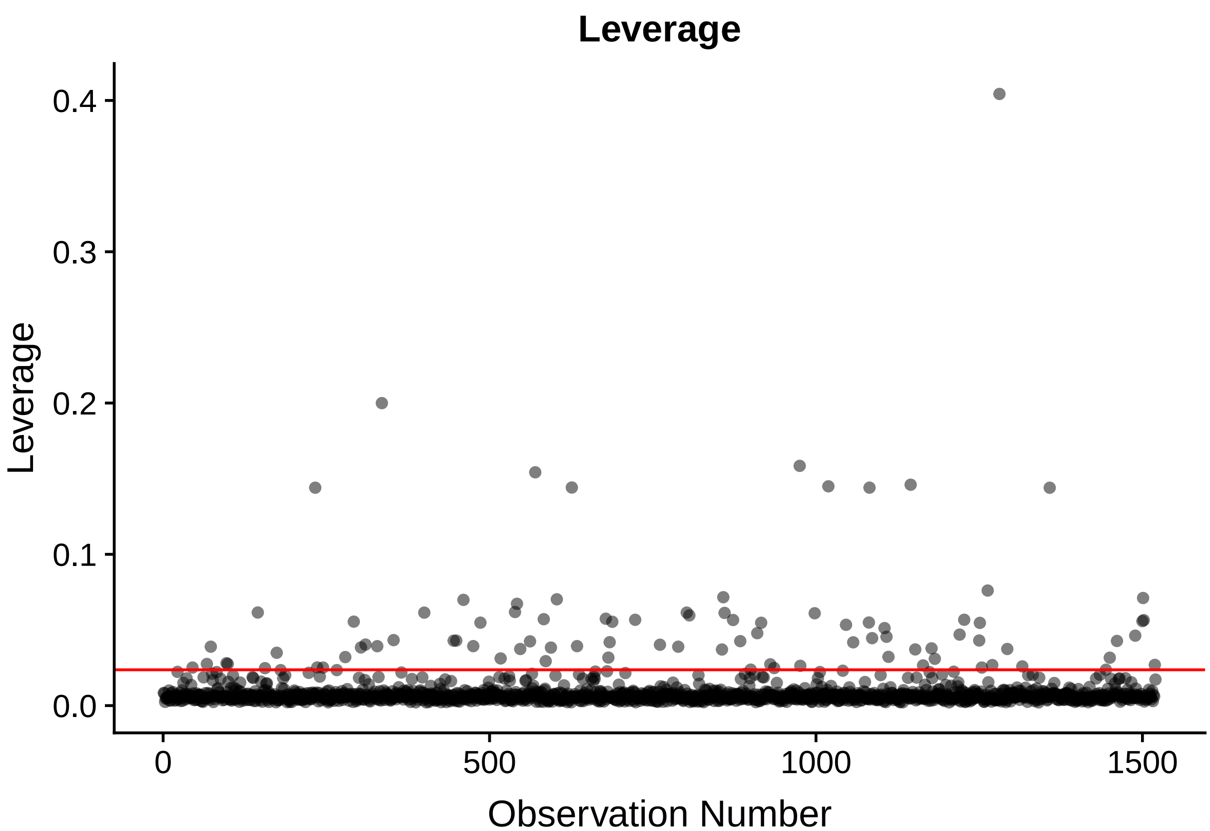



```r
plot_grid(p5,p6)
```

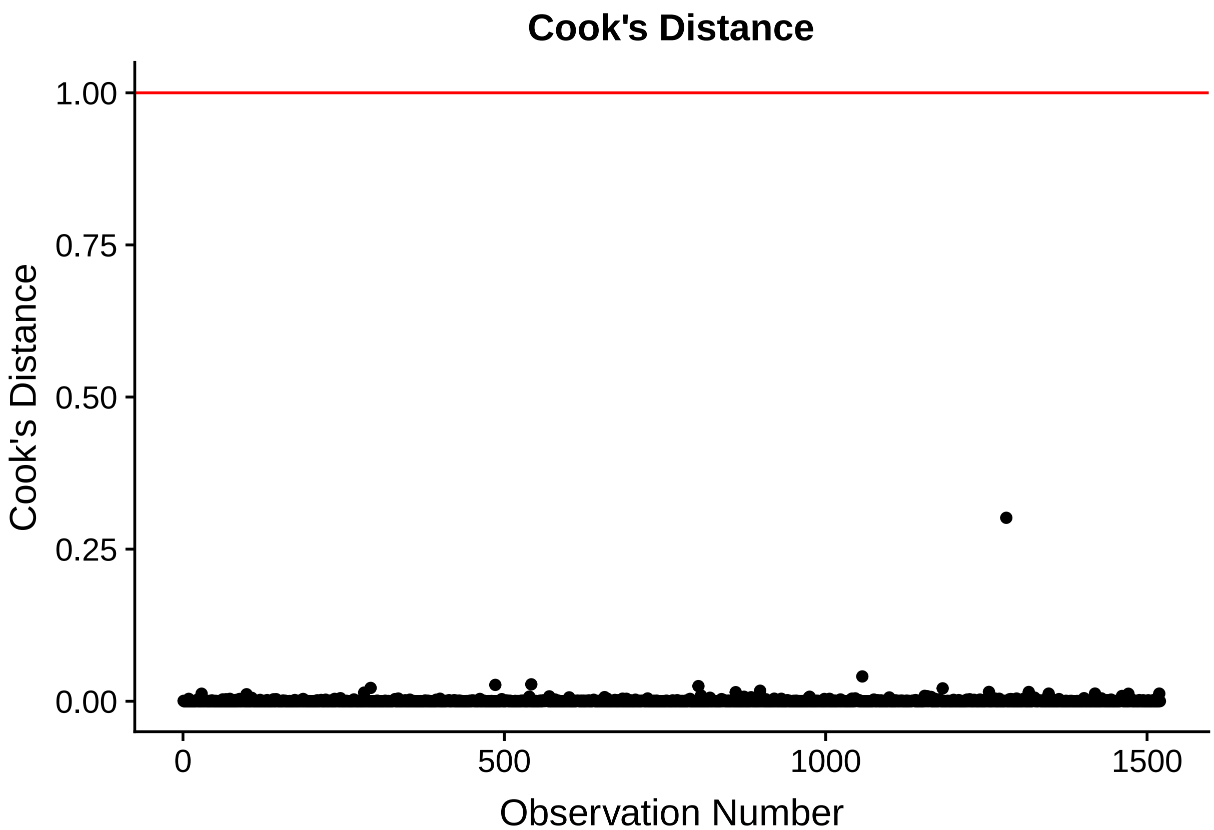


All of our residuals are approximately randomly distributed. There are some observations that appear like potential outliers, but we will address this later in our assumptions. In addition, we have an extremely large number of observations, so the effect of any few outliers would likely be minimal. There are no distinct patterns visible in any of our plots. Therefore, our constant variance assumption has been met.

```
#Calculate leverage, cook's distance and the observation number
train.airbnb <- train.airbnb %>%
  mutate(leverage = hatvalues(stepwise.interactions.model),
         cooks = cooks.distance(stepwise.interactions.model),
         obs.num = row_number())

#Plot of leverage
ggplot(data=train.airbnb, aes(x=obs.num,y=leverage)) +
  geom_point(alpha=0.5) +
  geom_hline(yintercept=36/1520,color="red")+
  labs(x="Observation Number",y="Leverage",title="Leverage")
```




```
#Plot of Cook's Distance
ggplot(data=train.airbnb, aes(x=obs.num,y=cooks)) +
  geom_point() +
  geom_hline(yintercept=1,color="red")+
  labs(x="Observation Number",y="Cook's Distance",title="Cook's Distance")
```



Although there are quite a few observations with large leverage, according to our Cook's Distance, none of these are influential.