Bike Sharing

Dataset Analysis and Prediction

Dataset

Time information

Weather information

Target variable

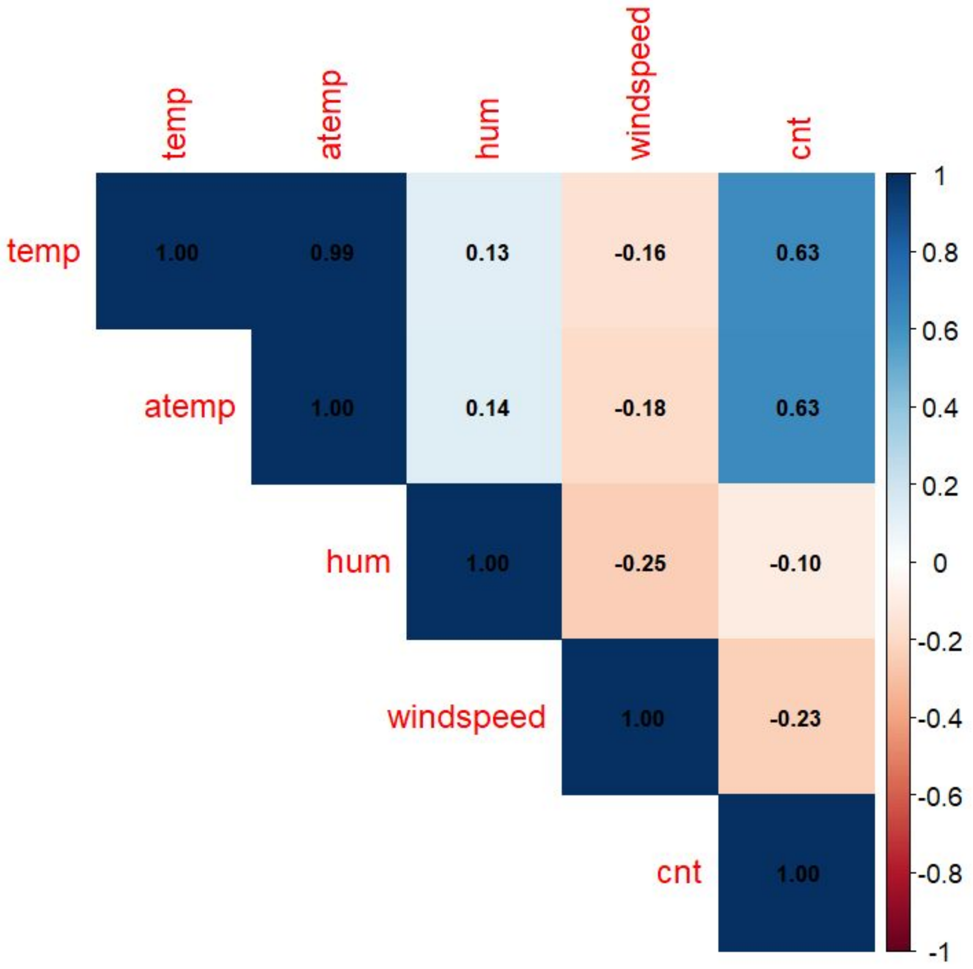
```
'data.frame':	731 obs. of  16 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : chr  "2011-01-01" "2011-01-02" "2011-01-03" "2011-01-04" ...
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
 $ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
 $ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
 $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
 $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
 $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
 $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
 $ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
 $ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
 $ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

- Remove “**instant**” and “**dteday**”
- No NA

```
> sum(is.na(df))
[1] 0
```

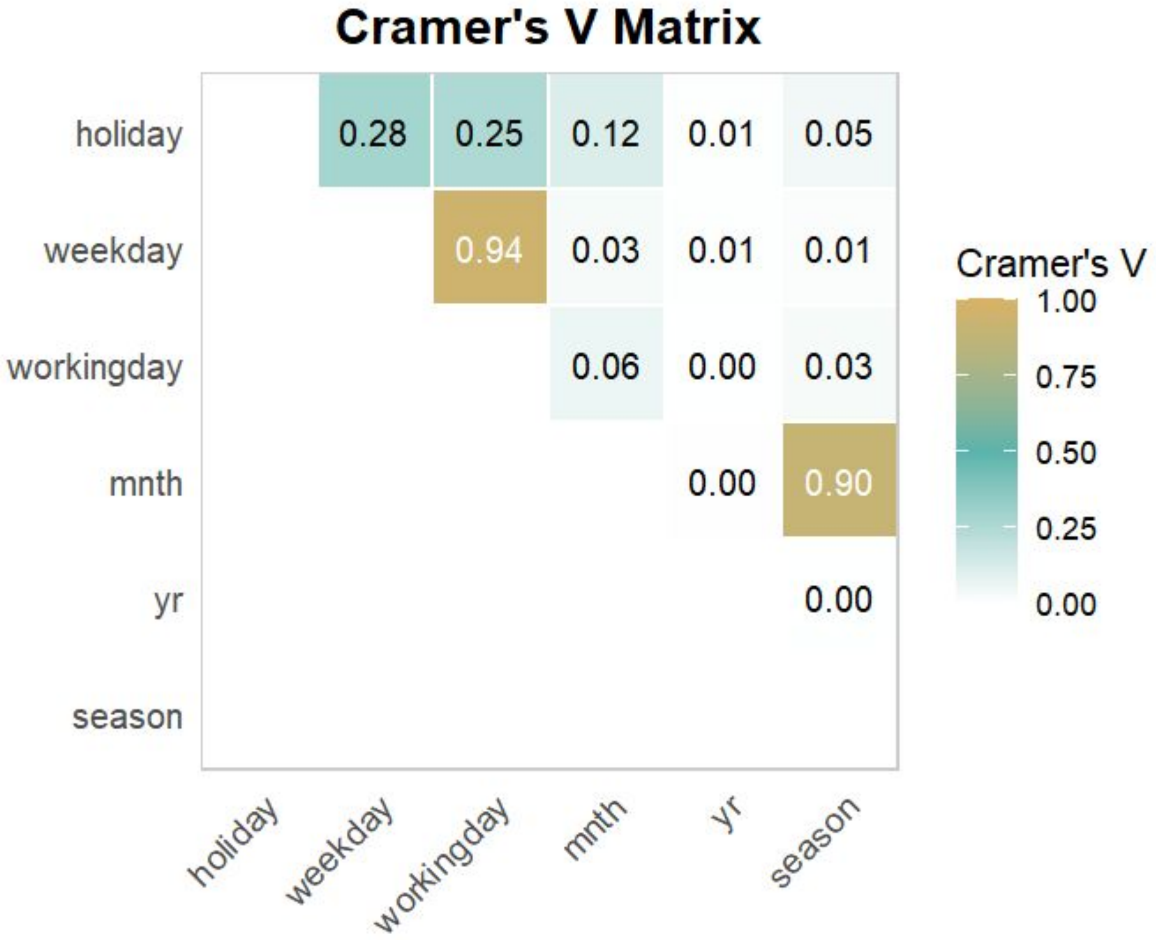
Pearson Correlation

- ***atemp*** and **temp** are highly correlated
- *Remove* ***atemp***



Categorical Correlations

- ***weekday*** and ***workingday*** are highly correlated, VIF analysis will fail. Drop ***workingday***.



Factorize Categorical Variables

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	hum	windspeed	casual	registered	cnt
1	Spring	2011	January	No	Saturday	No	Mist	0.344167	0.805833	0.1604460	331	654	985
2	Spring	2011	January	No	Sunday	No	Mist	0.363478	0.696087	0.2485390	131	670	801
3	Spring	2011	January	No	Monday	Yes	Clear	0.196364	0.437273	0.2483090	120	1229	1349
4	Spring	2011	January	No	Tuesday	Yes	Clear	0.200000	0.590435	0.1602960	108	1454	1562
5	Spring	2011	January	No	Wednesday	Yes	Clear	0.226957	0.436957	0.1869000	82	1518	1600
6	Spring	2011	January	No	Thursday	Yes	Clear	0.204348	0.518261	0.0895652	88	1518	1606

- For better visualization understanding
- Modelling will convert factorized variables to one-hot encoding.

Linear Regression Base Model

	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0.859	0.852	741.	126.	1.29e-217	27	-4697.	9453.	9580.	306868833.

- R^2 at 85.9 % → relatively high variance is explained.
- R^2 and adjusted R^2 are very close → good model parsimony.
- Extremely low P-value → model is highly significant

The model is statistically significant.

Summary Report

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	yr2012	2036.	62.8	32.4	1.47e-130
2	weathersitLight Snow/Rain	-2031.	216.	-9.42	1.22e- 19
3	temp	4056.	438.	9.26	4.52e- 19
4	seasonWinter	1567.	196.	8.01	6.51e- 15
5	windspeed	-3114.	452.	-6.89	1.52e- 11
6	(Intercept)	1578.	251.	6.29	6.59e- 10
7	weathersitMist	-467.	82.5	-5.66	2.48e- 8
8	seasonSummer	1060.	197.	5.37	1.15e- 7
9	hum	-1376.	312.	-4.42	1.21e- 5
10	seasonFall	962.	229.	4.20	3.05e- 5
11	mnthSeptember	1066.	285.	3.75	1.97e- 4
12	mnthMarch	639.	178.	3.59	3.60e- 4
13	weekdayThursday	353.	114.	3.10	2.03e- 3
14	weekdaySaturday	348.	115.	3.02	2.68e- 3
15	mnthOctober	750.	261.	2.87	4.23e- 3
16	weekdayFriday	302.	114.	2.64	8.47e- 3
17	weekdayWednesday	296.	115.	2.58	1.02e- 2
18	weekdayTuesday	302.	117.	2.57	1.03e- 2
19	mnthMay	670.	291.	2.30	2.15e- 2
20	mnthJune	602.	307.	1.96	5.06e- 2
21	mnthAugust	606.	327.	1.86	6.41e- 2
22	holidayYes	-371.	203.	-1.83	6.85e- 2
23	mnthFeburary	275.	158.	1.75	8.15e- 2
24	mnthApril	429.	273.	1.58	1.16e- 1
25	weekdayMonday	144.	118.	1.22	2.23e- 1
26	mnthJuly	172.	338.	0.509	6.11e- 1
27	mnthDecember	49.6	196.	0.253	8.00e- 1
28	mnthNovember	24.8	249.	0.0996	9.21e- 1

- Significant indicators (low p-value, high t-statistic) are mostly weather-related predictors: **weathersit, temp, hum, windspeed** and **season**, **year**.

Check Assumptions - VIF

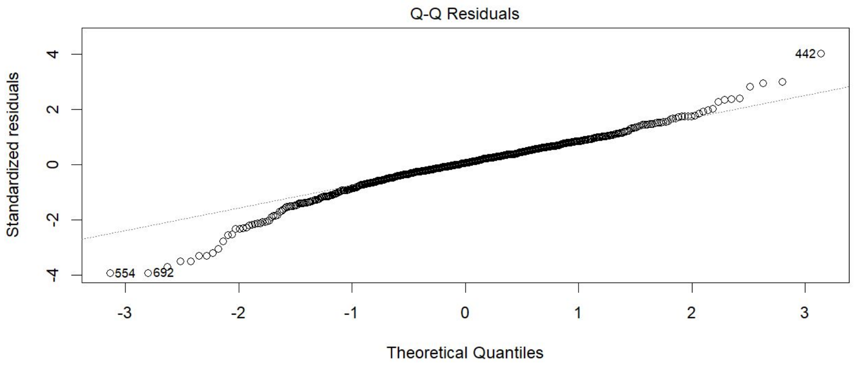
- Severe multicollinearity (vif >> 5) for month, due to high correlation with season (also vif >> 5) - we saw in Cramer’s V matrix.
- Remove **mnth**

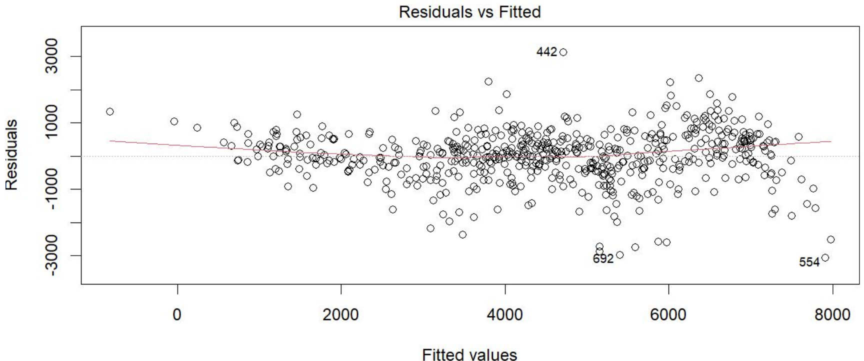
	GVIF	Df	GVIF^(1/(2*Df))
season	171.883799	3	2.357975
yr	1.051922	1	1.025632
mnth	420.546556	11	1.316025
holiday	1.172539	1	1.082838
weekday	1.252433	6	1.018934
weathersit	1.886827	2	1.172015
temp	6.931377	1	2.632751
hum	2.120935	1	1.456343
windspeed	1.230240	1	1.109162

	GVIF	Df	GVIF^(1/(2*Df))
season	3.366547	3	1.224233
yr	1.026183	1	1.013007
holiday	1.142812	1	1.069024
weekday	1.199153	6	1.015250
weathersit	1.708654	2	1.143309
temp	3.337254	1	1.826815
hum	1.727437	1	1.314320

Normality of Errors - QQ Plot + Heteroscedasticity

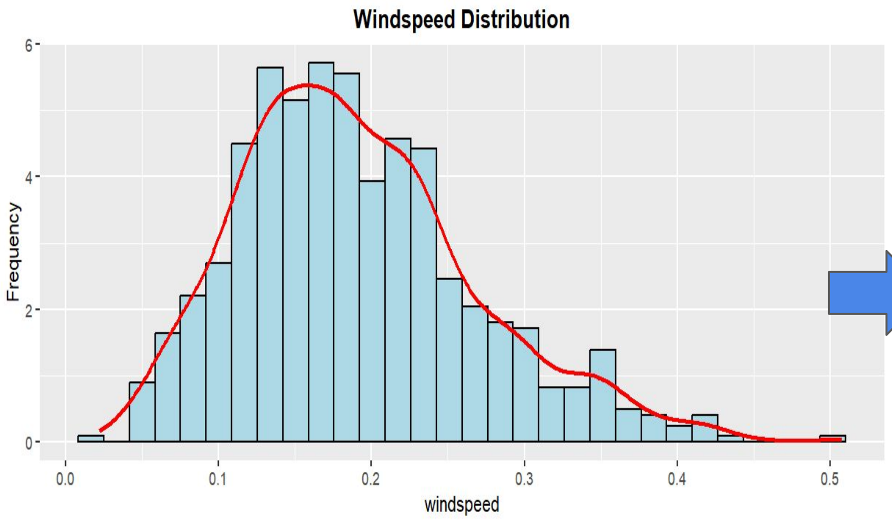
Residuals mostly normally distributed, lower end is in moderate offset → try transformations.

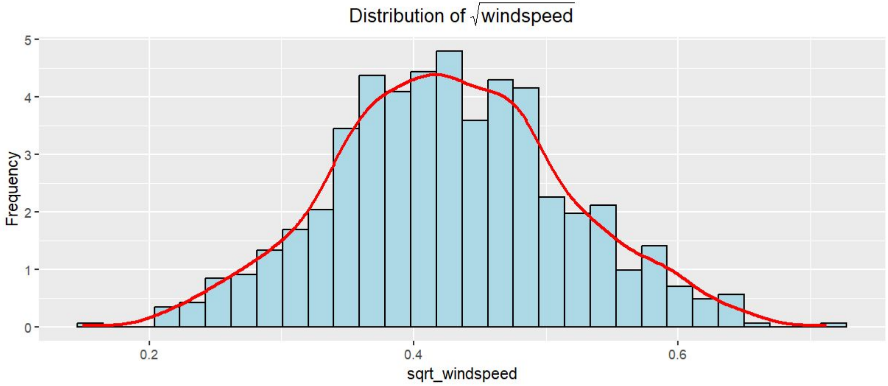




Curvature with high values → try log(cnt). Otherwise, variance remains approx. constant

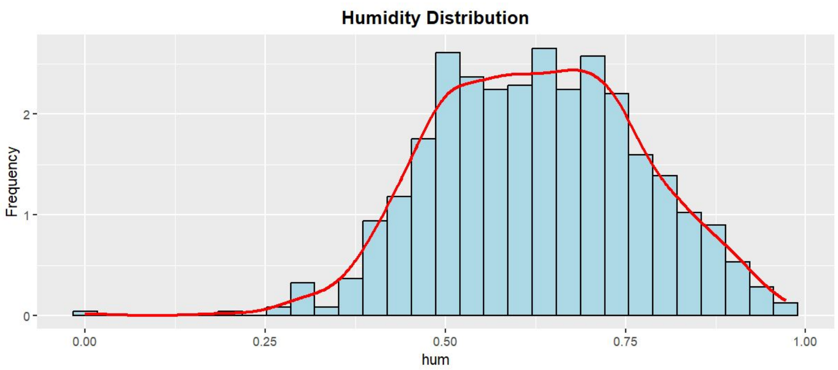
Windspeed Transform

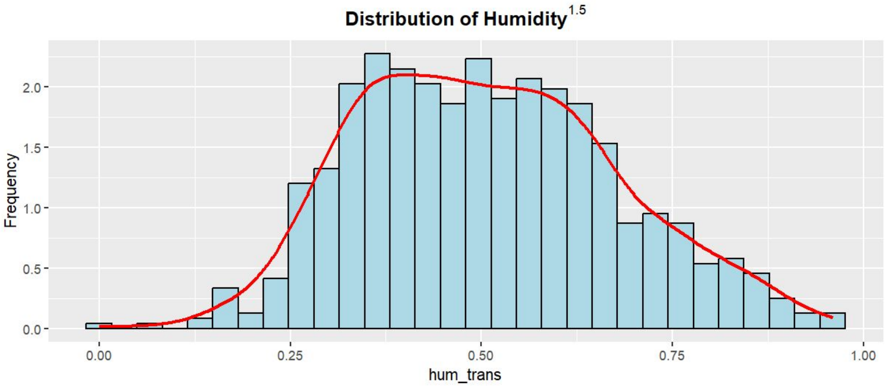




Skewed wind speed distribution → transform using root square.

Humidity Transform

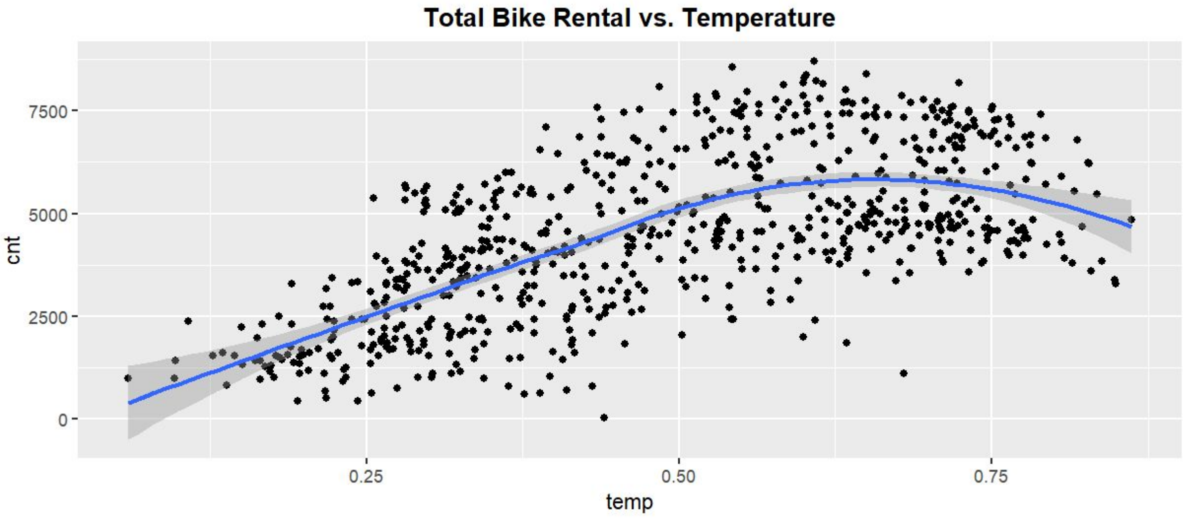




Skewed humidity distribution → transform using power of 1.5

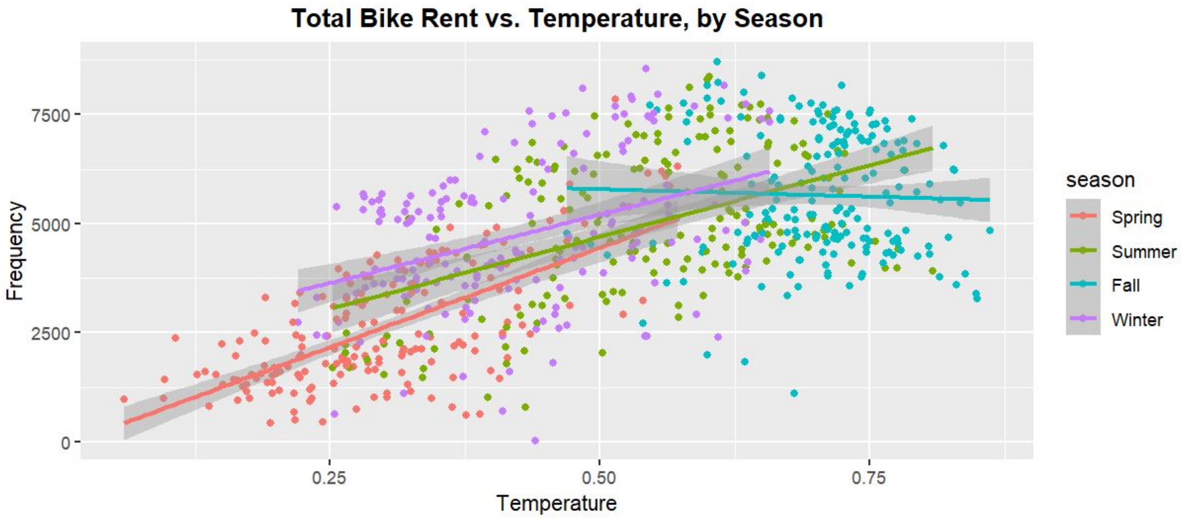
Temperature Non-linearity

- Non-linearity at the edges of the distribution.
- Consider adding polynomial term of second order to account for it.



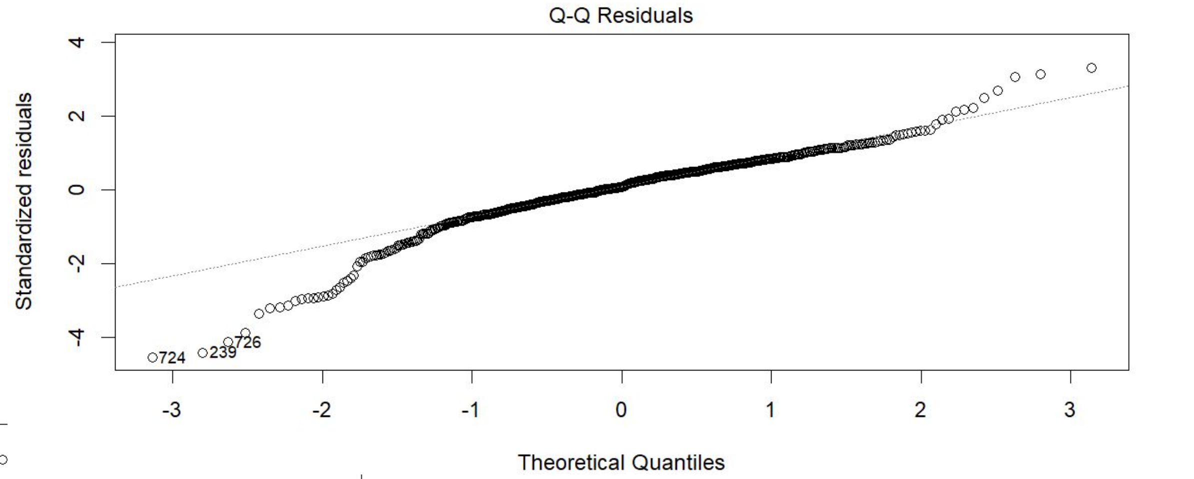
Interaction term season:temp

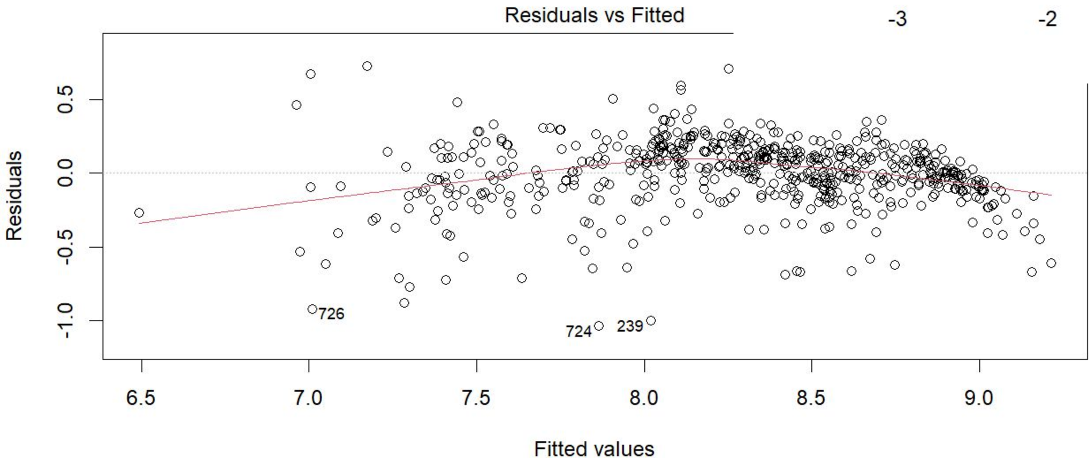
- High influence of temperature in most seasons.
- Suggests to include an interaction term between season and temperature



Normality of Errors - QQ Plot + Heteroscedasticitv

QQ plot didn’t improve





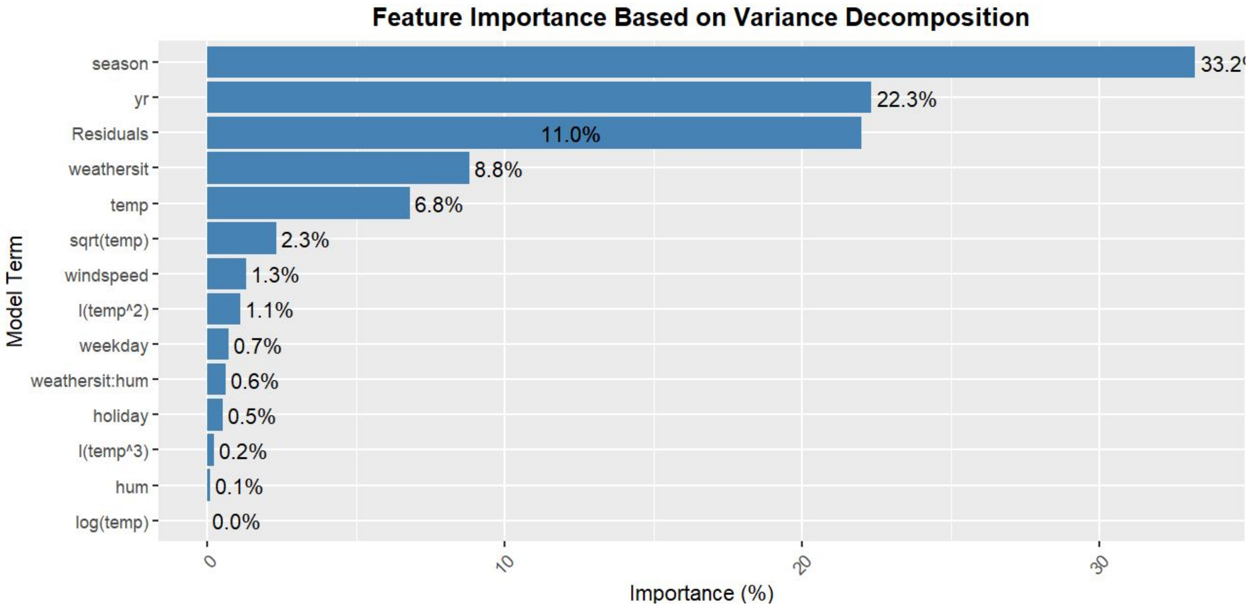
Still has curvature, but the scale is far less dominant.

Model Performance

- Test set constitutes 20% of dataset
- Measuring the following on test set:

	base_model	new
RMSE	815.8253	724.8601
MAE	605.6976	535.5766
MAPE	19.3582	16.1223
R_squared	0.8228	0.8601
Adjusted_R_squared	0.8004	0.8346

Feature Importance



- Most informative features: Season, yr, weathersit, temperature
- 11 % unexplained variance.

Conclusions

- Overall growth in demand from 2011 to 2012. This data must be collected and expanded.
- Generally, higher temperatures increase bike rentals, but extreme behaviour decreases it.
- Higher humidity negatively impacts bike rental demand.
- Higher wind speeds reduce bike rental demand.
- Light snow or rain significantly reduces rentals.
- There’s still some unexplained variance, thus more optimization is required.

High Casual Summer Demand Criteria

1. Class balance - A value that can maintain enough positive cases for the model to learn from.
2. Capture trends and not outliers. Outliers might negatively affect performance.
3. Business perspective:
   a. Reduce False positives, because this costs money e.g. additional staff scheduling and maintenance availability.
   b. If correctly predicted, the revenue is highest during these days
   c. Make sure enough bikes are available

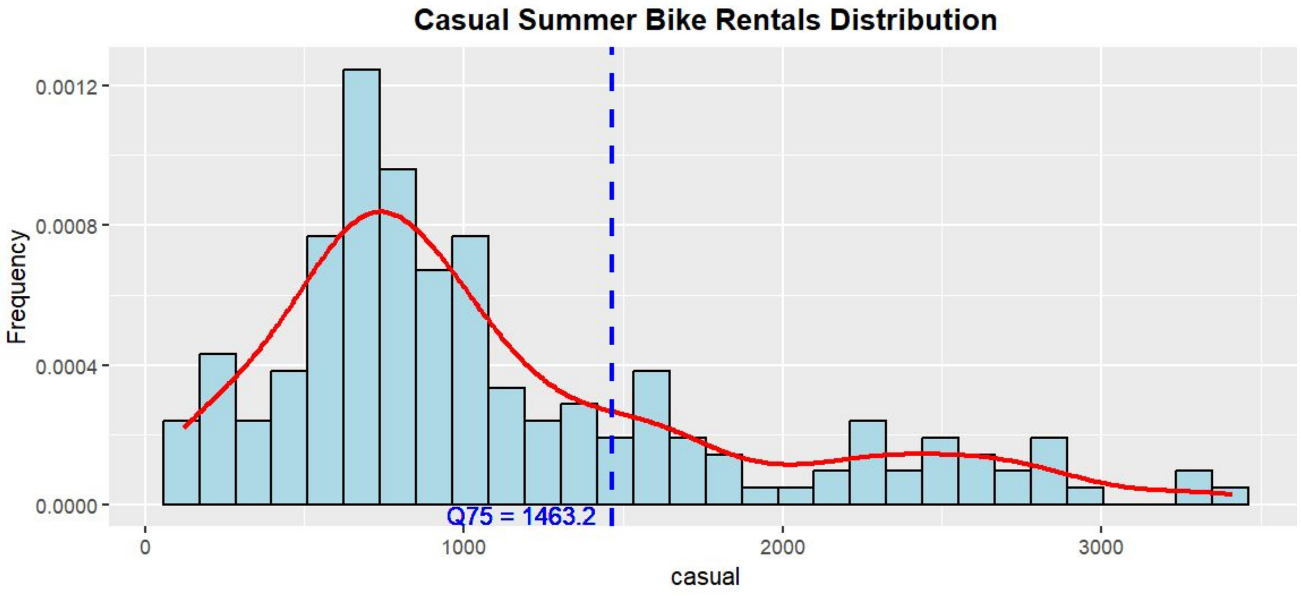
Based on the following plots,
I would select <u>75 percentile</u> as an adequate threshold.

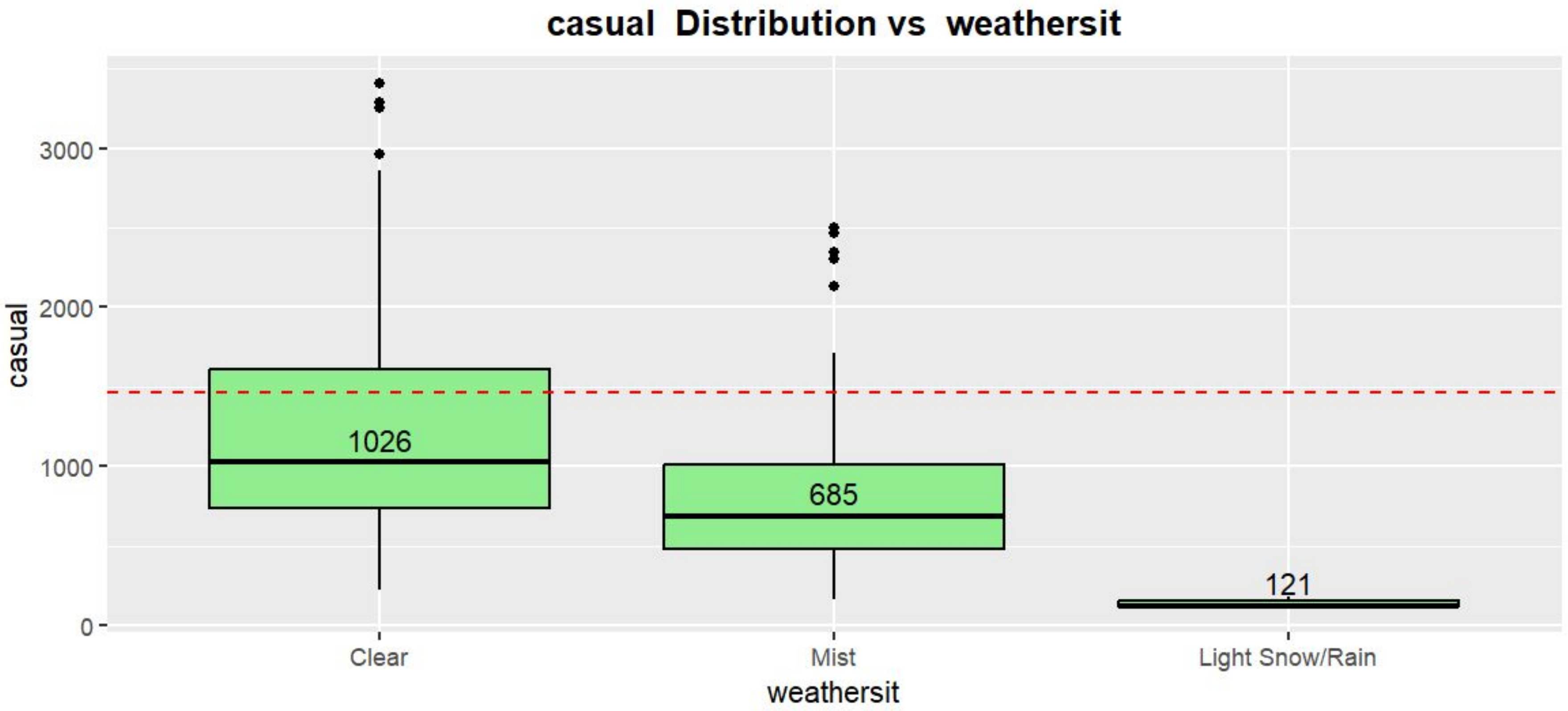
```
 table(df_sum$high_demand)

low high
138   46
```

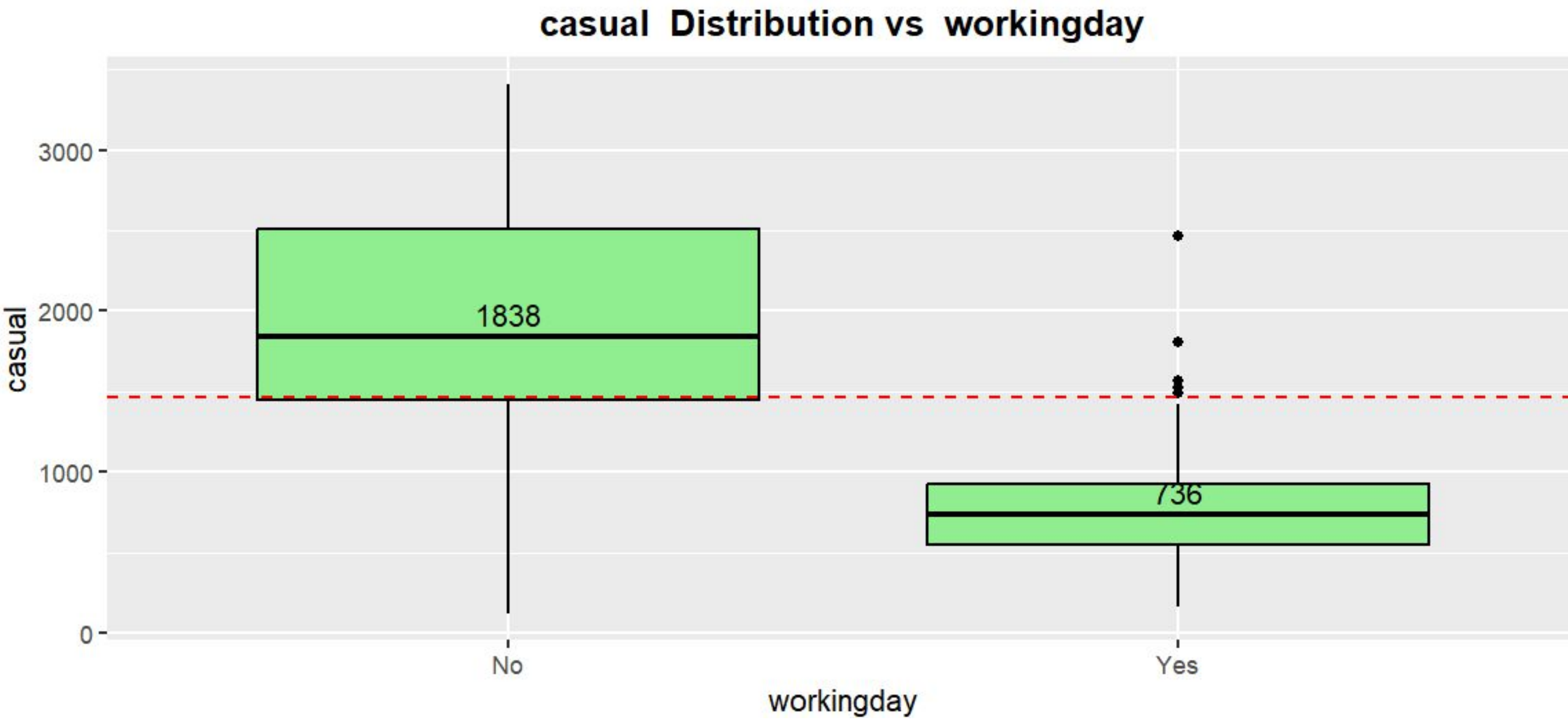
→**25**% high demand data, as expected.

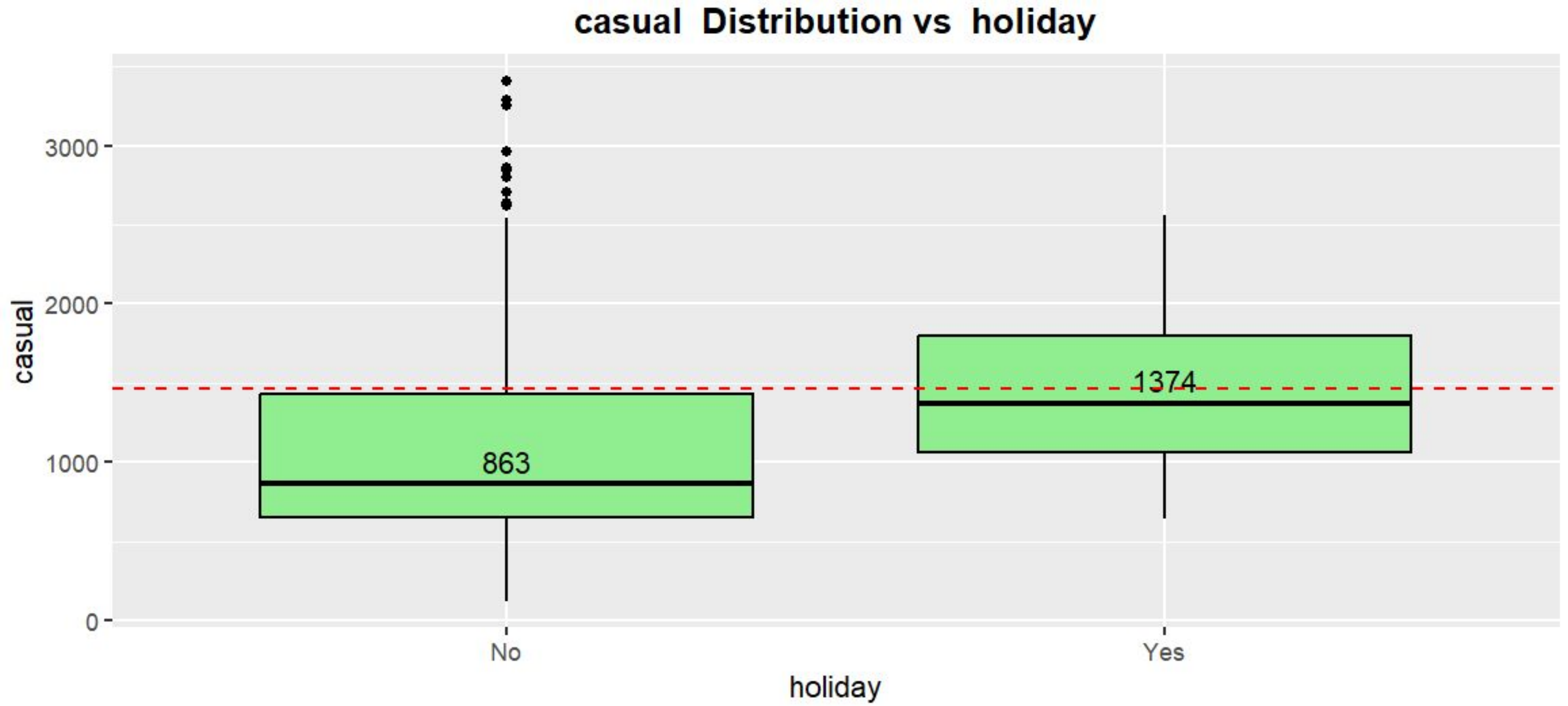
High Casual Bike Rental Demand

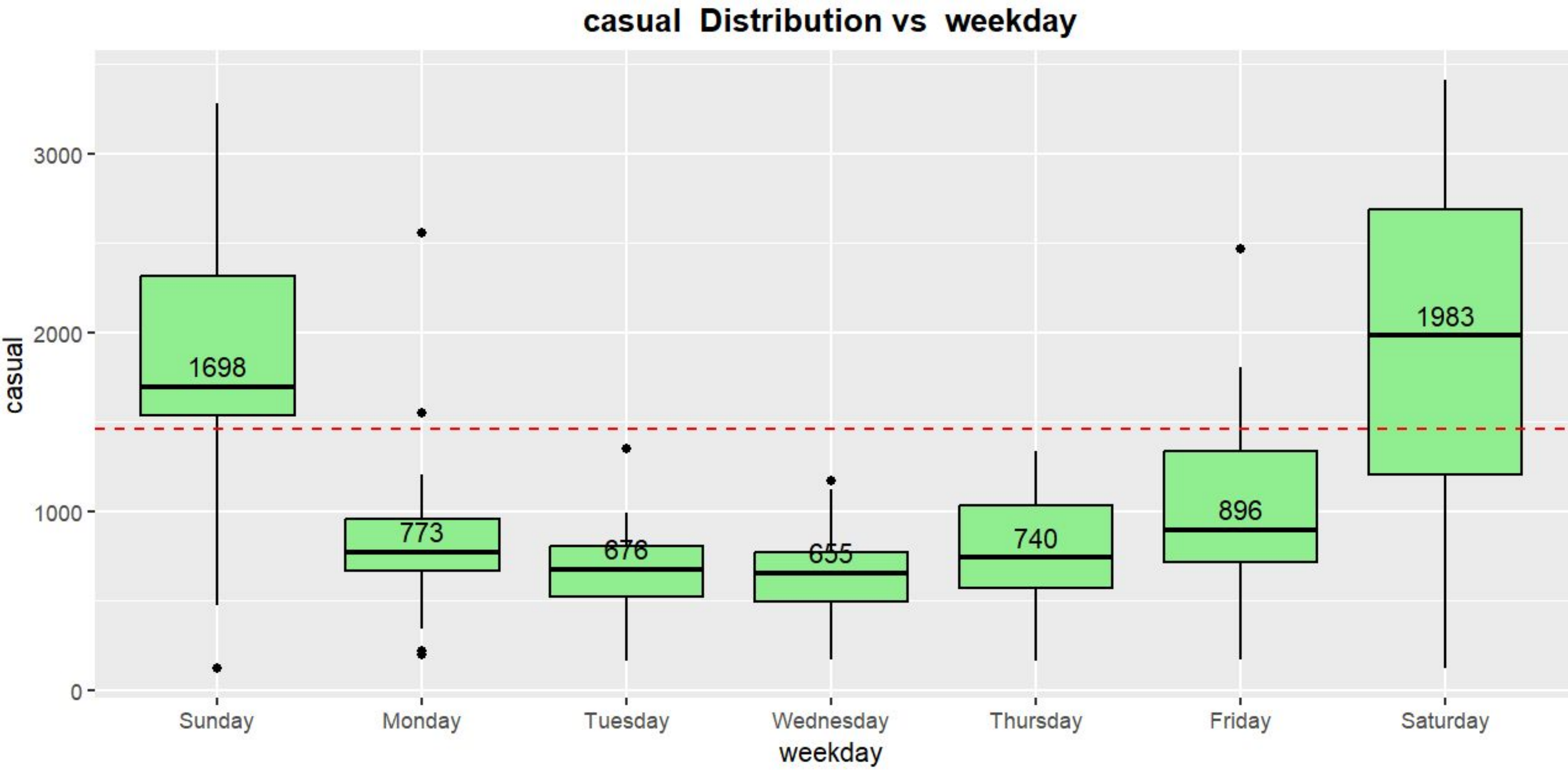




- 75 percentile


casual Distribution vs workingday
casual
3000
2000
1000
0
1838
736
No
Yes
workingday


casual Distribution vs holiday
casual
3000
2000
1000
0
863
1374
No
Yes
holiday



Q3 captures adequately weekend trends.

Logistic Regression Model

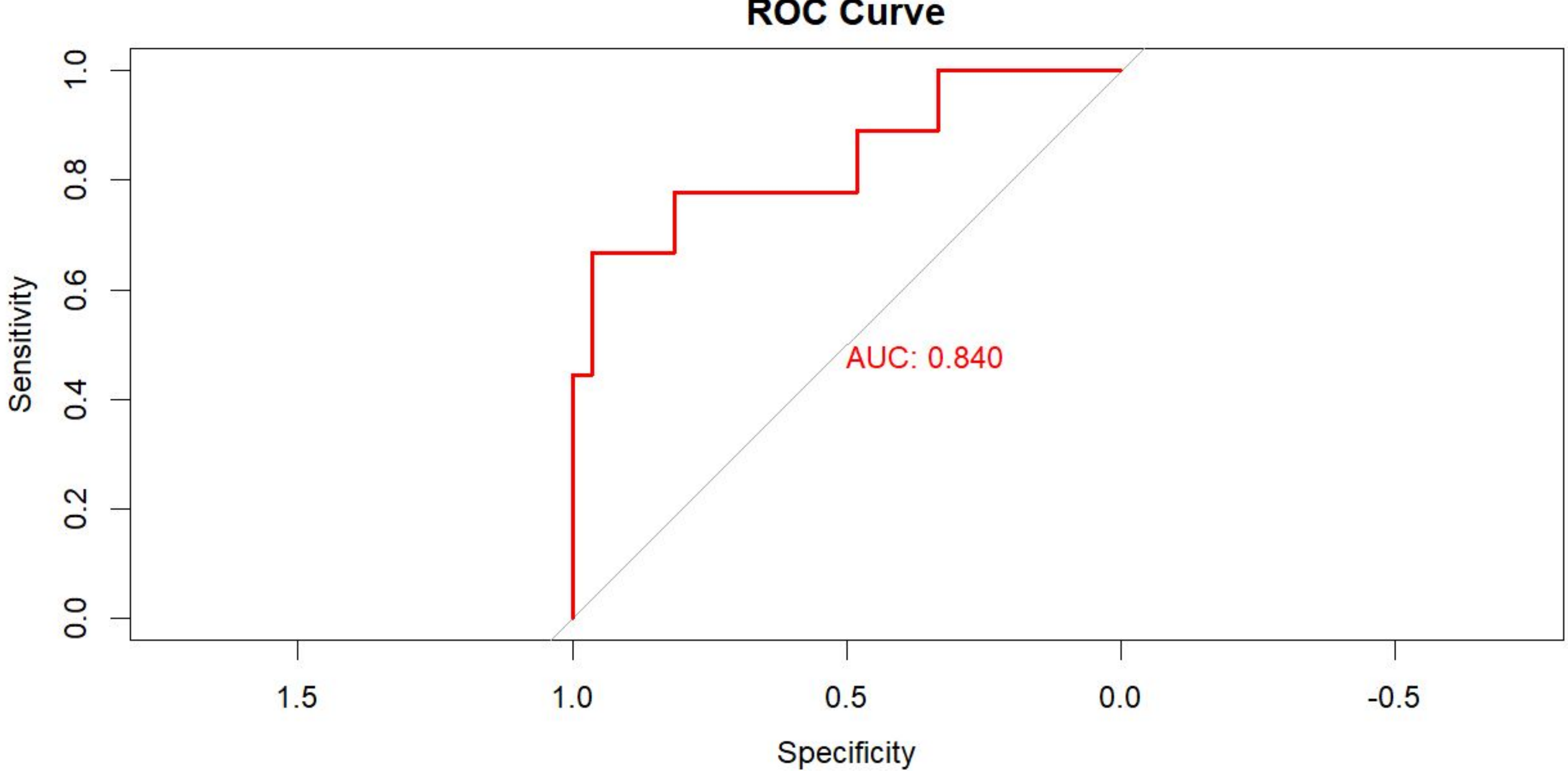
- Convert weathersit into binary, if weather is “Clear” or not.
- Add “is_weekend”, binary signallig whether it’s Saturday or Sunday.
- Statistically significant features: Year, Holiday, is_weekend, temp.
- Let’s simplify the model by removing non-signifcant predictors.

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	-13.3	5.43	-2.46	0.0140
2	yr2012	3.94	1.71	2.30	0.0214
3	mnthApril	1.19	1.67	0.716	0.474
4	mnthMay	1.14	2.12	0.538	0.591
5	mnthJune	1.93	2.54	0.761	0.447
6	holidayYes	8.04	3.72	2.16	0.0309
7	weathersit1	0.955	1.57	0.610	0.542
8	temp	17.3	7.49	2.30	0.0213
9	hum	-8.57	5.56	-1.54	0.123
10	windspeed	-6.61	8.76	-0.755	0.450
11	is_weekendTRUE	10.6	2.76	3.86	0.000113

Logistic Regression Model - Simplified

All remaining predictors
are statistically significant.

	term	estimate	std.error	statistic	p.value
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	-17.7	4.43	-4.00	0.0000641
2	yr2012	2.71	1.10	2.45	0.0141
3	holidayYes	5.78	2.67	2.16	0.0305
4	is_weekendTRUE	8.98	1.97	4.55	0.00000528
5	temp	15.8	5.20	3.03	0.00246
6	weathersit1	2.67	1.29	2.07	0.0387



Relatively high AUC on test data.

ROC curve above diagonal line, meaning that it’s better than just guessing

Odds Ratio Analysis

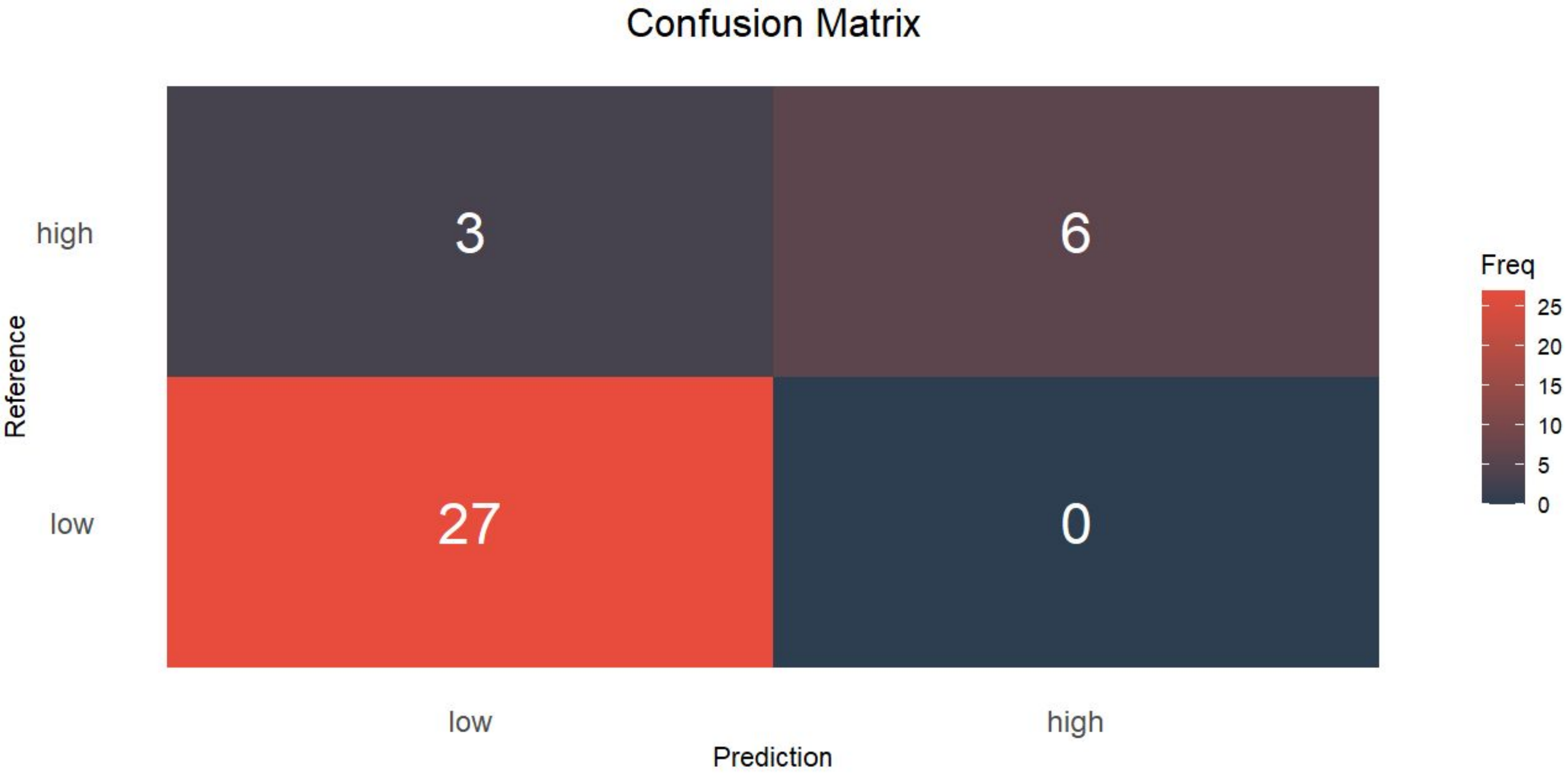
```
exp(coef(model2)) - 1
 (Intercept)          yr2012      holidayYes is_weekendTRUE               temp     weathersit1
    -1.00000        14.05649       322.07222     7935.90567   7001030.92728        13.48991
```

- 2012 increases demand by massive amount. Could reflect better service. This reflects a strong temporal effect. Collecting data for more years is crucial.
- Holiday increases strongly the likelihood of high demand, likely because casual users rent bikes for leisure activities.
- Is_weekend is by far the strongest predictor. Casual users primarily use bike-sharing services on weekends for recreation.
- Temp is highly predictive of high demand, but requires calibration.
- Clean weather an important factor, though its effect is smaller compared to others.

Conclusion: Focus operational resources (e.g., bike availability) on weekends, holidays, and during favorable weather.

Metrics considerations

1. On the one side, we would like to reduce FP: reduce days of anticipated high demand where it’s not.
2. On the other hand, we would like to reduce FN: minimize days when there’s actual high demand, but predicted not, thus not ready to provide elevated demand -> missing customers, revenue and prestige.
3. Assume then that both are equally important, so we would like to maximize F1.



FN = 0 and FP>0 on the test data, but overall seems to be quite good.

Performance Metrics on test data

```
[1] "Model Performance Metrics:"
> print(paste("F1 Score:", round(f1, 3)))
[1] "F1 Score: 0.947"
> print(paste("Accuracy:", round(conf_matrix$overall['Accuracy'], 3)))
[1] "Accuracy: 0.917"
> print(paste("Precision:", round(precision, 3)))
[1] "Precision: 0.9"
> print(paste("Recall:", round(recall, 3)))
[1] "Recall: 1"
```

Very high F1, Precision and Recall.

It seems that the model is quite good at generalizing to new data, and it achieves our desired high F1.

If business wishes to maximize Precision instead, more optimization is required.