Assessing Multiple Regression Models

why R^2 alone is not enough

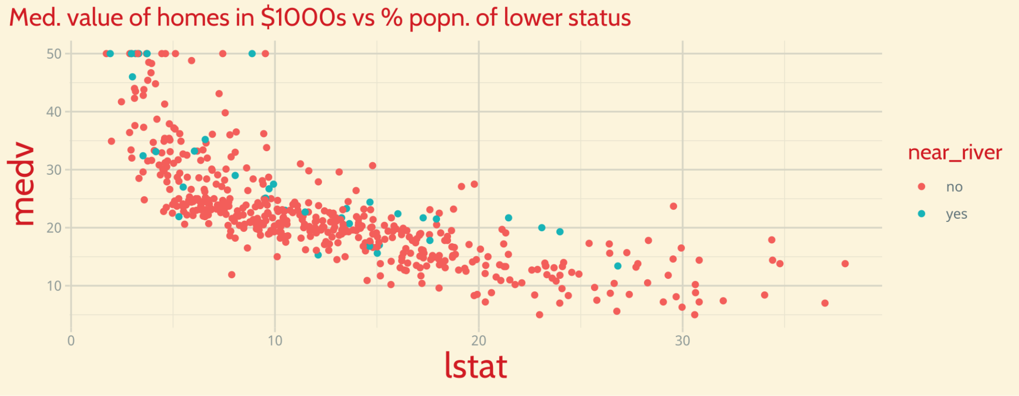
Vasant Marur

2021-06-08

Simple Linear Regression

Example from ISLR [1] using the `Boston` data set from `MASS` library

```r
Boston %>%
  mutate(near_river =
           ifelse(chas==1,"yes",
                  "no")) %>%
ggplot(.) +
  aes(lstat, medv,
      color = near_river) +
  geom_point() +
   theme_xaringan(text_font_size = 11) +
  scale_color_discrete() +
labs(title = "Med. value of homes in $1000s vs
   theme(plot.title = element_text(size=16),
         legend.title  = element_text(size=14)
```



[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013.

r^2 (Correlation^2) == R^2 ?

As a refresher *correlation* denoted by r is defined as

$$r = Cor(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

,is also a measure of linear relationship between X and Y. This means we might be able to use $r = Cor(X, Y)$ instead of R^2 to assess the fit of the linear model. For simple linear regression setting it can be shown $R^2 = r^2$, which is to say squared correlation and the R^2 statistic are identical.

The r for `medv,lstat` variables from the `Boston` data set is -0.7376627 and r^2 is 0.5441463.

Note how the values of r^2 are equivalent to values of R^2.

Meanwhile, the R^2 value from a simple regression model given by

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

which expressed in variable terms is

$$medv = \beta_0 + \beta_1 lstat + \epsilon$$

and the R^2 is 0.5441463. The R^2 is same even if we switch the X, Y variables. The R^2 for

$$lstat = \beta_0 + \beta_1 medv + \epsilon$$

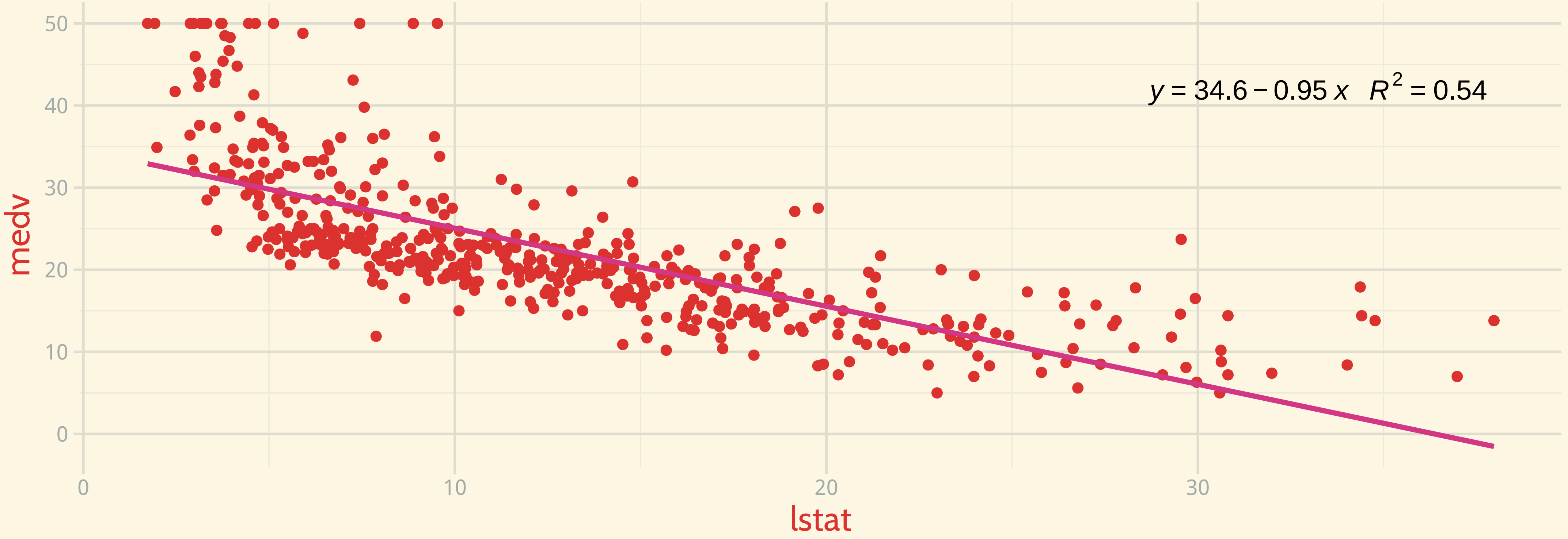
is 0.5441463.

These models are visualized on the next two slides.

Now with linear model fitted

$$medv = \beta_0 + \beta_1 lstat + \epsilon$$

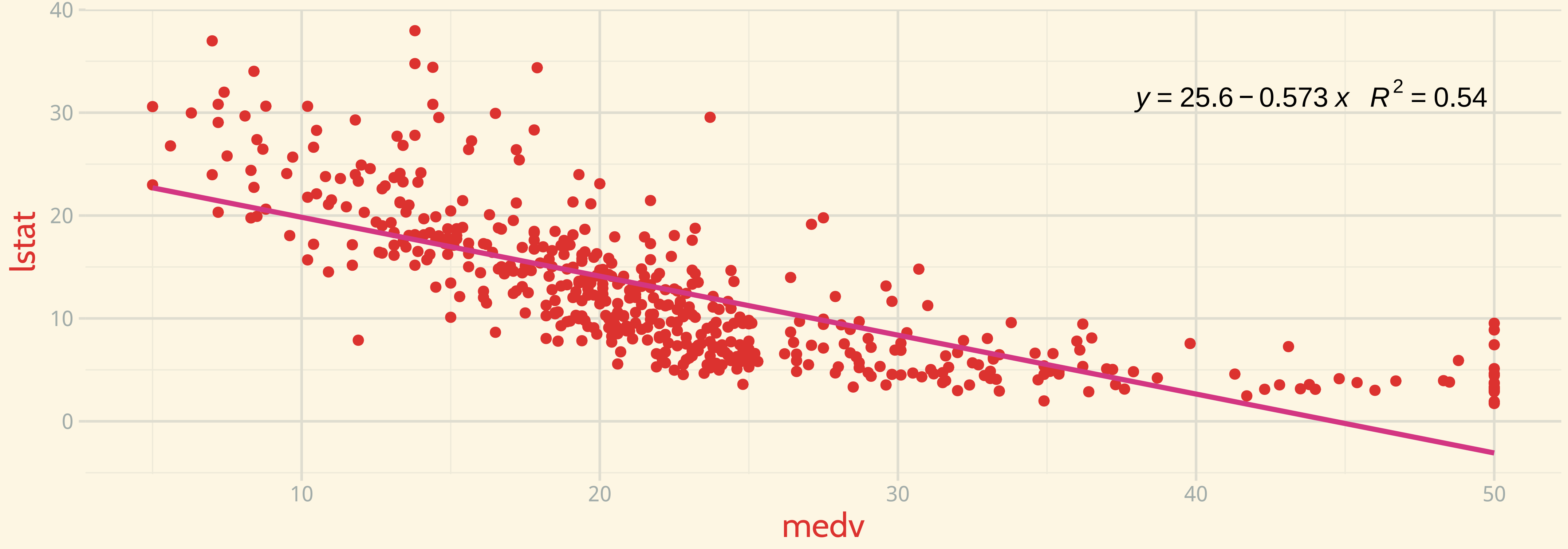
Med. value of homes in $1000s vs % popn. of lower status



Now with linear model fitted by switching Y & X

$$lstat = \beta_0 + \beta_1 medv + \epsilon$$

% popn. of lower status vs Med. value of homes in $1000s



How do the model stats look?

Model Stats for $medv = \beta_0 + \beta_1 lstat + \epsilon$

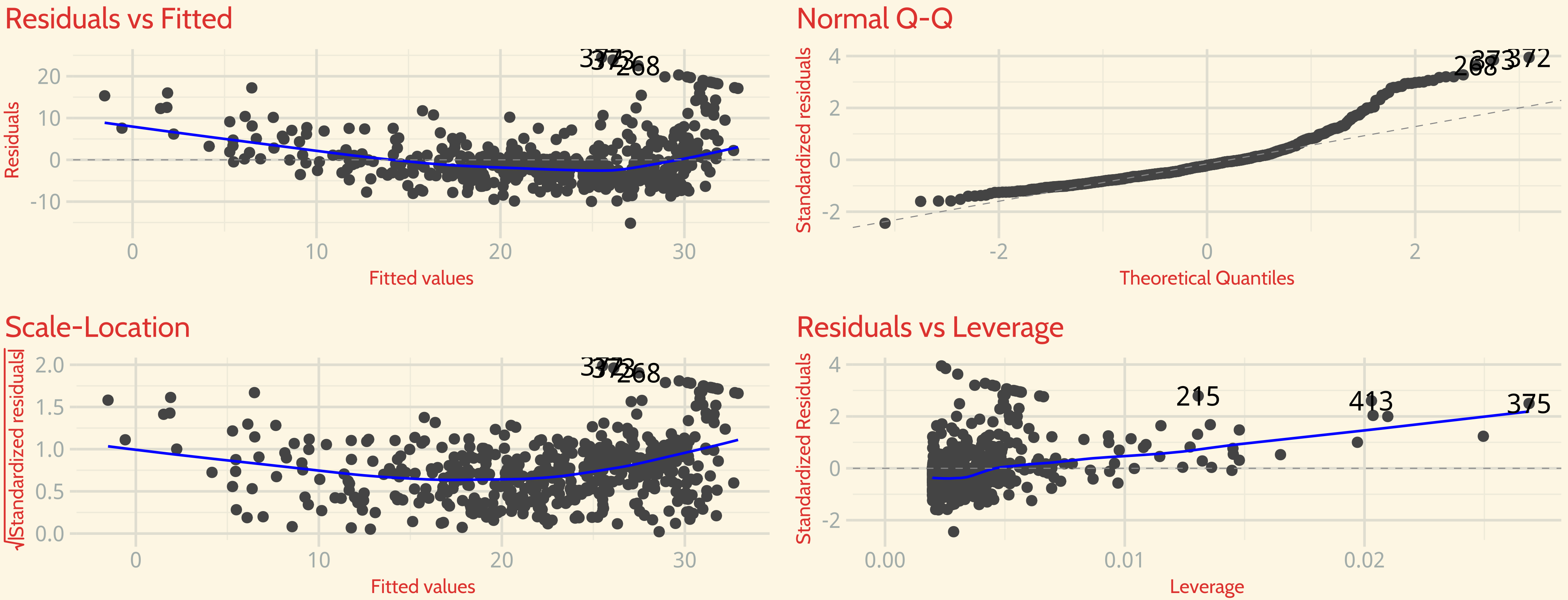
term	estimate	std.error	statistic	p.value
(Intercept)	34.55	0.5626	61.42	3.743e-236
lstat	-0.95	0.03873	-24.53	5.081e-88

the Model metrics

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.5441	0.5432	6.216	601.6	5.081e-88	1	-1641	3289	3302	19470	504	506

how do the model diagnostic plots look?

For $medv = \beta_0 + \beta_1 lstat + \epsilon$



How do the model stats look?

Model Stats for $lstat = \beta_0 + \beta_1 medv + \epsilon$

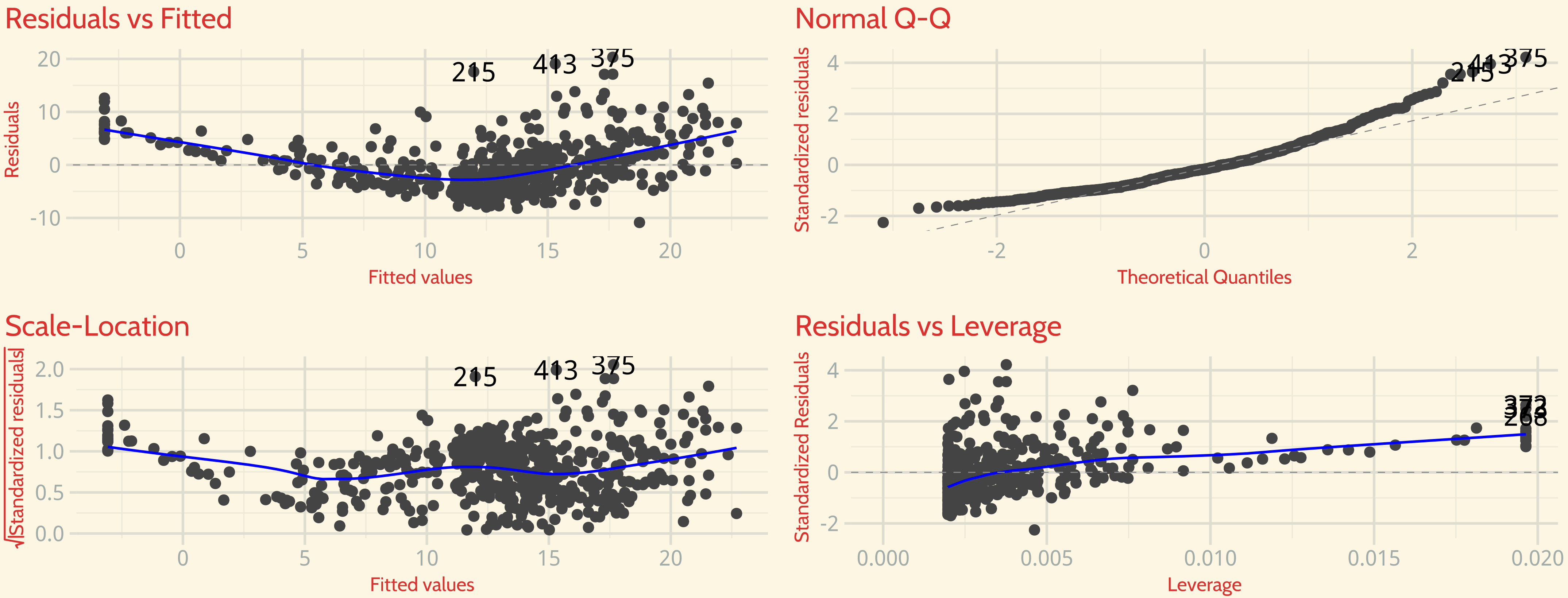
term	estimate	std.error	statistic	p.value
(Intercept)	25.56	0.5682	44.98	1.402e-178
medv	-0.5728	0.02335	-24.53	5.081e-88

the Model metrics

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.5441	0.5432	4.826	601.6	5.081e-88	1	-1513	3033	3046	11740	504	506

how do the model diagnostic plots look?

For $lstat = \beta_0 + \beta_1 medv + \epsilon$



Multiple Linear Regression

Generating some random data

```
set.seed(42)
y <-rnorm(1000, mean = 3, sd = 5)
random_data <- data.frame(y)
random_data <- random_data %>%
  mutate(x = rnorm_pre(y, mu = 6,
                       sd = 2, r = 0.8))
random_data <- random_data %>%
  mutate(x1 = rnorm_pre(y, mu = 6.1,
                        sd = 5, r = 0.97))
random_data <- random_data %>%
  mutate(x2 = rnorm_pre(y, mu = 6,
                        sd = 2, r = 0.67))
#
#
#
random_data <- random_data %>%
  mutate(x6 = rnorm_pre(y, mu = 5,
                        sd = 6, r = 0.4))
```

The generated data

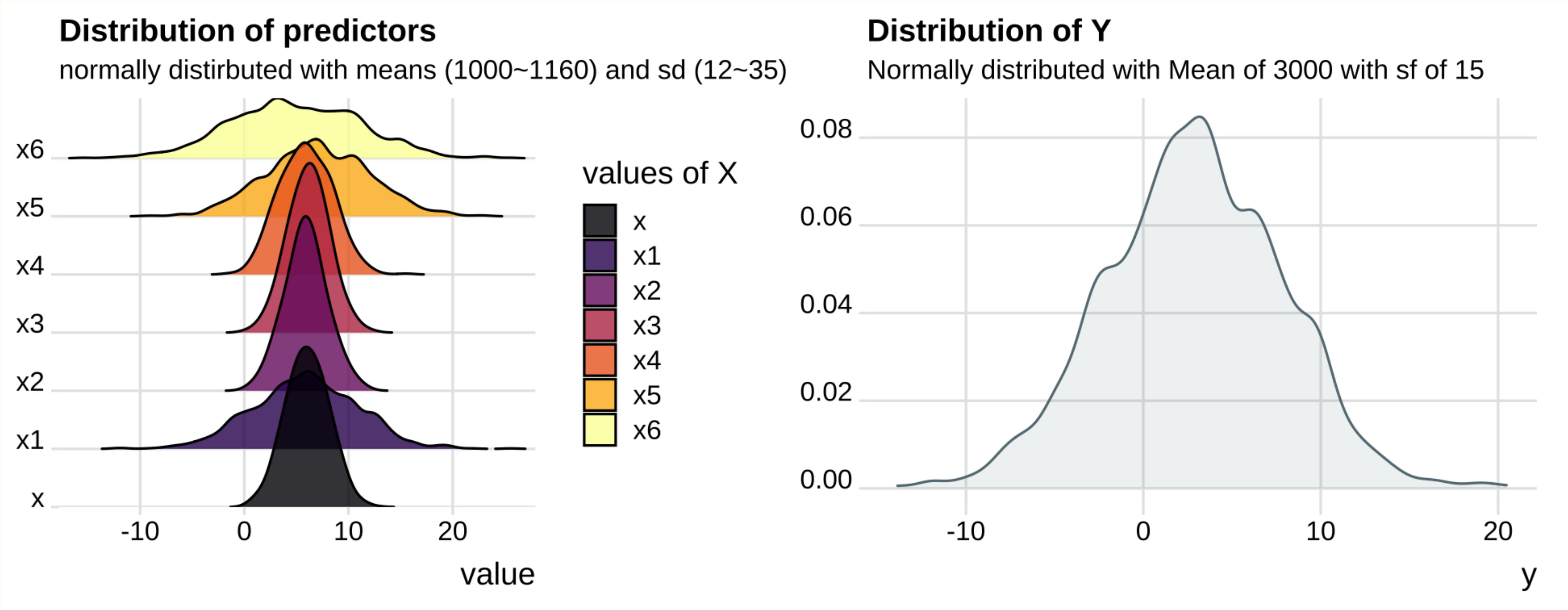
```
head(random_data,n = 20) %>%
              dplyr::mutate_if(is.numeric,
                   ~ as.character(round(.,2))) %>%
  DT::datatable(.,fillContainer = FALSE,
                options = list(pageLength = 4))
```

Show 4 entries Search:

	y	x	x1	x2	x3	x4	x5	x6
1	9.85	8.81	12.51	7.07	7.27	7.63	11.96	2.03
2	0.18	3.98	2.34	5.36	5.16	3.57	9.16	0.6
3	4.82	6	6.81	7.64	7.25	6.74	10.16	0.09
4	6.16	7.34	9.81	5.54	7.8	5.84	11.65	7.08

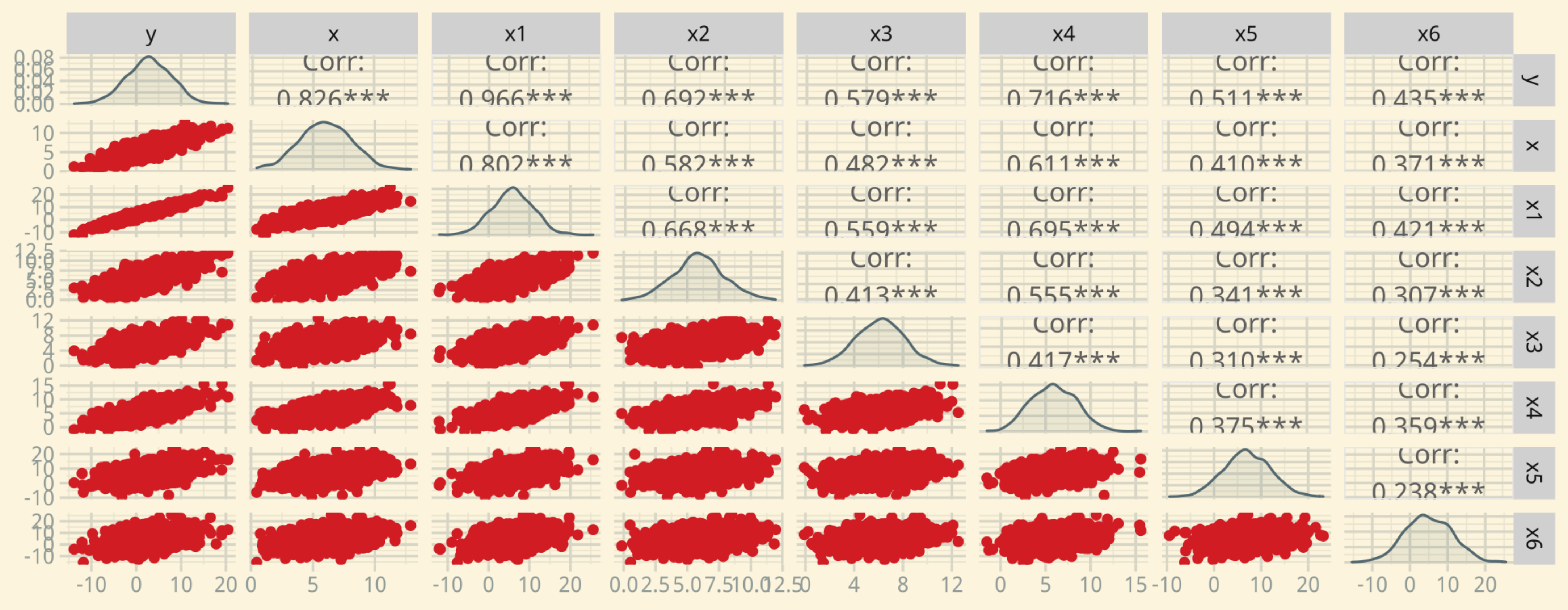
Showing 1 to 4 of 20 entries Previous 1 2 3 4 5 Next

Let's visualize the predictors and response variable



How do they correlate with each other and y

```r
ggpairs(random_data) +
  theme_xaringan(title_font_size = 10,
                 text_font_size = 11)
```



The multiple regression model

We're going to fit a multiple linear regression model, which takes the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

We will start with one predictor to get a baseline and add more as we go along and look at the metrics.

```
lm_random <- lm ( y ~ x, data = random_data)
lm_random_tidy <-tidy(lm_random)

lm_random_tidy %>%
  dplyr::mutate_if(is.numeric,
                   ~ as.character(signif(.,4))) %>%
  knitr::kable(., format = 'html')
```

term	estimate	std.error	statistic	p.value
(Intercept)	-9.361	0.2784	-33.62	2.727e-166
x	2.027	0.04372	46.38	3.01e-251

Goodness of fit Measures

```
glance(lm_random) %>% dplyr::mutate_if(is.numeric,
                                       ~ as.character(signif(.,4))) %>%
    knitr::kable(., format = 'html')
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.6831	0.6827	2.823	2151	3.01e-251	1	-2456	4918	4932	7956	998	1000

Adding more variables a.k.a stepwise

> Yes stepwise is not recommended, and more appropriate methods such as shrinkage a.k.a regularization are better [2]
>
> We're doing this to see the effect on R^2, AIC, BIC as we add more variables.

[2] Smith, G. Step away from stepwise. *J Big Data* 5, 32 (2018). https://doi.org/10.1186/s40537-018-0143-6 https://rdcu.be/clWQm

Adding one more variable

We're going to fit a multiple linear regression model, with two variables, which takes the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

term	estimate	std.error	statistic	p.value
(Intercept)	-4.247	0.143	-29.71	1.951e-139
x	0.3531	0.0316	11.17	2.173e-27
x1	0.846	0.01281	66.04	0

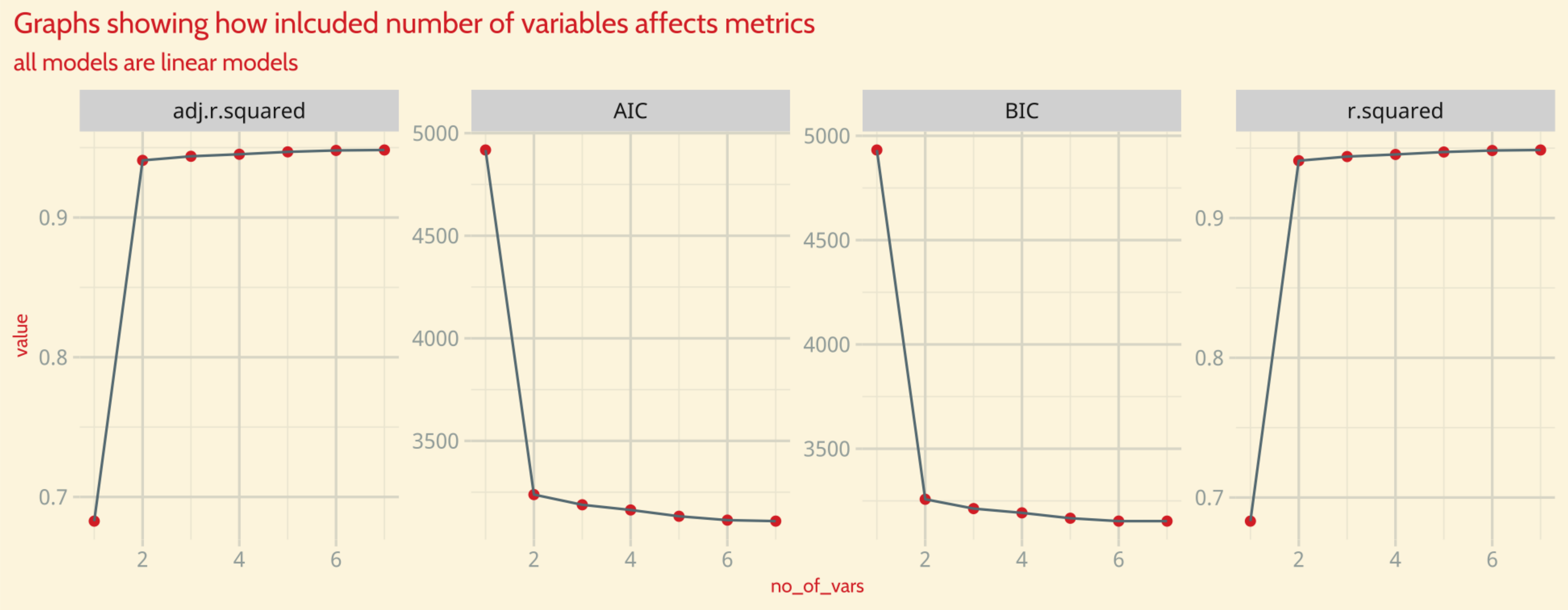
Goodness of fit Measures

```
glance(lm_random1) %>% dplyr::mutate_if(is.numeric,
                                       ~ as.character(signif(.,4))) %>%
    knitr::kable(., format = 'html')
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.941	0.9409	1.219	7954	0	2	-1615	3238	3258	1480	997	1000

Comparing model metrics

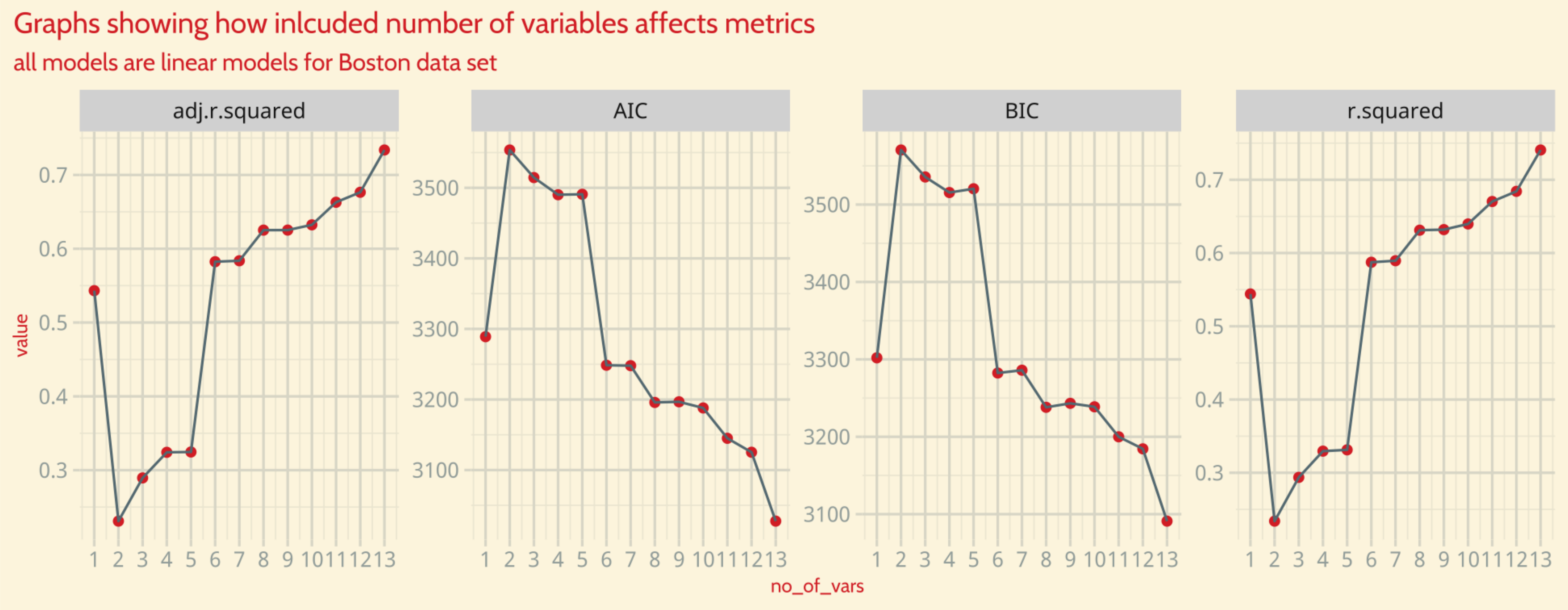
no_of_vars	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
1	0.6831	0.6827	2.823	2151	3.01e-251	1	-2456	4918	4932	7956	998	1000
2	0.94102263	0.94090432	1.2185446	7953.895	0	2	-1615.0935	3238.187	3257.818	1480.3965	997	1000
3	0.94399021	0.94382151	1.188088	5595.5354	0	3	-1589.2799	3188.5597	3213.0985	1405.907	996	1000
4	0.94549963	0.94528053	1.1725585	4315.4392	0	4	-1575.6204	3163.2407	3192.6873	1368.0189	995	1000
5	0.94725264	0.94698731	1.1541267	3570.1091	0	5	-1559.2735	3132.5469	3166.9012	1324.0164	994	1000
6	0.94833871	0.94802656	1.1427581	3038.0593	0	6	-1548.871	3113.742	3153.004	1296.7549	993	1000
7	0.94869284	0.94833079	1.1394086	2620.3617	0	7	-1545.4318	3108.8636	3153.0334	1287.866	992	1000



The metrics don't change much after 3 or more variables, of course this will change for real data which are not as correlated. but it's pretty evident from this graph and the table on previous slide that R^2 does increase as we add more variables to the model.

Let's see how this approach works on the Boston data set

no_of_vars	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
1	0.5441	0.5432	6.216	601.6	5.081e-88	1	-1641	3289	3302	19470	504	506
2	0.23398844	0.23094267	8.0654845	76.824026	7.6768693e-30	2	-1772.8009	3553.6018	3570.5079	32721.176	503	506
3	0.29371357	0.28949274	7.7523855	69.5866	1.2110319e-37	3	-1752.2633	3514.5265	3535.6592	30169.94	502	506
4	0.32952772	0.32417465	7.5608103	61.558619	2.6935556e-42	4	-1739.0975	3490.195	3515.5543	28640.092	501	506
5	0.3313127	0.32462583	7.5582861	49.546732	1.1691645e-41	5	-1738.4231	3490.8461	3520.4319	28563.844	500	506
6	0.58737696	0.58241556	5.9432398	118.38938	1.3157599e-92	6	-1616.2792	3248.5584	3282.3707	17625.728	499	506
7	0.58949016	0.58371994	5.9339503	102.1608	4.2005869e-92	7	-1614.9802	3247.9603	3285.9991	17535.46	498	506
8	0.63114876	0.62521152	5.6304644	106.30333	1.5284484e-102	8	-1587.9075	3195.815	3238.0804	15755.959	497	506
9	0.63194785	0.62526949	5.630029	94.626125	9.5525574e-102	9	-1587.3588	3196.7176	3243.2095	15721.824	496	506
10	0.6396628	0.63238326	5.5763335	87.87133	5.3062041e-103	10	-1581.9992	3187.9983	3238.7168	15392.27	495	506
11	0.67031409	0.6629729	5.33929	91.308713	1.8315851e-111	11	-1559.5075	3145.015	3199.96	14082.961	494	506
12	0.68420428	0.67651757	5.2309004	89.011316	4.8951312e-115	12	-1548.6172	3125.2343	3184.4058	13489.623	493	506
13	0.74064266	0.73378973	4.7452982	108.07667	6.7221748e-135	13	-1498.8043	3027.6086	3091.0066	11078.785	492	506



The metrics for Boston data set definitely change a lot. You can see how the r.squared decreases & AIC/BIC both increase(lower value better) when you add more variables initially. After 5 variables all the measures show an improvement. Again it's pretty evident from this graph and the table on previous slide that R^2 does increase as we add more variables to the model.

So only R^2 alone is bad idea, start by looking at residual plots, look and use confidence intervals for slope and intercept instead if trying to learn from the model and SE or prediction intervals if using model for prediction.

Read *Chapter 6 Linear Model Selection and Regularization* of ISLR for more details on C_p, AIC, BIC, R^2
Also, read is R-squared Useless

Thanks!

Slides created via the R packages:

xaringan
gadenbuie/xaringanthemer

The chakra comes from remark.js, **knitr**, and R Markdown.