Homework 2: Simple Linear Regression

Ruonan Zhao

September 04, 2024

Instructions:

The purpose of this homework assignment is to practice simple linear regression and diagnostic evaluation of models. Follow the instructions carefully. You are allowed to write as much code as needed in each code chunk but make sure you complete the task as described and only print the output that is requested. After answering each question, knit your RMarkdown file to PDF.

The Bike Sharing Dataset contains daily counts of rental bikes from a two-year period (2011-2012) in Washington D.C. We will be exploring relationships between features like temperature, windspeed, and bike counts. The dataset is available through UC Irvine's Machine Learning Repository. You can use the read_csv() function to import the dataset from "day.csv" provided for you.

After you finish the assignment, knit the RMarkdown file to PDF and submit the PDF to Gradescope under Homework 2.

Questions

Q1 (2 Points)

Load the **Bike Sharing Dataset** from the provided csv file using `read_csv()`. Then, use the `str()` function to preview the dataset.

```
bike_sharing = read_csv("day.csv")
```

```
## Rows: 730 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): dteday
## dbl (15): instant, season, yr, mnth, holiday, weekday, workingday, weathersi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(bike_sharing, )
```

```
## spc_tbl_ [730 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ instant   : num [1:730] 1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : chr [1:730] "1/1/11" "1/2/11" "1/3/11" "1/4/11" ...
##  $ season    : num [1:730] 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ yr        : num [1:730] 0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : num [1:730] 1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday   : num [1:730] 0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : num [1:730] 6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: num [1:730] 0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: num [1:730] 2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num [1:730] 0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num [1:730] 0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num [1:730] 0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num [1:730] 0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : num [1:730] 331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: num [1:730] 654 670 1229 1454 1518 ...
##  $ cnt       : num [1:730] 985 801 1349 1562 1600 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   instant = col_double(),
##   ..   dteday = col_character(),
##   ..   season = col_double(),
##   ..   yr = col_double(),
##   ..   mnth = col_double(),
##   ..   holiday = col_double(),
##   ..   weekday = col_double(),
##   ..   workingday = col_double(),
##   ..   weathersit = col_double(),
##   ..   temp = col_double(),
##   ..   atemp = col_double(),
##   ..   hum = col_double(),
##   ..   windspeed = col_double(),
##   ..   casual = col_double(),
##   ..   registered = col_double(),
##   ..   cnt = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Q2 (3 Points)

Create a new data frame called **BikeData** that contains only the columns: **temp** (normalized temperature), **atemp** (feels-like temperature), **windspeed**, and **cnt** (total bike counts).

Preview the structure of the dataset using the `str()` function. For all future questions, you will use the **BikeData** dataset.

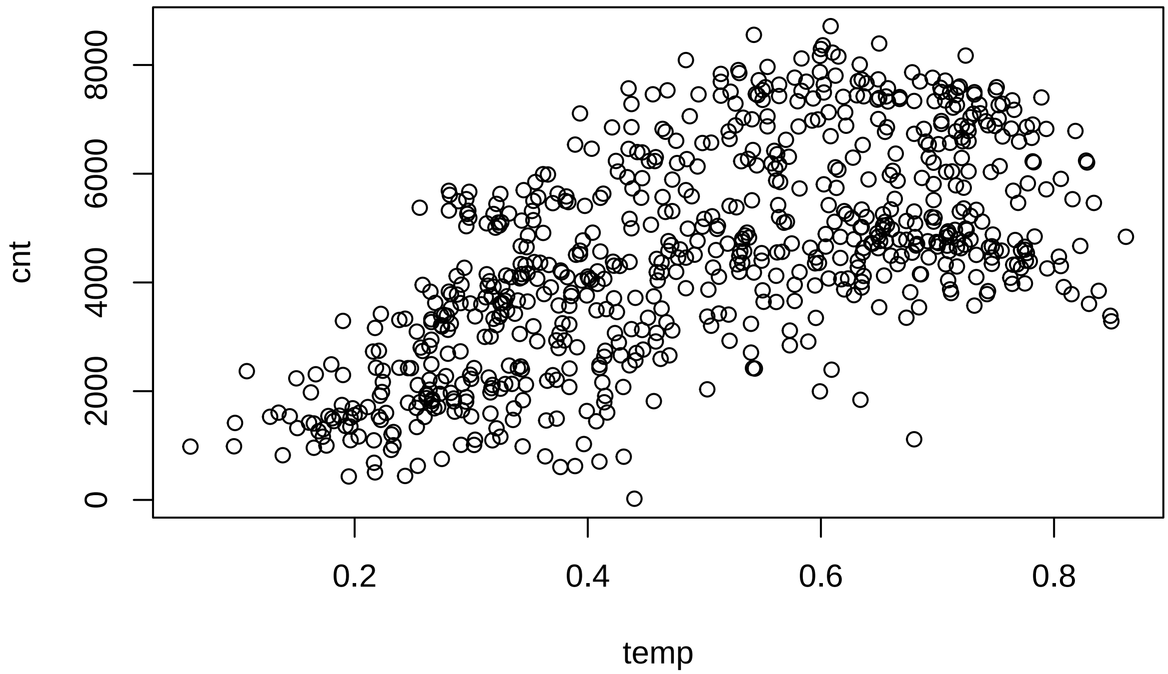
```
BikeData = bike_sharing[, c("temp", "atemp","windspeed", "cnt")]

str(BikeData)
```

```
## tibble [730 x 4] (S3: tbl_df/tbl/data.frame)
##  $ temp     : num [1:730] 0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp    : num [1:730] 0.364 0.354 0.189 0.212 0.229 ...
##  $ windspeed: num [1:730] 0.16 0.249 0.248 0.16 0.187 ...
##  $ cnt      : num [1:730] 985 801 1349 1562 1600 ...
```

Q3 (4 Points)

Create a scatterplot of **cnt** (bike counts) versus **temp** (normalized temperature). From the scatterplot, discuss whether you believe temperature is a good predictor of bike counts. Explain your reasoning in the space below in complete sentences.

```
plot(cnt ~ temp, data = BikeData)
```

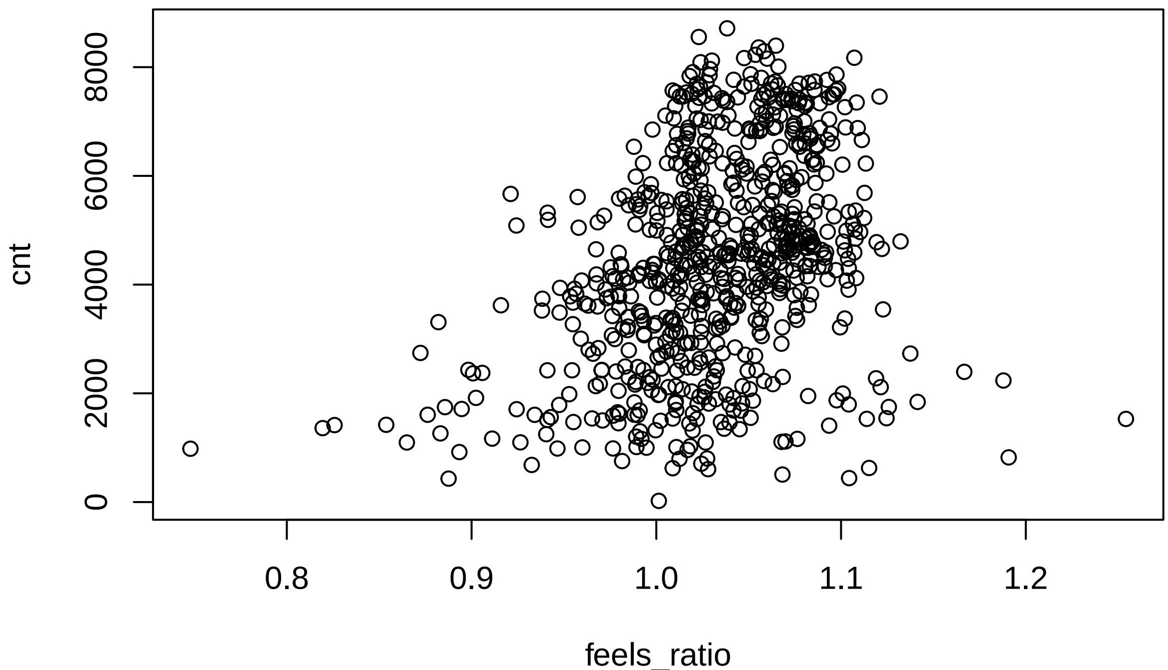


Response in Complete Sentences: : The data shows a positive correlation. Suggesting that as the temperature rises, bike rentals tend to increase. This indicates that temperature could be a good predictor of bike counts.

Q4 (3 Points)

Create a new variable in **BikeData** called **feels_ratio**, which is the ratio of the actual temperature to the feels-like temperature (temp/atemp). Then, generate a scatterplot of **cnt** (bike counts) versus **feels_ratio**. Does this new variable seem to provide useful information for predicting bike counts? Is this variable a better or worse predictor of bike counts compared to **temp** in Q3? Answer in complete sentences and explain why or why not.

```
feels_ratio = BikeData$temp / BikeData$atemp

BikeData$feels_ratio = feels_ratio

plot(cnt ~ feels_ratio, data = BikeData)
```



Response in Complete Sentences: *feels_ratio* does not appear to be a better predictor of bike counts compared to temperature alone. The normalized temperature show a more clearer pattern to be a more reliable predictor for the number of bike rentals.

Q5 (4 Points)

We want to know if a linear model can be useful for understanding the relationship between the temperature and bike counts. Fit a linear regression model with **cnt** as the response variable and **temp** as the explanatory variable. Print a summary of the model using the summary() function.

Next, in complete sentences, interpret the estimate of the slope to a student in the course who may have very little background in statistics. You should explain the slope in the context of the data. Think about what would be important to tell the student so that it is helpful in understanding the relationship between temperature and bike counts. You should convert the slope value to represent a change of 1 degree Celcius to improve the interpretation. Make sure you are precise in your language so you are not misleading someone to believe something that isn't guaranteed to be true.

Note, since the temperature values are normalized to be from 0 to 1, one unit increase in the temperature actually corresponds to an increase of 41 degrees Celsius.

```
mod1 = lm(cnt ~ temp, data = BikeData)

summary(mod1)
```

```
##
## Call:
```

```
## lm(formula = cnt ~ temp, data = BikeData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4611.8 -1134.6  -103.6  1042.5  3739.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1218.3      161.3   7.554 1.27e-13 ***
## temp          6630.1      305.6  21.695  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1510 on 728 degrees of freedom
## Multiple R-squared:  0.3927, Adjusted R-squared:  0.3918
## F-statistic: 470.7 on 1 and 728 DF,  p-value: < 2.2e-16
```

Response in Complete Sentences: There is a strong positive relationship between temperature and bike counts, with an increase of approximately 161.71 bikes for each degree Celsius increase in temperature. The slope(relationship between temperature and bike counts) is highly statistically significant (p-value < 2e-16), indicating that temperature is a meaningful predictor of bike counts.

Q6 (8 Points)

Now, fit a simple linear regression model with **cnt** as the response variable and **windspeed** as the explanatory variable. In your output, I don't care how much code you write, but I only want to see the following things in your output:

1. Summary of the model using `summary()` function (1 point)
2. Scatterplot showing the raw data with a "red" regression line plotted over the points. (2 points)
3. Boxplot of the residuals (1 point).
4. Scatterplot of the residuals versus the fitted values with a "blue" horizontal line representing perfect prediction. (2 points)
5. Histogram of residuals to check for normality. (1 point)
6. Numerical prediction of the number of bikes rented for a day with **windspeed** of 20. Provide a prediction with at least three decimal places. (1 point)

Note: **windspeed** values are normalized by dividing by the maximum value, 67.

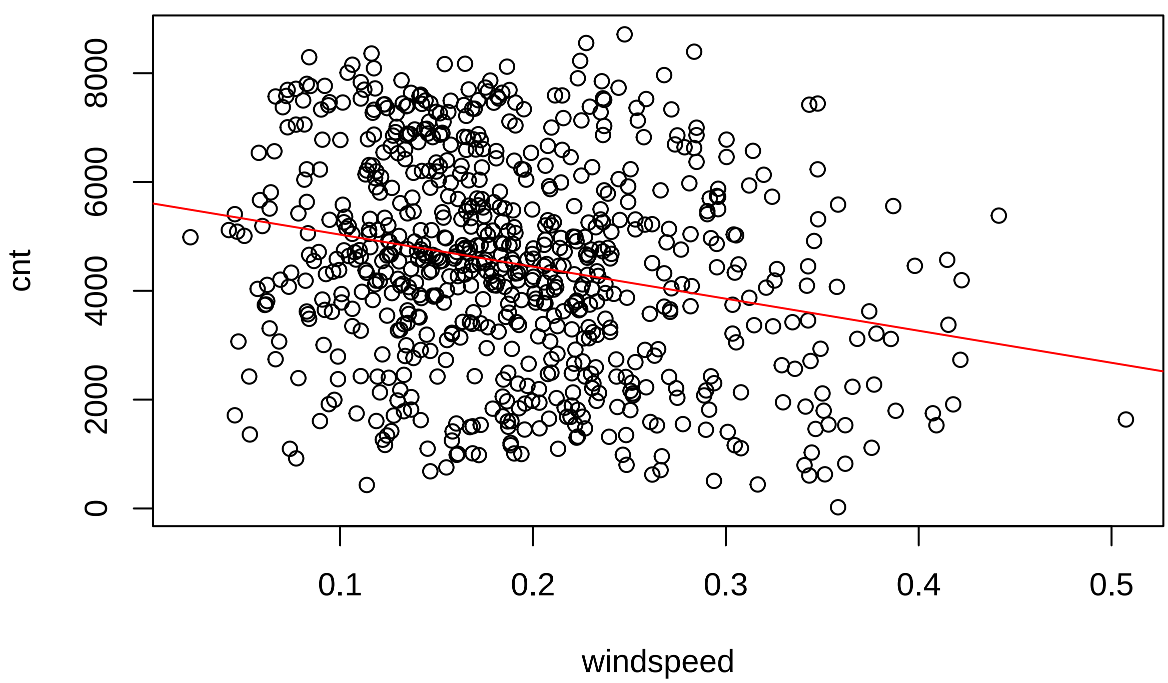
```r
mod2 = lm(cnt ~ windspeed, data = BikeData)
mod2
```

```
##
## Call:
## lm(formula = cnt ~ windspeed, data = BikeData)
##
## Coefficients:
## (Intercept)    windspeed
##        5622        -5890
```

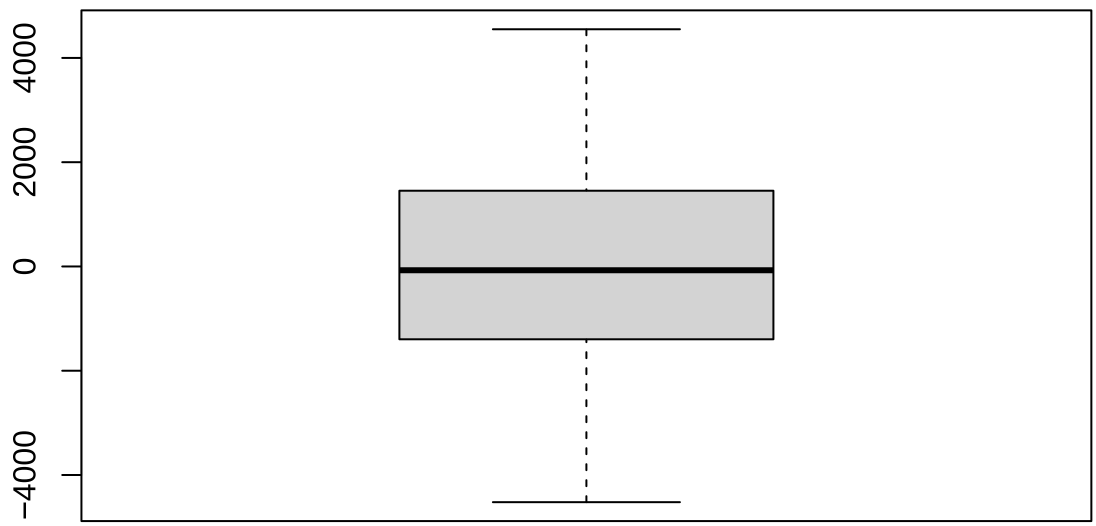
```r
# summary of the model
summary(mod2)
```

```
##
## Call:
## lm(formula = cnt ~ windspeed, data = BikeData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4520.8 -1384.0   -72.4  1450.7  4549.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5622.3      184.9  30.409  < 2e-16 ***
## windspeed    -5889.8      899.3  -6.549 1.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1883 on 728 degrees of freedom
## Multiple R-squared:  0.05564,    Adjusted R-squared:  0.05434
## F-statistic: 42.89 on 1 and 728 DF,  p-value: 1.093e-10
```

```r
# scatterplot of raw data
plot(cnt ~ windspeed, data = BikeData)
abline(mod2, col = 'red')
```



```r
# boxplot of the residuals
resid = mod2$residuals
boxplot(resid)
```




```r
#scatterplot of the residuals vs fitted values
mod2$fitted.values
```

```
##        1        2        3        4        5        6        7        8
## 4677.330 4158.476 4159.831 4678.213 4521.520 5094.806 4628.562 4050.898
##        9       10       11       12       13       14       15       16
## 3490.504 4307.324 4902.993 3828.127 3849.490 4876.983 4691.954 4512.491
##       17       18       19       20       21       22       23       24
## 4479.602 4757.850 4395.377 4468.488 3541.793 4609.455 4169.897 4689.792
##       25       26       27       28       29       30       31       32
## 4857.853 3891.602 4951.849 4896.113 4766.154 5186.585 4519.800 5308.914
##       33       34       35       36       37       38       39       40
## 4065.599 3986.417 4869.379 4673.601 4787.151 5354.883 3490.504 4510.099
##       41       42       43       44       45       46       47       48
## 4315.169 4982.988 4424.532 4085.772 3160.921 3906.185 4139.323 4267.055
##       49       50       51       52       53       54       55       56
## 4061.965 2633.456 4307.513 3809.168 4469.789 5068.020 4146.950 3581.272
##       57       58       59       60       61       62       63       64
## 4523.458 4884.640 3916.127 4347.622 3809.244 4292.676 4424.656 4138.851
##       65       66       67       68       69       70       71       72
## 3600.426 3611.823 4911.721 4325.683 4079.918 4250.175 4322.002 4028.517
```

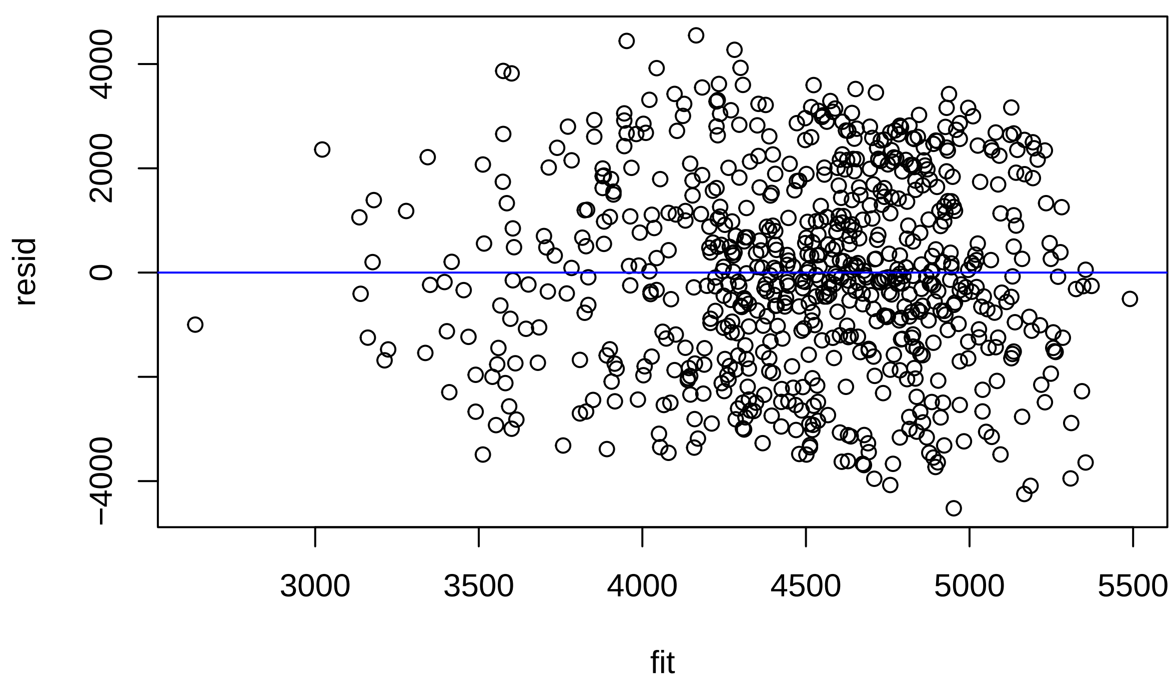
```
##       73       74       75       76       77       78       79       80
## 4815.859 4536.780 4426.004 4387.944 4261.678 3453.887 4398.888 3921.446
##       81       82       83       84       85       86       87       88
## 4292.700 4242.571 4186.465 4263.398 4387.991 4536.833 4372.483 4285.361
##       89       90       91       92       93       94       95       96
## 4604.048 4340.431 4098.583 4461.173 4549.126 3351.380 3336.679 4072.932
##       97       98       99      100      101      102      103      104
## 4666.339 4285.385 4838.493 4757.897 3711.232 4003.338 4146.237 4970.325
##      105      106      107      108      109      110      111      112
## 4289.019 3615.027 3834.789 4658.947 4691.907 4197.432 3706.614 4329.388
##      113      114      115      116      117      118      119      120
## 4263.398 4490.451 4530.749 3699.299 3783.523 3732.235 4208.399 4237.777
##      121      122      123      124      125      126      127      128
## 4995.923 4541.816 3604.078 3684.598 3882.520 4277.999 4677.306 5182.728
##      129      130      131      132      133      134      135      136
## 4585.719 4941.047 4911.769 4505.223 4563.779 4827.497 4721.309 4875.081
##      137      138      139      140      141      142      143      144
## 3988.761 4435.575 4984.973 4886.024 4911.721 4750.587 4245.039 4402.592
##      145      146      147      148      149      150      151      152
## 4713.923 4446.472 4204.770 4267.126 4362.270 4849.436 4966.621 4402.592
##      153      154      155      156      157      158      159      160
## 3900.808 4131.489 4897.044 4805.457 4904.383 4516.172 4816.501 4739.544
##      161      162      163      164      165      166      167      168
## 4794.490 4710.289 4658.947 3823.869 4036.298 4633.356 4406.250 4779.913
##      169      170      171      172      173      174      175      176
## 4919.037 5021.567 4706.608 4615.021 4607.735 4215.814 4314.639 4387.991
##      177      178      179      180      181      182      183      184
## 5065.545 4988.655 4772.527 4080.247 4530.873 5017.986 4944.634 4274.394
##      185      186      187      188      189      190      191      192
## 5142.431 4878.691 4739.544 4684.668 4296.357 4633.356 4541.716 3959.412
##      193      194      195      196      197      198      199      200
## 4442.867 4761.631 4204.841 4545.474 4395.230 4179.126 4351.280 4853.118
##      201      202      203      204      205      206      207      208
## 4951.967 4314.663 4838.393 4849.460 4625.941 5087.484 4442.843 4541.763
##      209      210      211      212      213      214      215      216
## 4571.118 4596.686 4629.675 4651.609 4699.293 4409.907 4823.769 4461.149
##      217      218      219      220      221      222      223      224
## 4534.501 4277.975 4435.605 4490.451 4732.252 4442.843 4651.709 4882.443
##      225      226      227      228      229      230      231      232
## 4376.901 4311.058 4391.625 4230.391 4776.155 4248.773 4801.829 5007.037
##      233      234      235      236      237      238      239      240
## 4157.210 3992.318 4757.920 4131.566 4380.558 5127.805 3410.008 3827.939
##      241      242      243      244      245      246      247      248
## 4680.987 4886.054 5131.511 4787.175 4798.171 4530.796 4406.273 4369.586
##      249      250      251      252      253      254      255      256
## 3596.562 5050.894 4487.076 4889.758 4717.604 4944.681 5098.647 4787.128
##      257      258      259      260      261      262      263      264
## 4636.961 4025.325 4655.319 4505.176 4571.095 4728.595 4827.473 5054.525
##      265      266      267      268      269      270      271      272
## 4867.695 5160.763 5160.665 5325.605 4970.325 4926.322 4746.930 4604.078
##      273      274      275      276      277      278      279      280
## 4406.226 3900.755 4314.710 5131.437 4409.884 4578.357 4827.473 5490.447
##      281      282      283      284      285      286      287      288
## 5354.907 5248.620 5373.165 4779.836 4160.767 4787.228 4303.696 4102.264
```

```
##      289      290      291      292      293      294      295      296
## 3963.063 4589.377 4973.936 4189.103 3135.200 4318.344 5076.539 5036.218
##      297      298      299      300      301      302      303      304
## 4922.665 4640.742 4746.853 4457.539 4270.737 3552.813 4582.085 4995.946
##      305      306      307      308      309      310      311      312
## 4823.839 5138.824 4816.501 4021.596 4508.786 5080.146 5285.284 5215.711
##      313      314      315      316      317      318      319      320
## 5256.007 4508.757 3768.946 4373.320 3963.040 3816.530 4446.525 4816.430
##      321      322      323      324      325      326      327      328
## 3823.798 4629.698 4300.086 4527.115 4809.215 4922.665 3644.376 4636.937
##      329      330      331      332      333      334      335      336
## 5039.850 5219.343 4391.625 4785.255 4102.211 4025.254 4325.636 5028.906
##      337      338      339      340      341      342      343      344
## 5058.182 5127.830 5255.933 4252.454 4054.603 4208.428 5135.142 4248.697
##      345      346      347      348      349      350      351      352
## 5230.410 5248.620 4794.514 5263.296 4043.607 4087.586 4190.116 4622.360
##      353      354      355      356      357      358      359      360
## 4604.001 5259.688 4318.321 5343.888 4007.066 4501.471 4708.870 4211.921
##      361      362      363      364      365      366      367      368
## 4512.391 3890.948 4919.013 4831.107 4325.659 4490.498 3680.657 3468.588
##      369      370      371      372      373      374      375      376
## 4534.478 4856.728 4633.380 4593.034 4494.179 5039.825 4517.680 4849.460
##      377      378      379      380      381      382      383      384
## 4556.464 3395.336 4519.853 4142.462 4259.670 3566.012 3175.522 4325.636
##      385      386      387      388      389      390      391      392
## 4428.166 4311.005 4446.495 4970.278 4893.363 4673.648 5190.041 3604.078
##      393      394      395      396      397      398      399      400
## 4380.582 4208.475 4351.351 4080.271 4508.757 4519.829 4571.018 4904.383
##      401      402      403      404      405      406      407      408
## 4585.743 4716.473 4754.292 4834.735 4479.484 4933.638 3915.479 3212.139
##      409      410      411      412      413      414      415      416
## 4637.061 4790.809 4552.736 5083.852 4409.931 4497.790 4131.542 4273.069
##      417      418      419      420      421      422      423      424
## 4410.691 4241.334 4497.884 4223.100 3138.928 4413.565 4043.660 4483.136
##      425      426      427      428      429      430      431      432
## 4567.360 4285.414 4768.869 4673.601 3651.762 4274.394 4439.209 3585.749
##      433      434      435      436      437      438      439      440
## 3021.597 3179.226 4292.700 4311.329 4398.935 4226.734 4944.634 4739.544
##      441      442      443      444      445      446      447      448
## 4955.648 4970.302 4875.010 4666.310 4908.064 5094.846 4929.909 4937.342
##      449      450      451      452      453      454      455      456
## 4486.870 4322.002 3344.018 4519.800 3904.436 3739.644 4809.138 4146.237
##      457      458      459      460      461      462      463      464
## 4607.706 3783.883 5032.563 4556.417 4329.340 3853.094 4003.385 4252.378
##      465      466      467      468      469      470      471      472
## 3512.615 4153.552 3883.215 3911.751 4706.632 4497.860 4296.357 3944.734
##      473      474      475      476      477      478      479      480
## 4010.700 4633.356 5234.018 4739.615 3952.049 3593.011 3834.789 4153.505
##      481      482      483      484      485      486      487      488
## 4922.665 4582.038 3574.829 4856.799 4933.761 4611.411 4702.950 4809.191
##      489      490      491      492      493      494      495      496
## 4834.883 4662.652 4721.232 4739.567 4263.421 3878.768 4347.699 3772.551
##      497      498      499      500      501      502      503      504
## 4226.810 4897.097 4296.428 4369.609 4754.215 4900.755 4270.760 4816.501
```

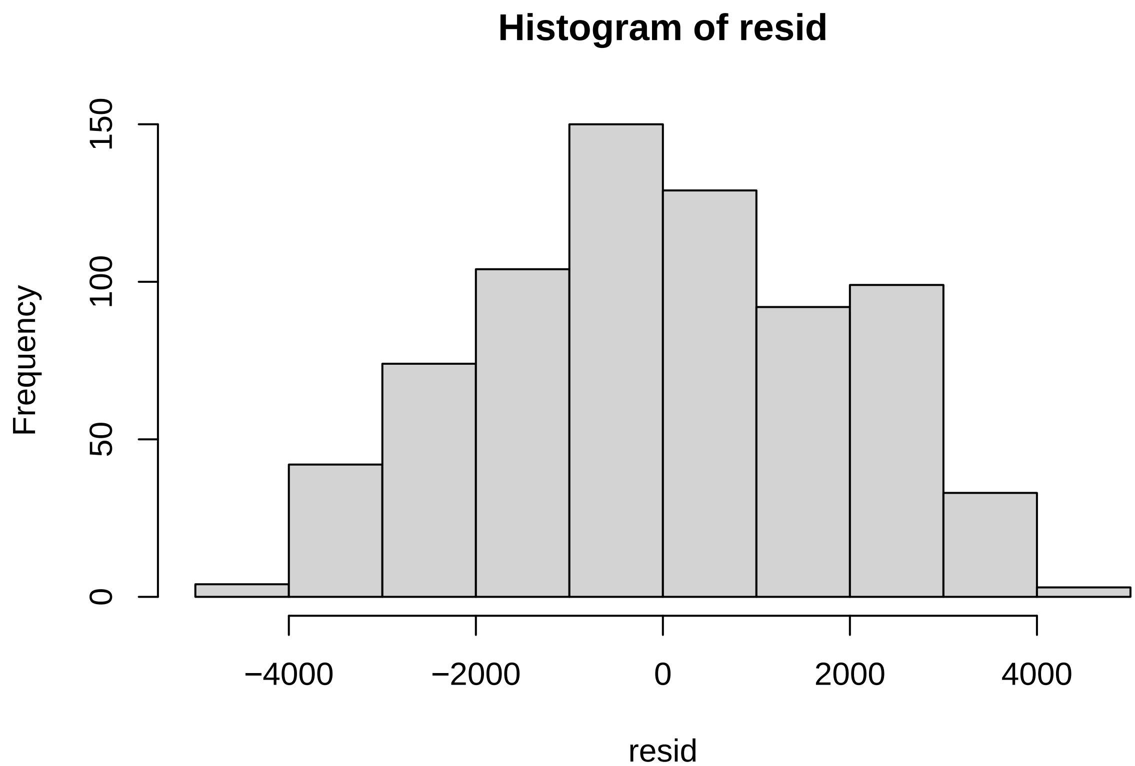
```
##      505      506      507      508      509      510      511      512
## 5127.731 4124.150 4248.797 4926.346 5021.567 4604.001 4794.514 4450.300
##      513      514      515      516      517      518      519      520
## 4355.008 4464.854 3882.473 4831.154 4472.169 4223.123 4523.511 4538.088
##      521      522      523      524      525      526      527      528
## 3944.711 4387.968 5168.077 4695.565 4589.353 4772.503 4834.735 4398.935
##      529      530      531      532      533      534      535      536
## 4358.689 3600.473 4127.884 4582.085 4640.689 4768.869 4593.105 4750.534
##      537      538      539      540      541      542      543      544
## 4951.996 4922.694 4545.421 4563.803 4765.212 3853.124 3574.776 4021.620
##      545      546      547      548      549      550      551      552
## 4611.340 4648.051 4673.648 4629.722 4472.240 4878.815 4827.497 4479.531
##      553      554      555      556      557      558      559      560
## 4761.578 4659.024 4882.396 4556.417 4728.624 4728.648 4757.850 5146.137
##      561      562      563      564      565      566      567      568
## 4776.084 4640.689 4655.296 4948.362 4812.819 4647.981 4391.549 4366.028
##      569      570      571      572      573      574      575      576
## 5069.152 4805.510 4376.901 4651.685 3944.828 4721.232 4695.565 4618.726
##      577      578      579      580      581      582      583      584
## 4717.551 4648.004 4790.809 4860.456 4351.351 4105.945 3911.798 4860.456
##      585      586      587      588      589      590      591      592
## 4933.761 4970.325 4702.927 4215.761 4406.203 4900.755 4820.064 4626.017
##      593      594      595      596      597      598      599      600
## 4622.407 4787.175 4574.723 5113.154 4856.775 5193.722 5208.373 5124.075
##      601      602      603      604      605      606      607      608
## 5197.403 4182.807 4274.394 4864.114 4497.813 4959.358 5168.125 4629.698
##      609      610      611      612      613      614      615      616
## 4955.677 5244.964 4732.252 4230.438 4516.172 4783.494 4611.363 3966.674
##      617      618      619      620      621      622      623      624
## 4300.086 4098.553 5080.146 4845.779 5135.118 5010.594 4164.472 5091.165
##      625      626      627      628      629      630      631      632
## 4728.595 3516.201 4354.985 4926.346 4713.947 3952.073 4307.377 4783.594
##      633      634      635      636      637      638      639      640
## 4230.438 4182.854 4831.078 4655.343 4281.780 4827.450 5087.508 5006.990
##      641      642      643      644      645      646      647      648
## 5230.386 4930.004 4995.946 4043.707 4790.909 4505.152 4501.494 4516.095
##      649      650      651      652      653      654      655      656
## 4552.760 4237.677 4761.578 3981.351 3878.721 4549.078 5025.272 4226.810
##      657      658      659      660      661      662      663      664
## 4827.473 4930.057 4644.299 5142.407 5065.471 5193.673 4889.782 4842.121
##      665      666      667      668      669      670      671      672
## 4234.143 3278.128 3512.591 4367.742 4640.689 4695.588 4054.603 4028.959
##      673      674      675      676      677      678      679      680
## 4567.413 4230.415 4600.367 3831.184 3573.639 4358.619 5281.628 4871.376
##      681      682      683      684      685      686      687      688
## 4600.344 3607.735 4446.572 4721.262 4615.021 4563.756 4281.733 4237.824
##      689      690      691      692      693      694      695      696
## 5135.093 5014.228 5310.930 4750.511 3402.622 4735.910 5347.569 4223.129
##      697      698      699      700      701      702      703      704
## 4380.629 4941.925 5277.947 5270.682 4889.758 5135.118 4596.739 3713.900
##      705      706      707      708      709      710      711      712
## 4593.058 4853.118 5025.224 4691.883 4501.447 3878.721 4662.658 4596.739
##      713      714      715      716      717      718      719      720
## 4849.413 4995.946 5028.976 5043.605 4318.297 4538.059 4842.145 3417.276
```

```
##      721      722      723      724      725      726      727      728
## 3223.129 4838.493 5167.456 4628.562 3757.926 3560.104 4706.632 4889.735
##      729      730
## 3556.447 4710.313
```

```r
fit = mod2$fitted.values

plot(fit, resid)
abline(h = 0, col = "blue")
```




```r
# histogram of residuals
hist(resid)
```




```r
# numerical prediction

predict(mod2, newdata=data.frame(windspeed = 20/67))
```

```
##       1
## 3864.17
```

Q7 (8 Points)

Write a summary in complete sentences of the simple linear regression model in Q6 where **cnt** is the response variable and **windspeed** is the explanatory variable. In your summary, include the following:

1. Interpretation of the regression coefficients, including the model formula. (3 points)
2. Analysis of the fit of the model, including R-squared and any significant predictors. (2 points)
3. Assessment of the residuals, including the distribution, presence of any patterns, and normality. (3 points)

Response in Complete Sentences:

#1.Interpretation Formula: $cnt = B0 + B1 windspeed$. *cnt* is the response variable (bike count). B0 representing the expected value of bike counts when windspeed is 0. B1 is the slope (coefficient of *windspeed*), which indicates the change in bike counts for a one-unit increase in *windspeed*. *windspeed* is the explanatory variable. The coefficient for *windspeed* is negative, indicating that as *windspeed* increases, the number of bike rentals tends to decrease. The slope value suggests that for each one-unit increase *windspeed*, bike rentals decrease by a certain number of units. The intercept represents the expected number of bike rentals when the *windspeed* is zero.

#2.Analysis The R-squared value shows how well the model explains the variance in the bike counts. A higher R-squared value indicates a better fit. The number of R-squared is 0.05564. This means that about 5.56% of the variance in bike counts is explained by the wind speed. So it explains only a small portion of the variation in the number of bike rentals.

The p-value for the *windspeed* coefficient is very low, indicating that *windspeed* is statistically significant in predicting the bike counts.

#3. Assessment residuals The boxplot shows the distribution of the residuals, with most residuals centered around zero and no extreme outliers.

There is no clear patterns on scatterplot, the residuals are scattered somewhat randomly around zero.

The histogram shows that the residuals are normally distributed.

Overall, the linear regression model is a reasonably good fit for the data.