Vilniaus Universitetas

Regresija įvykių skaičiui

Laboratorinis darbas

Darbą atliko:

Vainius Gataveckas, Matas Gaulia, Dovydas Martinkus

Duomenų Mokslas

3 kursas 2 gr.

Vilnius, 2022

Turinys

Naudoti metodai

Darbas atliktas naudojant R ir Python.

Naudoti R paketai:

tidyverse

AER

lMASS

rsample

corrplot

effects

yardstick

VGAM

Naudoti Python paketai:

numpy

pandas

matplotlib

seaborn

statsmodels

Duomenys ir jų šaltiniai

Išnuomotų dviračių kiekio pagal dienos ir oro sąlygų duomenys.

Duomenų šaltinis – Kaggle. Prieiga per internetą:

https://www.kaggle.com/datasets/brajeshmohapatra/bike-count-prediction-data-set?select=train.csv.

“Datetime” – data ir lakas.

“Season” – metų laikas.

“Holiday” – ar diena yra šventė.

“Workingday” – ar diena yra darbo.

“Weather” – kategorinis oro sąlygų kintamasis.

“Temp” – temperatūra Celcijaus laipsniais.

“Atemp” – jutiminė temperatūra Celcijaus laipsniais.

“Humidity” – oro drėgnumas.

“Windspeed” – vėjo greitis.

“Casual” – neregistruotų vartotojų išsinuomotų dviračių kiekis.

“Register” - registruotų vartotojų išsinuomotų dviračių kiekis.

“Counts” – bendras išsinuomotų dviračių kiekis (atsako kintamasis).

Tikslas ir uždaviniai

Tikslas: Sudaryti regresijos modelį išnuomotų dviračių skaičiui, įvertinti kokią įtaką tam tikri požymiai daro dviračių nuomos paklausai, panaudoti sudarytą modelį prognozuoti dviračių paklausą esant tam tikroms sąlygoms.

Uždaviniai:

Sudaryti įvykių skaičiaus regresijos modelius turimai duomenų aibei.

Atlikti sudarytų modelių tinkamumo analizę.

Tinkamiausio modelio parinkimas.

Modelio koeficientų interpretacija.

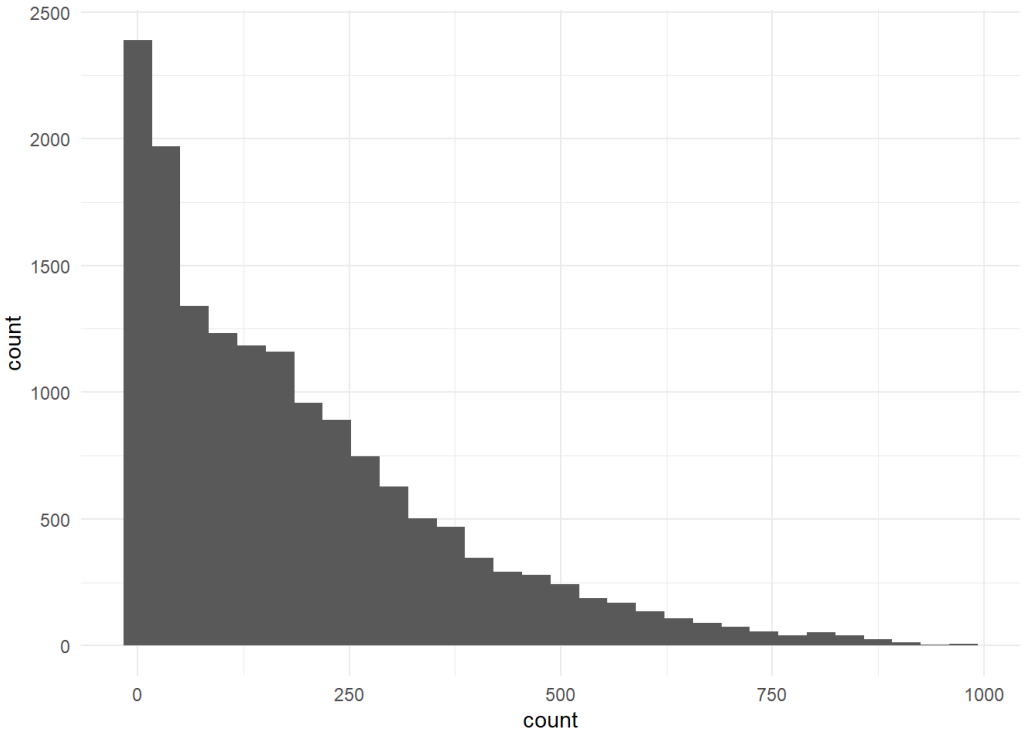
Modelio panaudojimas prognozuoti dviračių nuomos paklausą esant tam tikroms sąlygoms.

Atliktos analizės aprašymas

1. Naudojant R

Duomenų aibę sudaro 17379 stebėjimai. Duomenų aibėje nėra praleistų reikšmių. Duomenis pasirinkta padalinti į mokymo ir testavimo aibes naudojant 90-10 santykį.

```
# Duomenys
# https://www.kaggle.com/datasets/brajeshmohapatra/bike-count-prediction-data-set?select=train.csv

library(tidyverse)
library(AER)
library(MASS)


tr <- read.csv("train.csv")
te <- read.csv("test.csv")
te$count <- te$casual + te$registered
full <- rbind(tr, te)
full <- full %>%
  dplyr::select(-c(datetime, casual, registered)) %>%
 mutate(season = factor(season),
        holiday = factor(holiday),
        workingday = factor(workingday),
        weather = factor(weather))

head(full)

##   season holiday workingday weather temp  atemp humidity windspeed count
## 1      1       0          0       1 9.84 14.395       81    0.0000    16
## 2      1       0          0       1 9.02 13.635       80    0.0000    40
## 3      1       0          0       1 9.02 13.635       80    0.0000    32
## 4      1       0          0       1 9.84 14.395       75    0.0000    13
## 5      1       0          0       1 9.84 14.395       75    0.0000     1
## 6      1       0          0       2 9.84 12.880       75    6.0032     1

# Perdaromi mokymo ir testavimo duomenų rinkiniai
library(rsample)
full_split <- initial_split(full,prop = 0.9)
train <- training(full_split)
test <- testing(full_split)


min(train$count)

## [1] 1



ggplot(train, aes(x=count)) + geom_histogram() + theme_minimal()
```

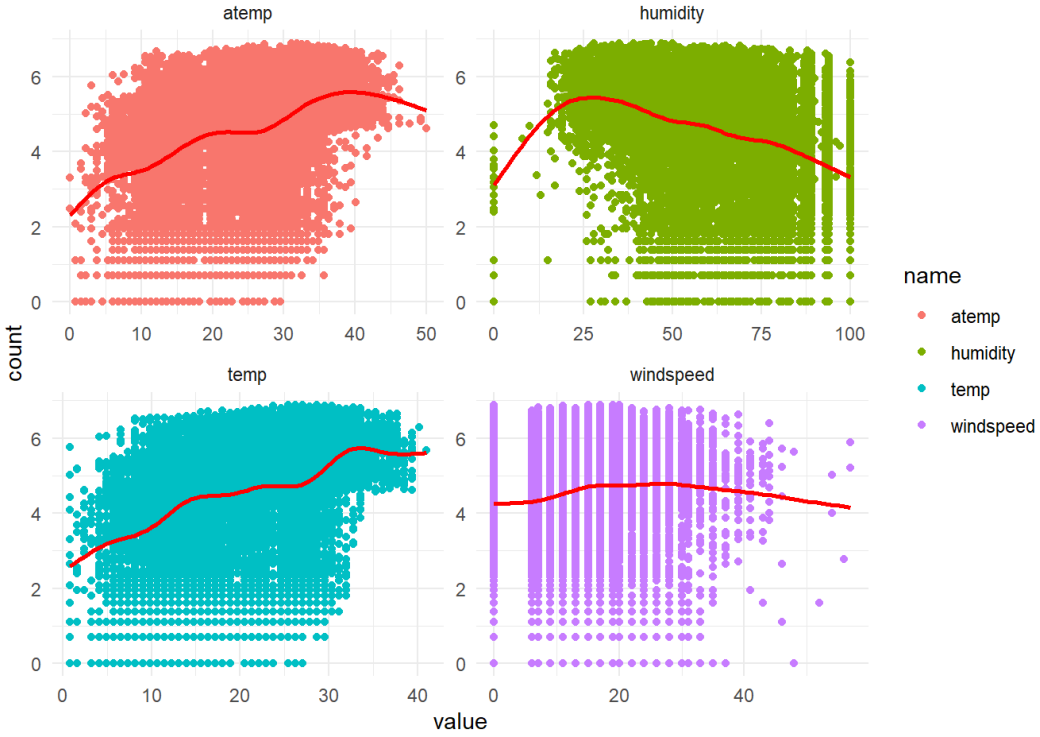



```
# Dispersija didesnė už vidurkį
mean(train$count)

## [1] 190.061

var(train$count)

## [1] 33048.88

train %>% mutate(q = ntile(count,6)) %>%
 group_by(q) %>%
 summarize(mean = mean(count), var = var(count))

## # A tibble: 6 x 3
##       q  mean    var
##   <int> <dbl>  <dbl>
## 1     1  8.24   26.1
## 2     2 42.3    226.
## 3     3 106.    418.
## 4     4 182.    566.
## 5     5 284.   1443.
## 6     6 517.  17617.
```

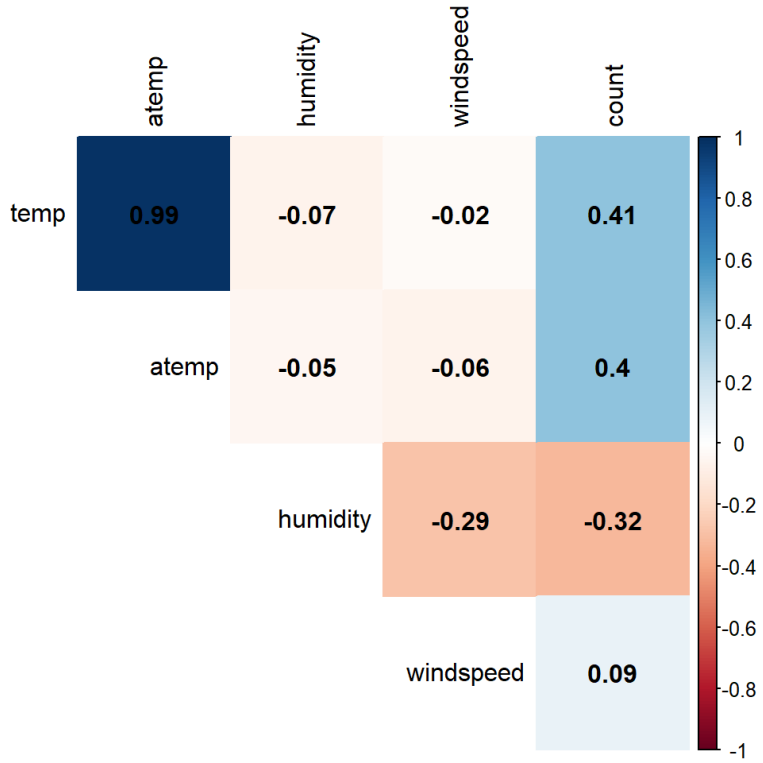
Duomenų aibėje esantys stebėjimai fiksuoti tik tada, kai buvo išnuomotas bent vienas dviratis, todėl duomenyse mažiausia esanti reikšmė yra 1. Apskritai pasirinkta laikyti, kad duomenis generuojantis procesas gali generuoti nulines reikšmes, tačiau tokių stebėjimų tiesiog neturima duomenų aibėje.

Iš atsako kintamojo histogramos matoma, kad turimi duomenys su dešiniąja asimetrija. Apskaičiavus aprašomosios statistikos charakteristikas rasta, kad išnuomotų dviračių skaičiaus dispersija stipriai didesnė už vidurkį, be to dispersijos ir vidurkio santykis nėra pastovus: šis santykis didėja esant didesnėms įvykių skaičiaus reikšmėmis. Dėl šių priežasčių daroma prielaida, kad Puasono regresijos modelis duomenims nėra tinkamas.

```
train %>% mutate(count = log(count)) %>% dplyr::select(c(temp,atemp,windspeed,humidity,count)) %>% pivot_longer(-count)
%>%
ggplot(aes(x = value, y = count, colour = name)) +
 geom_point() + geom_smooth(se=F,color="red") +
 facet_wrap(~name, scales = "free") + theme_minimal()
```




```
# Tarpusavio koreliacijos
library(corrplot)

correlation_matrix <- train %>% dplyr::select(where(is.numeric)) %>%
    cor()

corrplot(correlation_matrix, order = "original", method = "color",type="upper",diag=FALSE,tl.col = "black", addCoef.col =
"black")
```

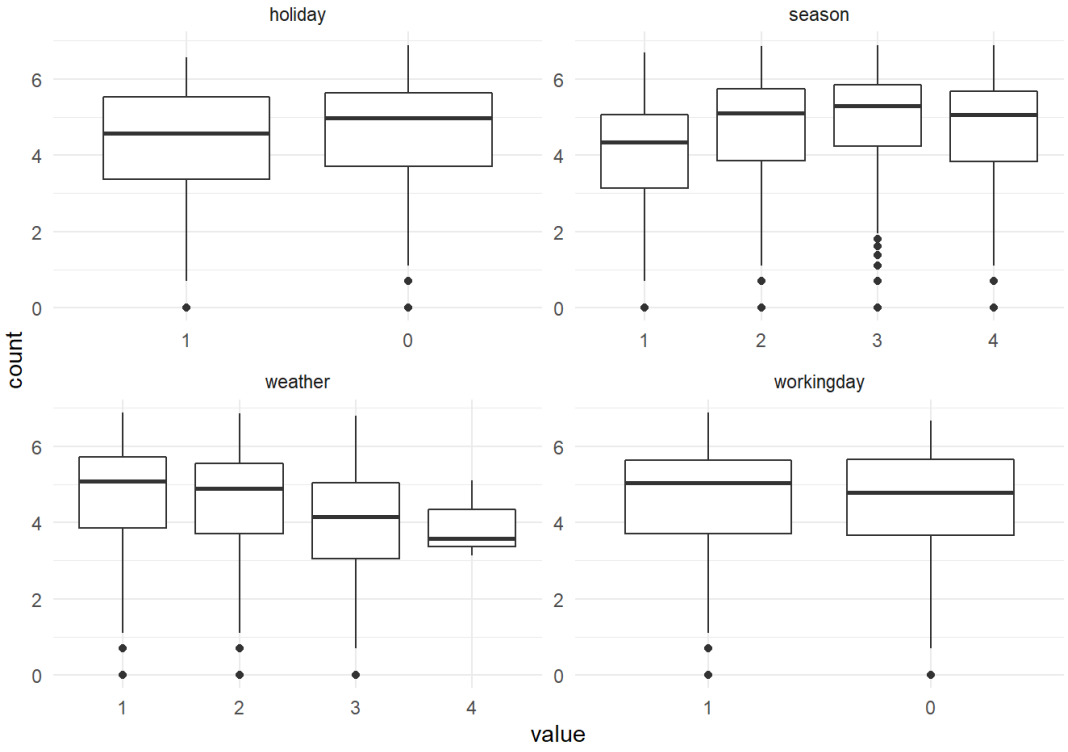


Rasti pakankamai tiesiniai ryšiai tarp skaitinių požymių ir išnuomotų dviračių skaičiaus logaritmo. Apskaičiavus koreliacijas tarp skaitinių duomenų aibės požymių rasta beveik visiškai tiesinis ryšys tarp oro temperatūros ir jutiminės oro temperatūros (r=0.99) . Dėl šios priežasties pasirinkta sudarant modelius kaip kovariantę įtraukti tik jutiminę oro temperatūrą.

```
train <- train %>% dplyr::select(-c(temp))

name <- full %>% dplyr::select(where(is.factor)) %>% names()

group <- function(x) {
 full %>%
   group_by(!!sym(x)) %>%
   summarize(mean = mean(count), var = var(count), n= n())
}

purrr::map(name,group)

## [[1]]
## # A tibble: 4 x 4
##   season  mean    var     n
##   <fct>  <dbl>  <dbl> <int>
## 1 1       111. 14214.  4242
## 2 2       208. 35480.  4409
## 3 3       236. 39090.  4496
## 4 4       199. 33477.  4232
##
## [[2]]
```

```
## # A tibble: 2 x 4
##   holiday  mean   var     n
##   <fct>   <dbl>  <dbl> <int>
## 1 0       190. 33117. 16879
## 2 1       157. 24573.   500
##
## [[3]]
## # A tibble: 2 x 4
##   workingday  mean   var     n
##   <fct>      <dbl>  <dbl> <int>
## 1 0          181. 29878.  5514
## 2 1          193. 34265. 11865
##
## [[4]]
## # A tibble: 4 x 4
##   weather  mean   var     n
##   <fct>   <dbl>  <dbl> <int>
## 1 1      205.  35906. 11413
## 2 2      175.  27368.  4544
## 3 3      112.  17897.  1419
## 4 4       74.3  6072.    3
```

```
train %>% mutate(count = log(count)) %>% dplyr::select(c(season,holiday,workingday,weather,count)) %>% pivot_longer(-count) %>%
ggplot(aes(x = value,y = count, group = value)) +
 geom_boxplot() +
 facet_wrap(~name, scales = "free") + theme_minimal()
```



Apskaičiuotas išnuomotų dviračių skaičiaus vidurkis esant skirtingiems kategorinių kintamųjų lygmenins, pasiskirstymas pavaizduotas stačiakampėmis diagramomis.

```r
# Puasono modelis
model_1 <- glm(count ~ ., family="poisson", data=train)
summary(model_1)
```

```
##
## Call:
## glm(formula = count ~ ., family = "poisson", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -26.930   -9.707   -2.988    4.583   41.893
##
## Coefficients:
##              Estimate Std. Error  z value Pr(>|z|)
## (Intercept)  4.637e+00  3.652e-03 1269.684  < 2e-16 ***
## season2      1.398e-01  2.273e-03   61.483  < 2e-16 ***
## season3     -4.008e-02  2.691e-03  -14.895  < 2e-16 ***
## season4      4.462e-01  2.061e-03  216.521  < 2e-16 ***
## holiday1    -1.448e-01  3.911e-03  -37.010  < 2e-16 ***
## workingday1  1.250e-02  1.300e-03    9.612  < 2e-16 ***
## weather2     9.612e-02  1.445e-03   66.517  < 2e-16 ***
## weather3    -1.460e-01  2.915e-03  -50.075  < 2e-16 ***
## weather4     4.252e-01  6.700e-02    6.346  2.2e-10 ***
## atemp        4.956e-02  1.083e-04  457.583  < 2e-16 ***
## humidity    -1.464e-02  3.588e-05 -408.064  < 2e-16 ***
## windspeed    4.707e-03  7.467e-05   63.040  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2606087  on 15640  degrees of freedom
## Residual deviance: 1832759  on 15629  degrees of freedom
## AIC: 1932662
##
## Number of Fisher Scoring iterations: 5
```

```r
cat("Deviacija padalinta iš laisvės laipsnių: ",model_1$deviance / model_1$df.residual,"\n")
```

```
## Deviacija padalinta iš laisvės laipsnių:  117.2666
```

```r
cat("Siekiama, kad būtų tarp 0.7 ir 1.3")
```

```
## Siekiama, kad būtų tarp 0.7 ir 1.3
```

```r
# Tikrinima hipotezė, kad modelis nėra per didelės dispersijos
dispersiontest(model_1, trafo = 2)
```

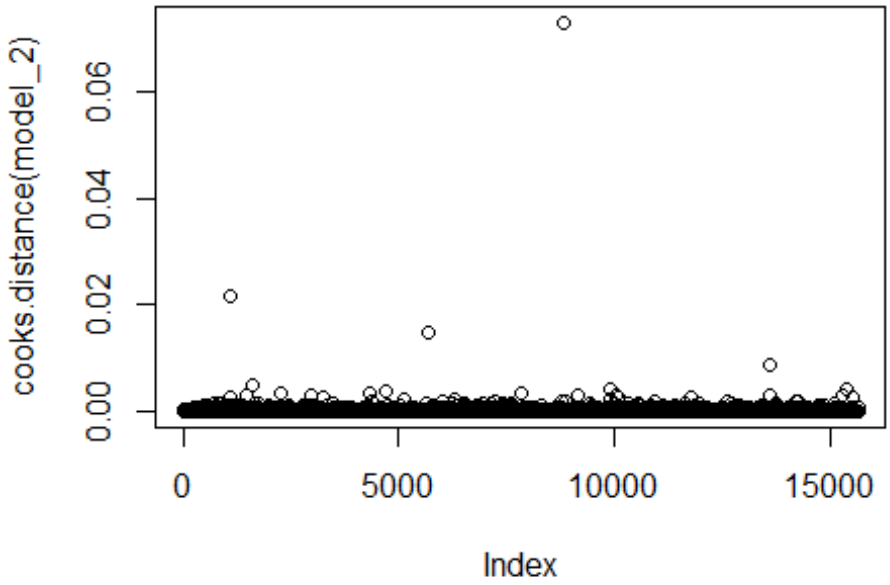
```
##
##  Overdispersion test
##
## data:  model_1
## z = 47.778, p-value < 2.2e-16
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##     alpha
## 0.5138437
```

Sudarytas Puasono regresijos modelis, naudojantis visas duomenyse esančias kovariantes, ir įvertintas pasitelkiant nykščio taisyklę, teigiančią, kad deviacija, padalinta iš jos laisvės laipsnių turi priklausyti intervalui [0.7,1.3]. Gauta reikšmė nepatenka į šį interval, kaip ir tikėtasi.

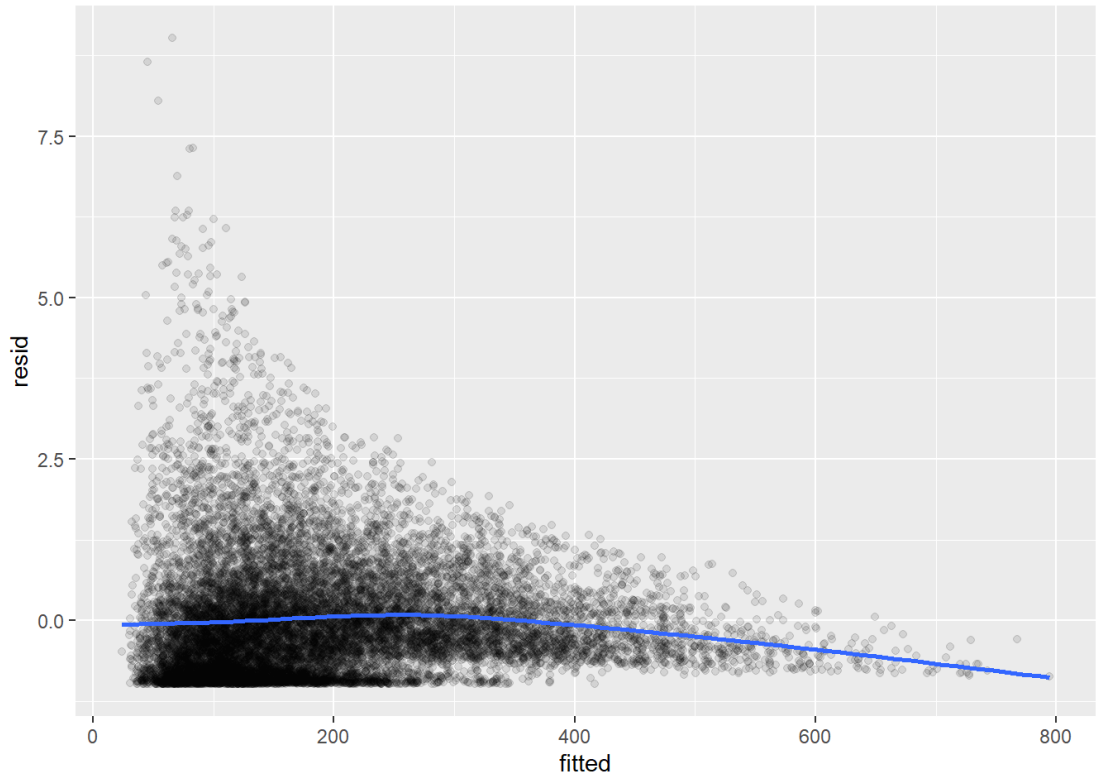
Hipotezė, kad modelio parametras *alpha* neigiamo binominio modelio dispersijos išraiškoje lygus 0 atmesta alternatyvai, kad parametro reikšmė didesnė už 0. Dėl šios priežasties pasirinkta sudaryti neigiamą binominį regresijos modelį.

```
# Neigiamas binominis modelis
model_2 <- glm.nb(count-1 ~ ., data = train)
summary(model_2)

##
## Call:
## glm.nb(formula = count - 1 ~ ., data = train, init.theta = 0.9875132249,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3080  -0.9454  -0.2370   0.3255   3.6912
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.4683659  0.0481042  92.889  < 2e-16 ***
## season2      0.0468850  0.0286283   1.638 0.101481
## season3     -0.1370228  0.0356018  -3.849 0.000119 ***
## season4      0.3897365  0.0252983  15.406  < 2e-16 ***
## holiday1    -0.1638248  0.0501087  -3.269 0.001078 **
## workingday1  0.1206863  0.0179617   6.719 1.83e-11 ***
## weather2     0.1600360  0.0197434   8.106 5.24e-16 ***
## weather3    -0.1162245  0.0329922  -3.523 0.000427 ***
## weather4     0.4347027  0.5858772   0.742 0.458106
## atemp        0.0572506  0.0014911  38.395  < 2e-16 ***
## humidity    -0.0156793  0.0004993 -31.402  < 2e-16 ***
## windspeed    0.0049474  0.0010552   4.689 2.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.9875) family taken to be 1)
##
##     Null deviance: 22643  on 15640  degrees of freedom
## Residual deviance: 18113  on 15629  degrees of freedom
## AIC: 190784
##
## Number of Fisher Scoring iterations: 1
##
##
##              Theta:  0.9875
##          Std. Err.:  0.0102
##
##  2 x log-likelihood:  -190758.3850

cat("Deviacija padalinta iš laisvės laipsnių: ",model_2$deviance / model_2$df.residual,"\n")

## Deviacija padalinta iš laisvės laipsnių:  1.15894
```

```
plot(cooks.distance(model_2))
```




```
tibble(fitted = model_2$fitted.values, resid = resid(model_2,"pearson")) %>%
ggplot(aes(fitted,resid)) + geom_point(alpha=0.1) + geom_smooth(se=F)
```



Sudarytas neigiamas binominis modelis, naudojantis visas kovariantes, atitinka prieš tai minėtą nykščio taisyklę. Naudojant Kuko mato reikšmių grafiką nerasta stiprių išskirčių.

```
# Stepwise selection
model_2_step <- stepAIC(model_2)

## Start:  AIC=190782.4
## count - 1 ~ season + holiday + workingday + weather + atemp +
##     humidity + windspeed
##
##             Df    AIC
## <none>         190782
## - holiday     1 190791
## - windspeed   1 190802
## - workingday  1 190825
## - weather     3 190878
## - season      3 191292
## - humidity    1 191696
## - atemp       1 192170



# Gaunamas lygiai toks pat modelis
anova(model_2, model_2_step)

## Likelihood ratio tests of Negative Binomial Models
##
## Response: count - 1
##                                                  Model
## 1 season + holiday + workingday + weather + atemp + humidity + windspeed
## 2 season + holiday + workingday + weather + atemp + humidity + windspeed
```

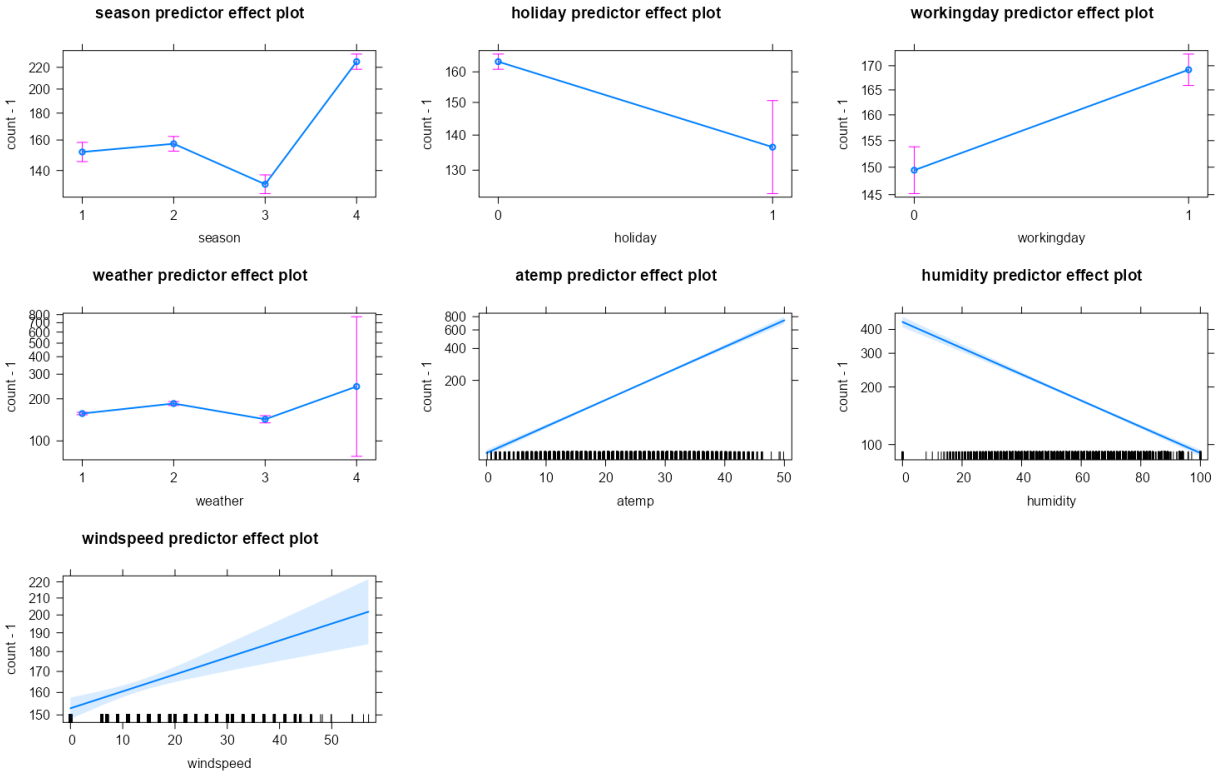
```
##      theta Resid. df    2 x log-lik.   Test    df LR stat. Pr(Chi)
## 1 0.9875132     15629      -190758.4
## 2 0.9875132     15629      -190758.4 1 vs 2     0       0      1
```

Naudojant pažingsninę regresiją gautamas toks pat modelis su visomis kovariantėmis.

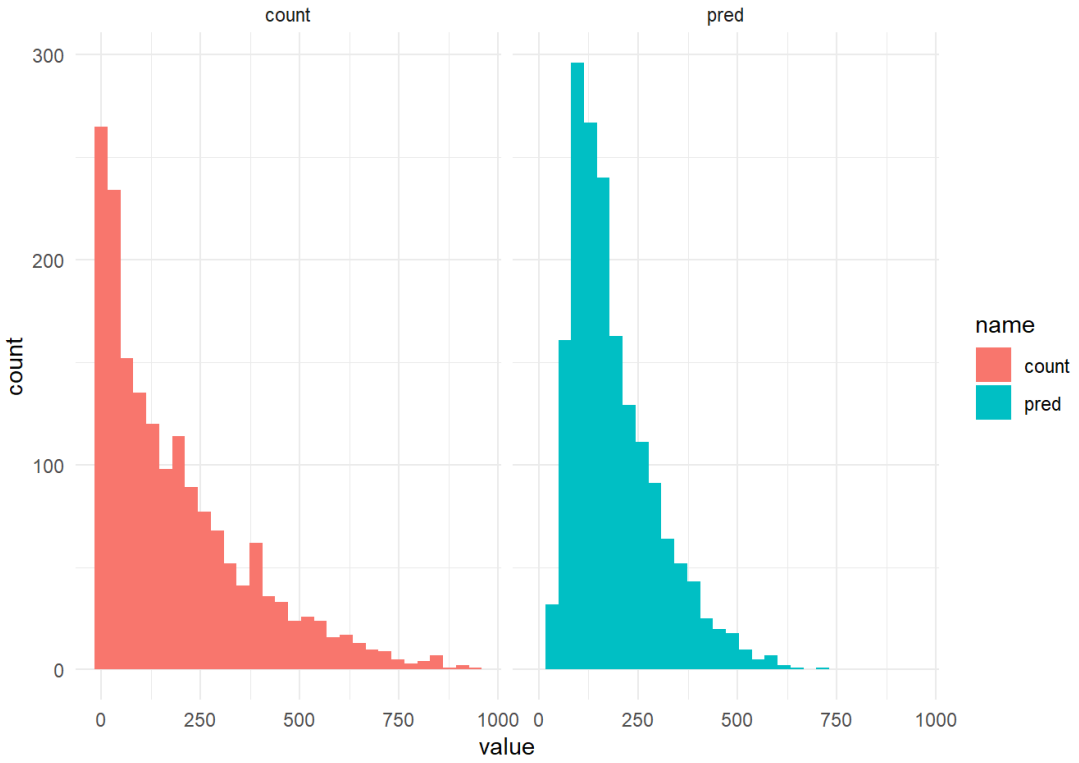
Gauto modelio koeficientų interpretacija įprasta modeliams, naudojantiems logaritminę jungties funkciją: koeficientų reikšmės atitinka kiekybinį atsako vidurkio logaritmo pokytį tą koeficientą atitinkančiai kovariantei pakitus vienetu, o likusioms kovariantėms esant fiksuotoms. Eksponencijuojant šiuos koeficientus gaunama kiek kartų padidėja atsako vidurkis.

Modelio gauti koeficientai panaudoti interpretuoti duomenyse esančių kovariačių ir atsako ryšį: Šventinėmis dienomis išnuomojama 16% mažiau dviračių negu įprastomis. Darbo dienomis išnuomojamų dviračių skaičius 12% procentų didesnis negu nedarbo dienomis. Didėjant temperatūrai ir mažėjant oro drėgnumui išnuomojama daugiau dviračių. Didesnis vėjo greitis teigiamai veikia išnuomojamų dviračių skaičių.

```
# Modelio koeficientų reikšmės
est <- cbind(Estimate = exp(coef(model_2)), exp(confint(model_2)))
est

##             Estimate     2.5 %     97.5 %
## (Intercept) 87.2140906 79.1320318 96.1533081
## season2      1.0480014  0.9914426  1.1077815
## season3      0.8719504  0.8153415  0.9324510
## season4      1.4765916  1.4064702  1.5502012
## holiday1     0.8488907  0.7704783  0.9377049
## workingday1  1.1282710  1.0891549  1.1686020
## weather2     1.1735532  1.1283699  1.2207516
## weather3     0.8902753  0.8348130  0.9501101
## weather4     1.5445039  0.5855209  6.3147840
## atemp        1.0589212  1.0558235  1.0620248
## humidity     0.9844430  0.9834496  0.9854365
## windspeed    1.0049597  1.0028406  1.0070870

library(effects)
plot(predictorEffects(model_2))
```




```
# Modelio panaudojimas prognozėms naudojant testavimo duomenų aibę
test_with_pred <- test %>% mutate(count = count,pred = predict(model_2_step,test, type = "response"))

test_with_pred %>%
  dplyr::select(c(count,pred)) %>% pivot_longer(everything()) %>%
  ggplot(aes(x=value,fill=name)) + geom_histogram() + theme_minimal() + facet_wrap(vars(name))
```



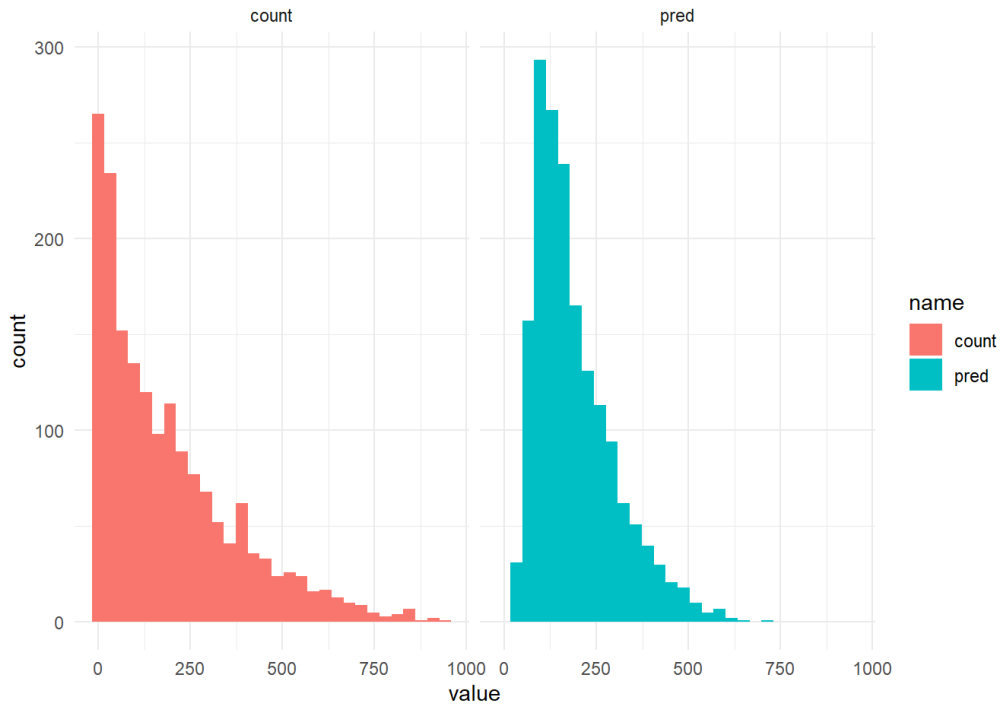

```
library(yardstick)

rmse(test_with_pred,count,pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       152.

mae(test_with_pred,count,pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mae     standard       112.
```

Modelis panaudotas atlikti prognozes naudojant testavimo aibės duomenis (RMSE=152, MAE=112). Iš prognozuotų ir stebėtų reikšmių histogramos pastebima, kad gautas modelis prastai prognozuoja ekstremalias reikšmės abiejose pusėse.

Kaip alternatyva sudarytas benulinis neigiamo binominio skirstinio regresijos modelis. Laikyta, kad toks modelis patencialiai gali tiksliau prognozuoti testavimo aibėje esančias reikšmes, nes šioje aibėje nėra nulinių reikšmių. Aišku toks modelis netiktų bendrai situacijai, kai įmanoma, kad išnuomotų dviračių nėra.

```
# Zero-truncated modelis kaip alternatyva
library(VGAM)
ztrunc <- vglm(count ~ ., family = posnegbinomial(), data = train)
```

```
summary(ztrunc)

##
## Call:
## vglm(formula = count ~ ., family = posnegbinomial(), data = train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  4.468613   0.047692  93.696  < 2e-16 ***
## (Intercept):2  0.004881   0.011217   0.435 0.663422
## season2        0.046896   0.028382   1.652 0.098472 .
## season3       -0.136994   0.035296  -3.881 0.000104 ***
## season4        0.389701   0.025081  15.538  < 2e-16 ***
## holiday1      -0.163800   0.049679  -3.297 0.000977 ***
## workingday1    0.120639   0.017807   6.775 1.25e-11 ***
## weather2       0.160002   0.019574   8.174 2.97e-16 ***
## weather3      -0.116211   0.032709  -3.553 0.000381 ***
## weather4       0.434703   0.580882   0.748 0.454249
## atemp          0.057243   0.001478  38.722  < 2e-16 ***
## humidity      -0.015678   0.000495 -31.671  < 2e-16 ***
## windspeed      0.004947   0.001046   4.729 2.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: loglink(munb), loglink(size)
##
## Log-likelihood: -95379.84 on 31269 degrees of freedom
##
## Number of Fisher scoring iterations: 6
##
## No Hauck-Donner effect found in any of the estimates

# Modelio prognozės
test_with_pred <- test %>% mutate(pred = predict(ztrunc,test, type = "response")[,1])

test_with_pred %>%
  dplyr::select(c(count,pred)) %>% pivot_longer(everything()) %>%
  ggplot(aes(x=value,fill=name)) + geom_histogram() + theme_minimal() + facet_wrap(vars(name))
```




```
rmse(test_with_pred,count,pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       152.

mae(test_with_pred,count,pred)

## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mae     standard       113.
```

Gautame modelyje visos kovariantės reikšmingos, gauti beveik identiški modelio koeficientai lyginant su paprastu neigiamu binominiu modeliu. Modelį įvertinant naudojant testavimo aibės reikšmes šis modelis taip pat prastai prognozuoja ekstremaliais reikšmes (RMSE=152, MAE=113).

Išvados:

Laikyta, kad Puasono regresijos modelis yra netinkamas dėl per didelės atsako dispersijos. Rasta, kad vidurkio ir dispersijos santykis yra didesnis tiems elementams, kurių vidurkis didesnis, todėl pasirinktas neigiamas binominis modelis.

Sudarytas neigiamas binominis modelis išnuomotų dviračių skaičiui. Gautame modelyje visos duomenyse esančios kovariantės reikšmingos.

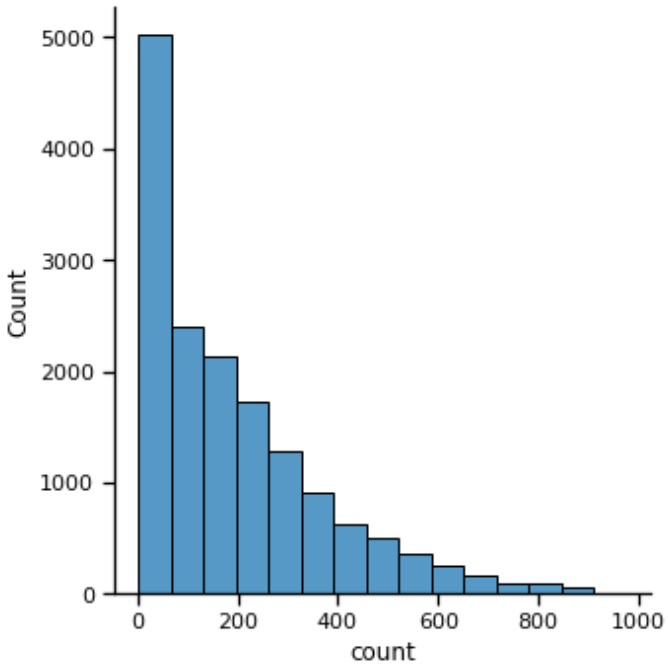
Naudojant modelį interpretuotas duomenyse esančių kovariančių ir atsako ryšys: Šventinėmis dienomis ir savaitgaliais išnuomojama atitinkamai 16% ir 12% mažiau dviračių. Didesnė oro temperatūra ir vėjo greitis teigiamai įtakoja dviračių nuomos paklausą, oro drėgnumas – neigiamai.

Dėl didelių gaunamų paklaidų prognozuojant išnuomotų dviračių kiekį naudojant testavimo aibę modelis netinkamas prognozuoti dviračių nuomos paklausai.

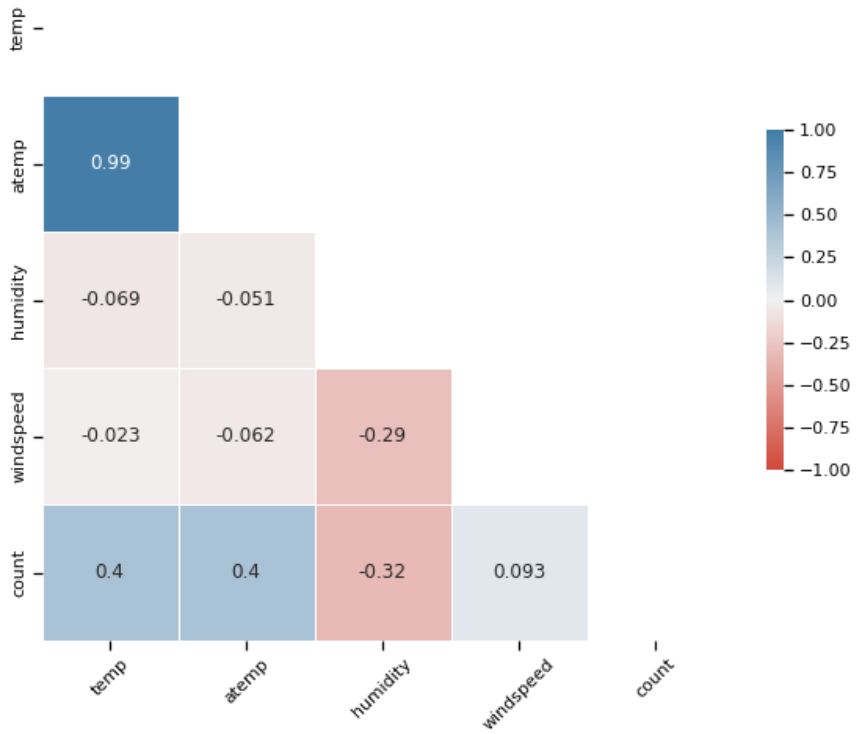
2. Naudojant Python

Atlikta analizė pakartotinai atlikta naudojant Python.

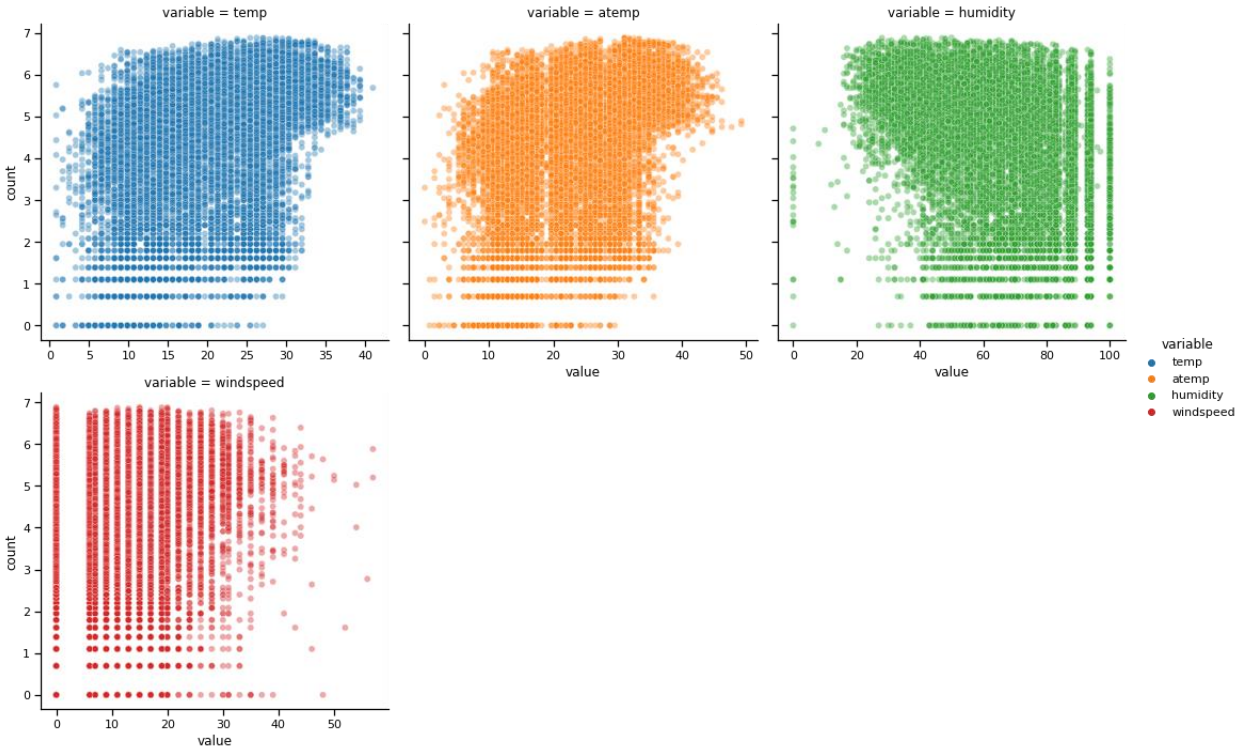
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns

test = pd.read_csv("test_from_R.csv")
train = pd.read_csv("train_from_R.csv")

train['season'] = train.season.astype('category')
train['holiday'] = train.holiday.astype('category')
train['workingday'] = train.workingday.astype('category')
train['weather'] = train.weather.astype('category')



test['season'] = test.season.astype('category')
test['holiday'] = test.holiday.astype('category')
test['workingday'] = test.workingday.astype('category')
test['weather'] = test.weather.astype('category').cat.add_categories(4)

sns.set_context("notebook")
sns.displot(x="count",data=train,kind="hist",bins=15)
```



```
df_numeric = train[["temp","atemp","humidity","windspeed","count"]]
corr = df_numeric.corr()
mask = np.triu(np.ones_like(corr, dtype=bool))

f, ax = plt.subplots(figsize=(10,8))
cmap = sns.diverging_palette(15,240,as_cmap=True)
plot=sns.heatmap(corr,mask=mask,vmax=1,vmin=-
1,center=0,cmap=cmap,square=True,cbar_kws={"shrink":.5},
            linewidth=1,ax=ax,annot=True)
plot.tick_params(axis='x', rotation=45)
```



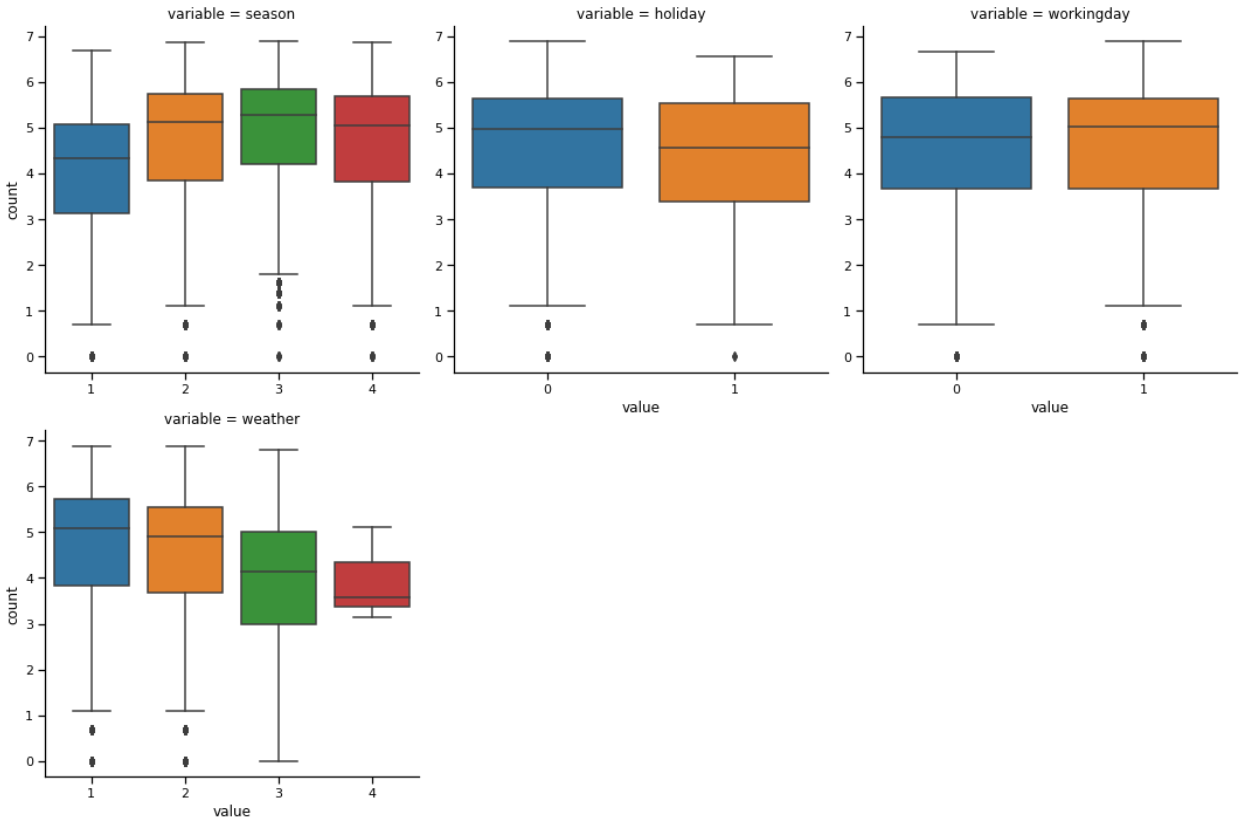

```
df_numeric = train[["temp","atemp","humidity","windspeed","count"]]
df_numeric["count"] = np.log(df_numeric["count"])
df_long = df_numeric.melt("count")

sns.relplot(x="value",y="count",data=df_long,col="variable",hue="variable",
            alpha=0.4,kind="scatter",col_wrap=3,facet_kws={'sharex': False})
```




```
train = train.drop("temp",axis=1)
test = test.drop("temp",axis=1)
```

```
df_categorical = train[["season","holiday","workingday","weather","count"]]
df_categorical["count"] = np.log(df_categorical["count"])
df_long = df_categorical.melt("count")
```

```
sns.catplot(x="value",y="count",data=df_long,col="variable",kind="box",col_wrap=3,sharey=False
,sharex=False)
```




```
import patsy

y, X = patsy.dmatrices('count ~ season + holiday + workingday + weather + atemp + humidity +
windspeed',
                       data=train, return_type='dataframe')

y_test, X_test = patsy.dmatrices('count ~ season + holiday + workingday + weather + atemp +
humidity + windspeed',
                       data=test, return_type='dataframe')
```

```
model_1=sm.GLM(y,X,family=sm.families.Poisson())
res_1=model_1.fit()
res_1.summary()
res_1.deviance / res_1.df_resid
```

```
117.83461428923191
```

```
model_2=sm.GLM(y,X,family=sm.families.NegativeBinomial())
res_2 = model_2.fit()
res_2.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	count	**No. Observations:**	15641
Model:	GLM	**Df Residuals:**	15629
Model Family:	NegativeBinomial	**Df Model:**	11

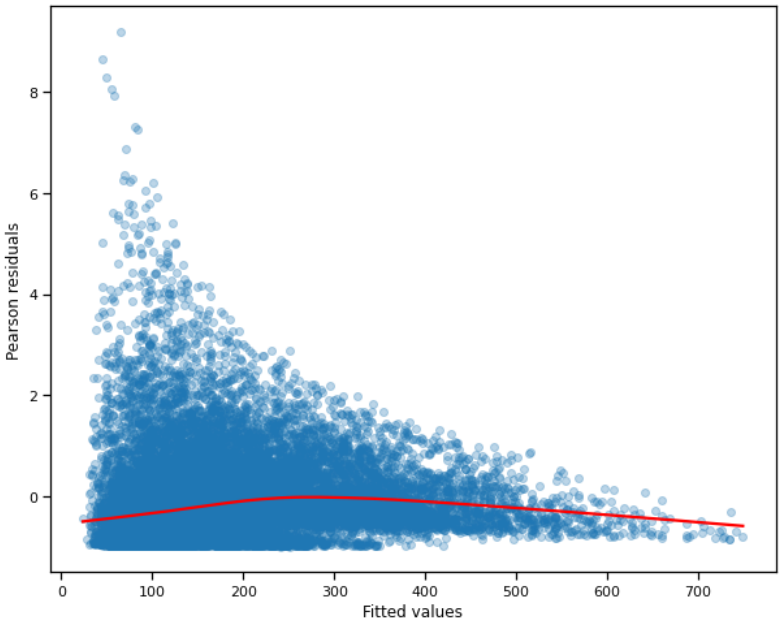
Link Function:	log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-95450.
Date:	Fri, 25 Mar 2022	**Deviance:**	17232.
Time:	21:31:12	**Pearson chi2:**	1.38e+04
No. Iterations:	8		
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	4.4686	0.048	93.570	0.000	4.375	4.562
season[T.2]	0.0455	0.028	1.603	0.109	-0.010	0.101
season[T.3]	-0.1435	0.035	-4.069	0.000	-0.213	-0.074
season[T.4]	0.3857	0.025	15.350	0.000	0.336	0.435
holiday[T.1]	-0.1716	0.050	-3.450	0.001	-0.269	-0.074
workingday[T.1]	0.1247	0.018	6.967	0.000	0.090	0.160
weather[T.2]	0.1535	0.020	7.846	0.000	0.115	0.192
weather[T.3]	-0.1029	0.033	-3.138	0.002	-0.167	-0.039
weather[T.4]	0.4365	0.582	0.750	0.453	-0.705	1.578
atemp	0.0573	0.001	38.639	0.000	0.054	0.060
humidity	-0.0156	0.000	-31.404	0.000	-0.017	-0.015
windspeed	0.0049	0.001	4.685	0.000	0.003	0.007

```
res_2.deviance / res_2.df_resid
```

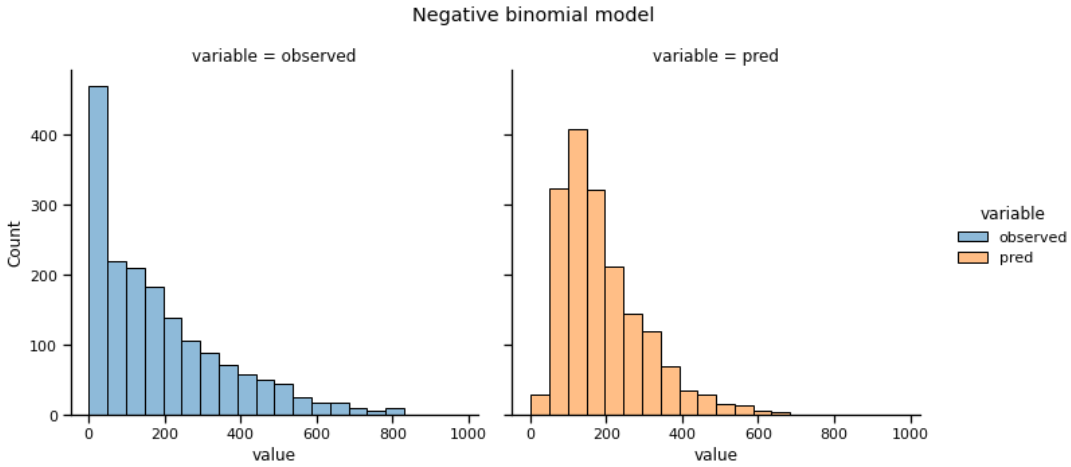
```
1.1025645630645862
```

```
fig, ax = plt.subplots(1,1,figsize=(10, 8))
ax =
sns.regplot(res_2.mu,res_2.resid_pearson,ax=ax,scatter_kws={"alpha":0.3},line_kws={"color":"re
d"},lowess=True)
ax.set_xlabel("Fitted values")
ax.set_ylabel("Pearson residuals")
```




```
np.exp(res_2.params)
```

```
Intercept           87.230857
season[T.2]          1.046529
season[T.3]          0.866350
season[T.4]          1.470679
holiday[T.1]         0.842355
workingday[T.1]      1.132844
weather[T.2]         1.165865
weather[T.3]         0.902246
weather[T.4]         1.547219
atemp                1.059012
humidity             0.984545
windspeed            1.004926
dtype: float64
results =
pd.DataFrame({"observed":test["count"],"pred":res_2.get_prediction(X_test).predicted_mean})
ax = sns.displot(x="value",col="variable",hue="variable",data=results.melt(),bins=20)
ax.fig.subplots_adjust(top=0.85)
ax.fig.suptitle("Negative binomial model")
```




```
from sklearn.metrics import mean_squared_error, mean_absolute_error

print("RMSE:",np.sqrt(mean_squared_error(results["observed"],results["pred"])))

print("MAE:",mean_absolute_error(results["observed"],results["pred"]))
```

```
RMSE: 154.3438454234047
MAE: 112.84354144093801
```

```
model_ztrunc = sm.NegativeBinomialP(y,X)
res_ztrunc = model_ztrunc.fit()
```

```
res_ztrunc.summary()
```

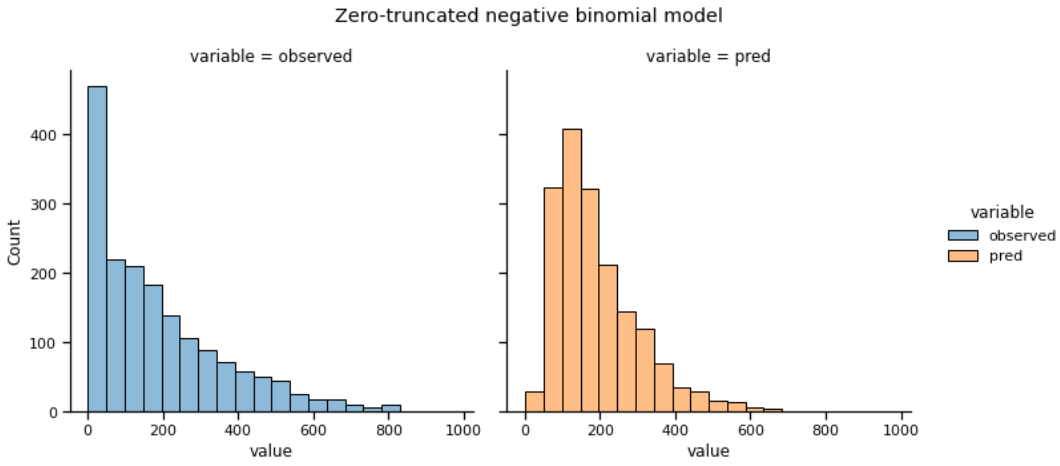
NegativeBinomialP Regression Results

Dep. Variable:	count	**No. Observations:**	15641
Model:	NegativeBinomialP	**Df Residuals:**	15629
Method:	MLE	**Df Model:**	11
Date:	Fri, 25 Mar 2022	**Pseudo R-squ.:**	0.02161
Time:	21:23:12	**Log-Likelihood:**	-95442.
converged:	False	**LL-Null:**	-97551.
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
Intercept	4.4685	0.049	92.039	0.000	4.373	4.564
season[T.2]	0.0456	0.028	1.656	0.098	-0.008	0.099
season[T.3]	-0.1434	0.033	-4.302	0.000	-0.209	-0.078

season[T.4]	0.3858	0.024	15.975	0.000	0.338	0.433
holiday[T.1]	-0.1720	0.049	-3.528	0.000	-0.268	-0.076
workingday[T.1]	0.1247	0.018	7.100	0.000	0.090	0.159
weather[T.2]	0.1534	0.020	7.855	0.000	0.115	0.192
weather[T.3]	-0.1030	0.032	-3.193	0.001	-0.166	-0.040
weather[T.4]	0.4254	0.568	0.749	0.454	-0.687	1.538
atemp	0.0573	0.001	39.192	0.000	0.054	0.060
humidity	-0.0156	0.001	-31.009	0.000	-0.017	-0.015
windspeed	0.0049	0.001	4.669	0.000	0.003	0.007
alpha	0.9614	0.010	97.429	0.000	0.942	0.981

```
results_ztrunc = pd.DataFrame({"observed":test["count"],"pred":res_ztrunc.predict(X_test)})
ax = sns.displot(x="value",col="variable",hue="variable",data=results_ztrunc.melt(),bins=20)
ax.fig.subplots_adjust(top=0.85)
ax.fig.suptitle("Zero-truncated negative binomial model")
```




```
print("RMSE:",np.sqrt(mean_squared_error(results_ztrunc["observed"],results_ztrunc["pred"])))

print("MAE:",mean_absolute_error(results_ztrunc["observed"],results_ztrunc["pred"]))
```

```
RMSE: 154.34331245939458
MAE: 112.84345477317076
```