Poisson and quasipoisson regression to predict counts

Enrique J. De La Hoz D.

UTB - Data Science

Predicting Counts

- Linear regression: predicts values in $[-\inf, inf]$
- Counts: integers in range $[0, inf]$

Poisson/Quasipoisson Regression

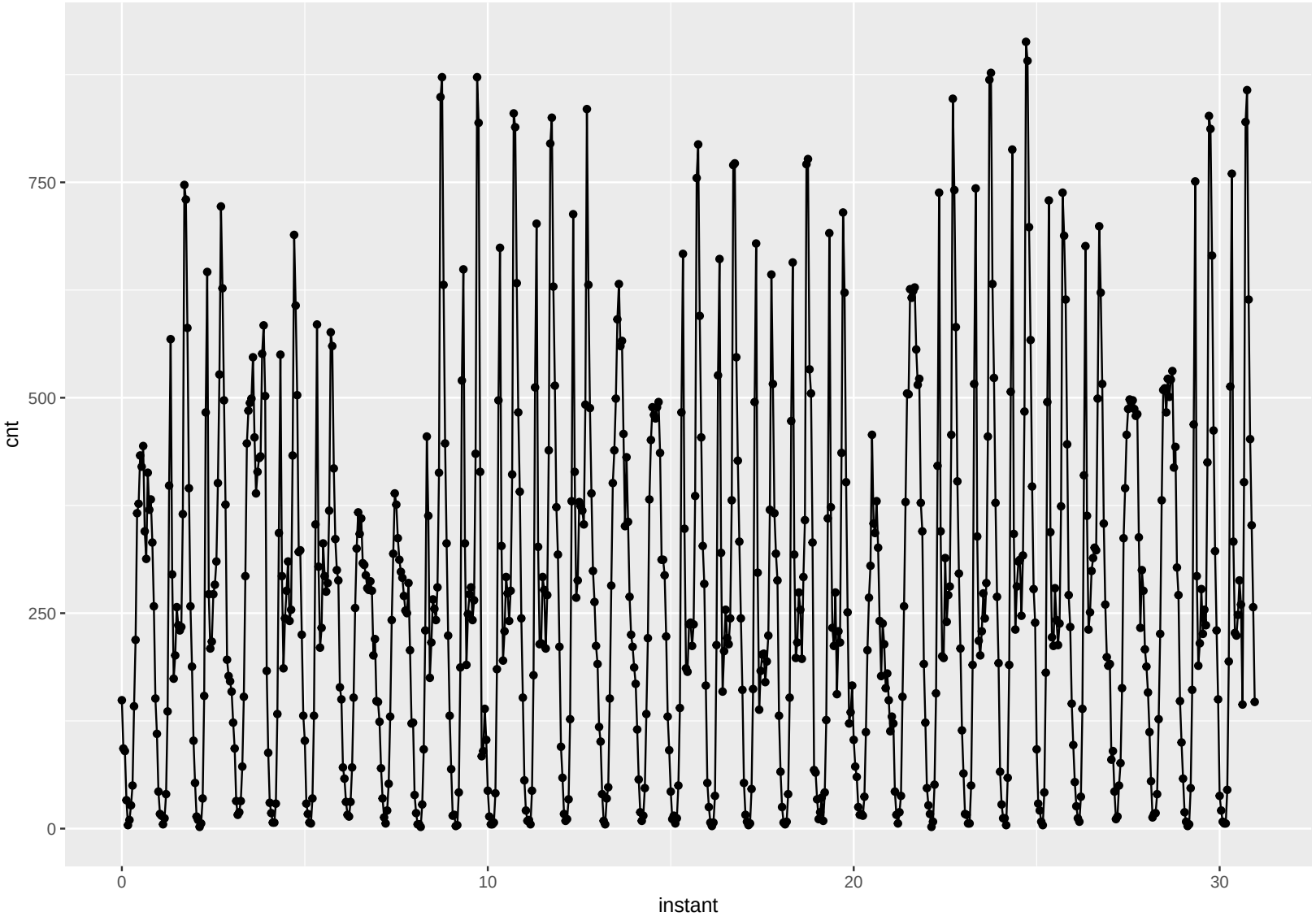
```
glm(formula, data, family)
library(broom)
```

- family: either poisson or quasipoisson
- inputs additive and linear in log(count)
- outcome: integer
  - counts: e.g. number of traffic tickets a driver gets
  - rates: e.g. number of website hits/day
- prediction: expected rate or intensity (not integral)
  - expected # traffic tickets; expected hits/day

Poisson vs. Quasipoisson

- Poisson assumes that mean(y) = var(y)
- If var(y) much different from mean(y) - quasipoisson
- Generally requires a large sample size
- If rates/counts >> 0 - regular regression is fine

Example: Bike rentals prediction



Fit the model

```
bikesJuly %>%
 summarize(mean = mean(cnt), var = var(cnt))
```

```
      mean      var
1 273.6653 45863.84
```

Since var(cnt) >> mean(cnt) it is better to use quasipoisson

GLM Family

```
fmla <- cnt ~ hr + holiday + workingday+
  weathersit + temp + atemp + hum + windspeed

model <- glm(fmla, data = bikesJuly,
             family = quasipoisson)
```

- Check the predictors variables

Check model fit

$$pseudoR^2 = 1 - \frac{deviance}{Null.deviance}$$

```
library(broom)
glance(model) %>%
 summarize(pseudoR2 = 1 - deviance/null.deviance)
```

```
   pseudoR2
1 0.7842393
```

Predicting from the model

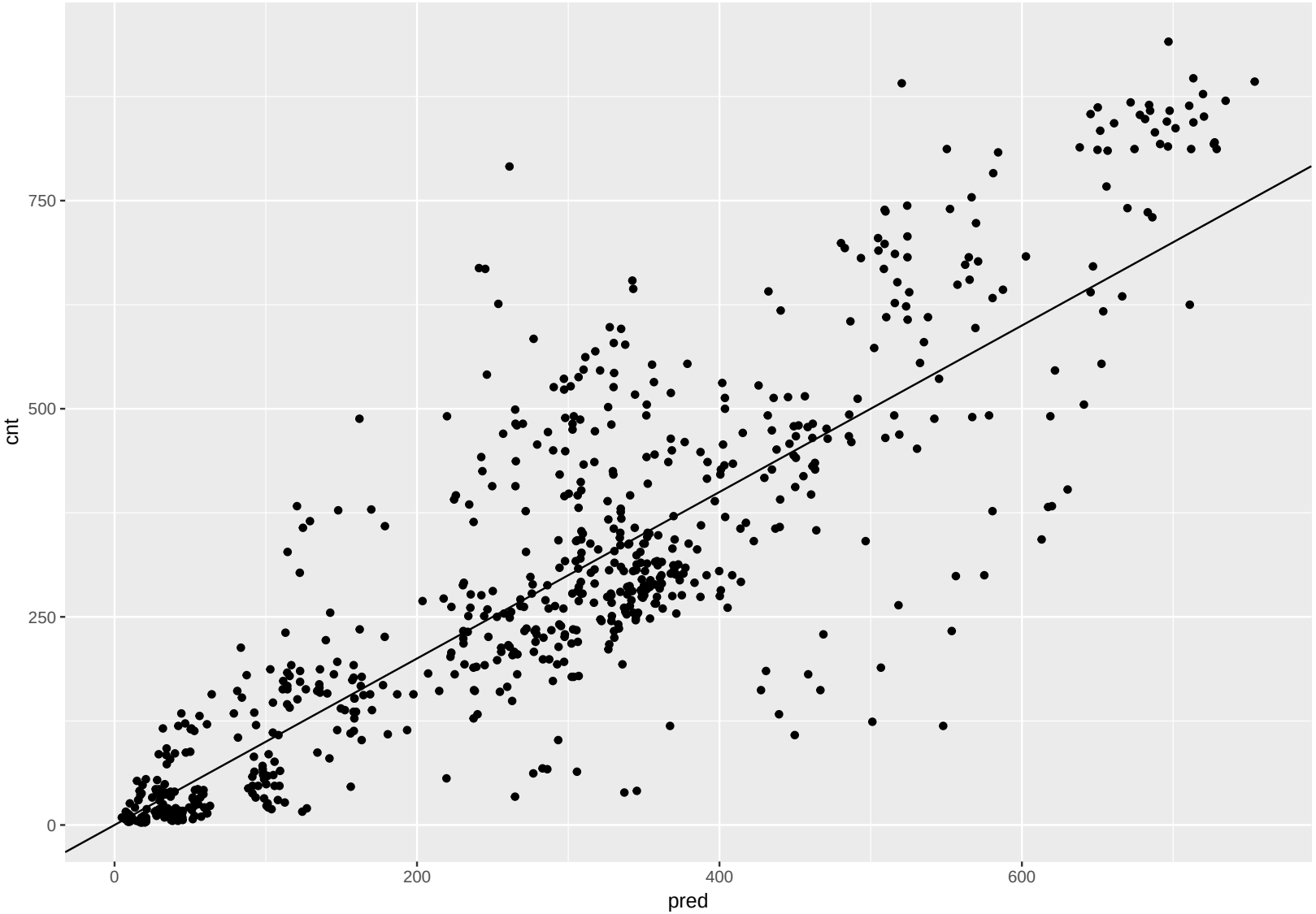
```
bikesAugust$pred<- predict(model, newdata = bikesAugust,
                           type = "response")
```

Evaluate the model

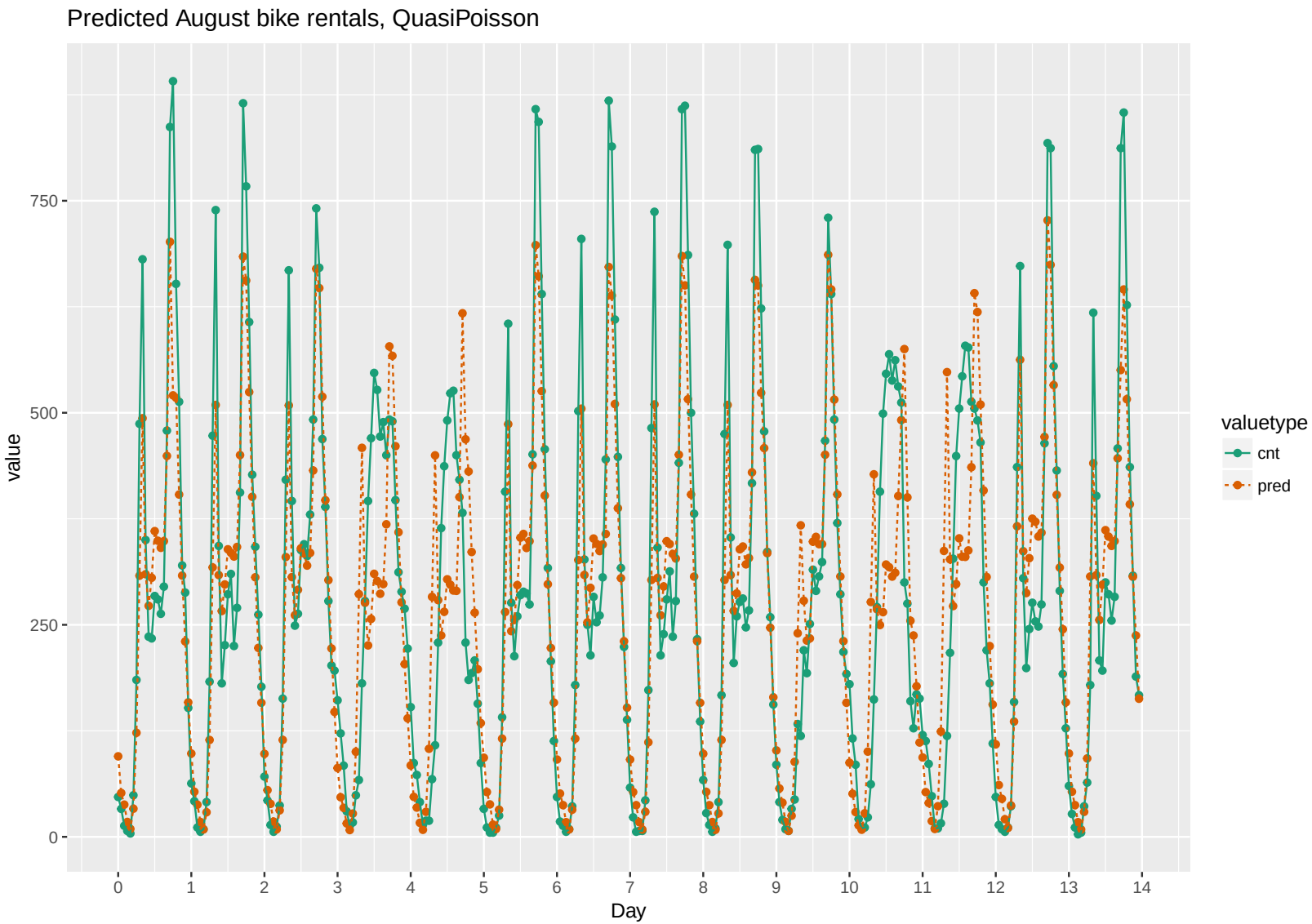
- It seem to be that the model is performing well

```
     rmse   Stdev
1 112.5815 227.875
```

Visual Inspection



Compare Predictions and Actual Outcomes



Code in R

```
library(tidyr)

xg_plot<-bikesAugust %>%
    # set start to 0, convert unit to days
  mutate(instant = (instant - min(instant))/24) %>%
  gather(key = valuetype, value = value, cnt, pred) %>%
  filter(instant < 14) %>% # first two weeks
  ggplot(aes(x = instant, y = value, color = valuetype,
             linetype = valuetype)) +
  geom_point() +
  geom_line() +
  scale_x_continuous("Day", breaks = 0:14, labels = 0:14) +
  scale_color_brewer(palette = "Dark2") +
  ggtitle("Predicted August bike rentals, QuasiPoisson")

xg_plot
```

Non Linear transformations GAM

- Generalized Additive Models (GAMs)

$$y = b_0 + s1(x1) + s2(x2)...$$

Learning Non-linear Relationships

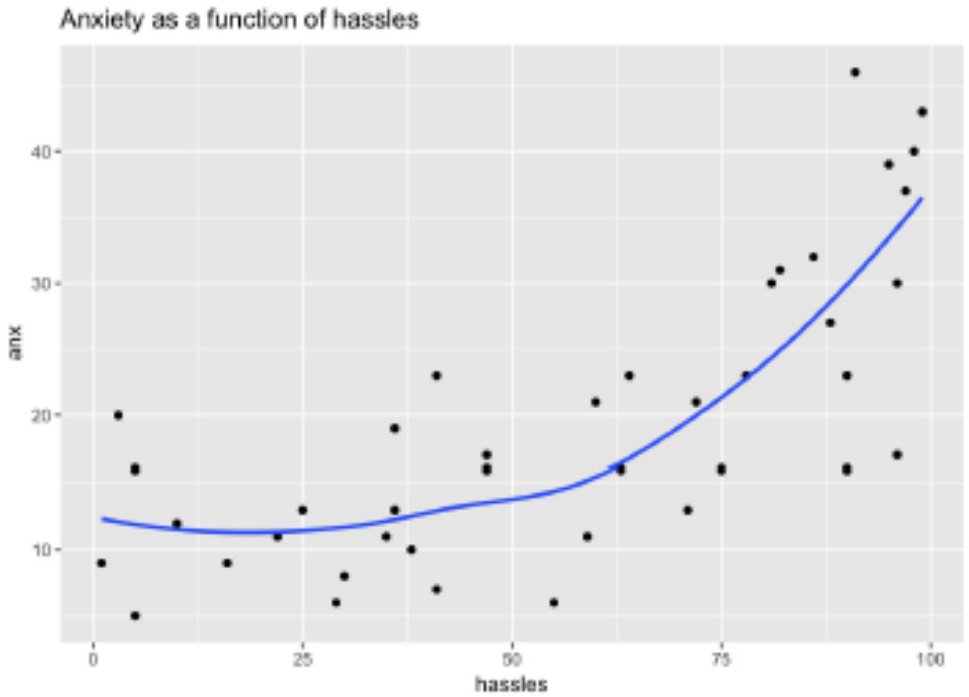


Figure 1:

GAM in R

- mgcv package

```
gam(formula, family, data)
```

- Family:
- gaussian (default): “regular” regression
  - binomial: probabilities
  - poisson/quasipoisson: counts

Best for larger data sets

The s() function

```
anx ~ s(hassles)
```

- s() designates that variable should be non-linear
- Use s() with continuous variables
  - More than about 10 unique values

Example in R

```
model <- gam(anx ~ s(hassles), data = hassleframe,
             family = gaussian)
summary(model)
## ...
##
## R-sq.(adj) = 0.619 Deviance explained = 64.1%
```