

Count



<https://www.youtube.com/watch?v=ZIniljT5IJI>

Count Data

DR. SUSAN SIMMONS

MSA 2025

Examples of Count Data

- Number of bicycles rented at a bicycle shop
- Number of highway deaths
- Number of customers visiting a store
- Number of diseased trees
- Number of people with Dengue Fever in Peru
- Number of open data science jobs
- Many, many more....

Poisson Distribution

Most common distribution to model count data is the Poisson distribution

Why not Normal distribution?

- **Mean must be positive**
- **Variance must be constant**
- **Errors are more appropriate with Poisson regression (when dealing with count data)**

Poisson Distribution

The Poisson distribution:

$$P(y) = \frac{e^{-\lambda} \lambda^y}{y!}$$

The 'λ' in the distribution is the mean (and variance!!) of this distribution!!

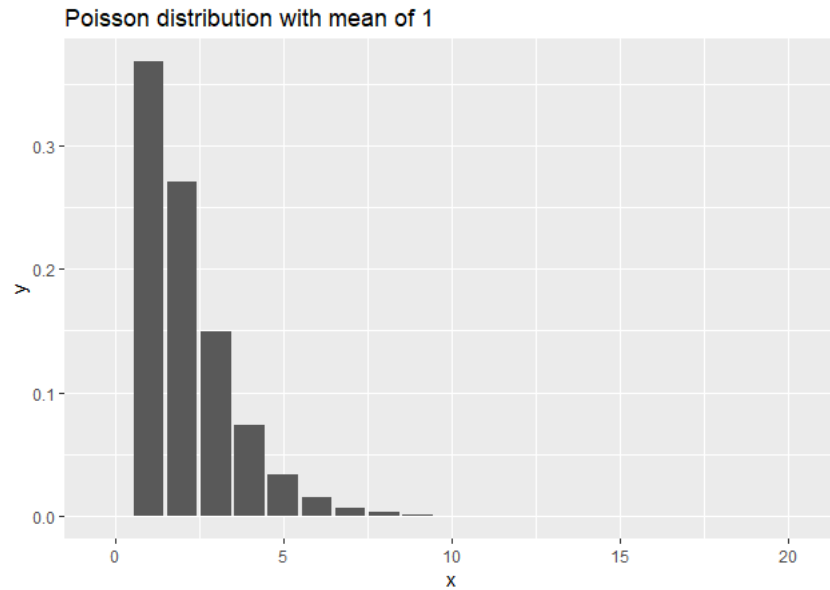
NOTE: Mean is EQUAL to variance

NOTE: Mean is always positive

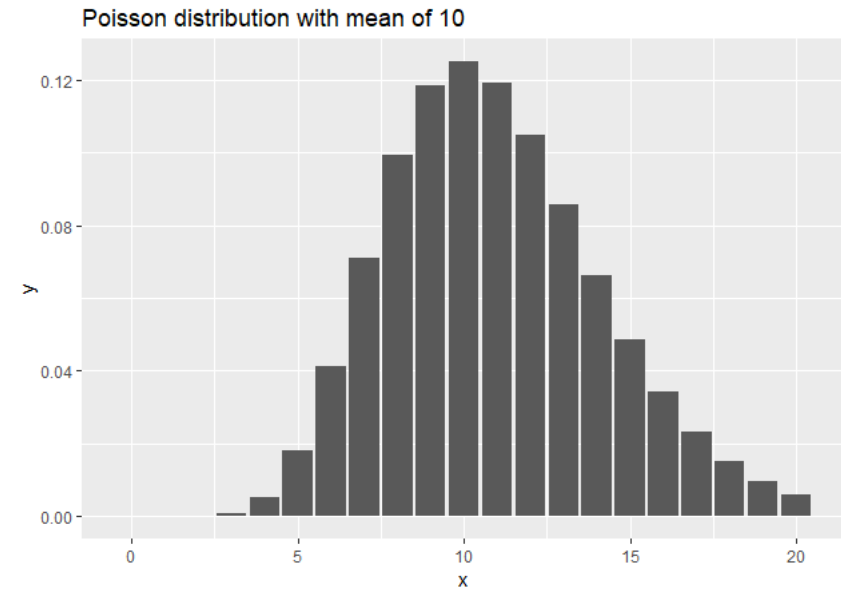
We will model the mean of this distribution

Examples of Poisson distribution

$$\lambda=1$$



$$\lambda=10$$



Poisson Regression

Poisson regression

In Poisson regression, we model the mean (λ_i)

The mean, λ_i , must be positive, however,

$$\lambda_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon_i$$

Can be negative!!!

Poisson regression

In Poisson regression, we model the mean (λ_i)

The mean, λ_i , must be positive, soooo, we force it to be positive... $\lambda_i = e^{x\beta}$

Poisson regression

In Poisson regression, we model the mean (λ_i)

The mean, λ_i , must be positive, sooooo, we force it to be positive... $\lambda_i = e^{x\beta}$

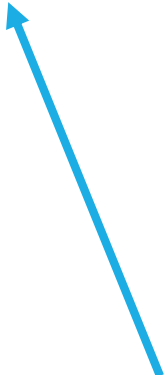
$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon_i$$

Poisson regression

In Poisson regression, we model the mean (λ_i)

The mean, λ_i , must be positive, sooooo, we force it to be positive... $\lambda_i = e^{x\beta}$

$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon_i$$



This is called a “link” function...links mean to the linear predictor!

Other link functions....

Identity Link

$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon_i$$

Log Link

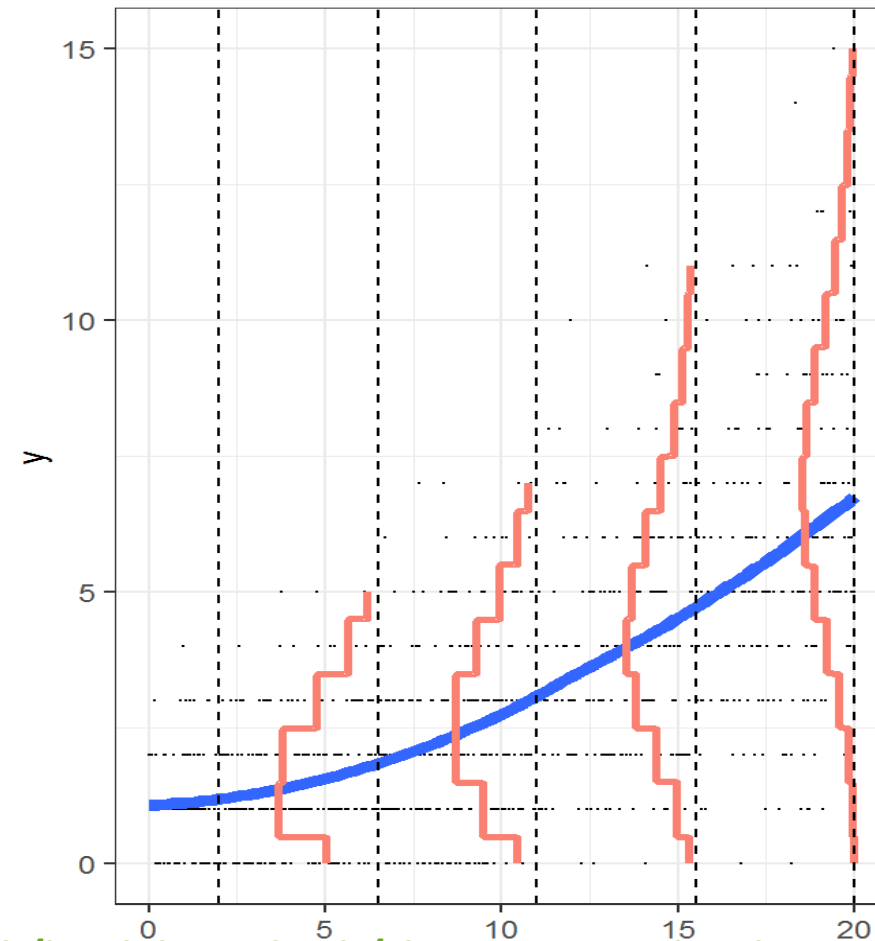
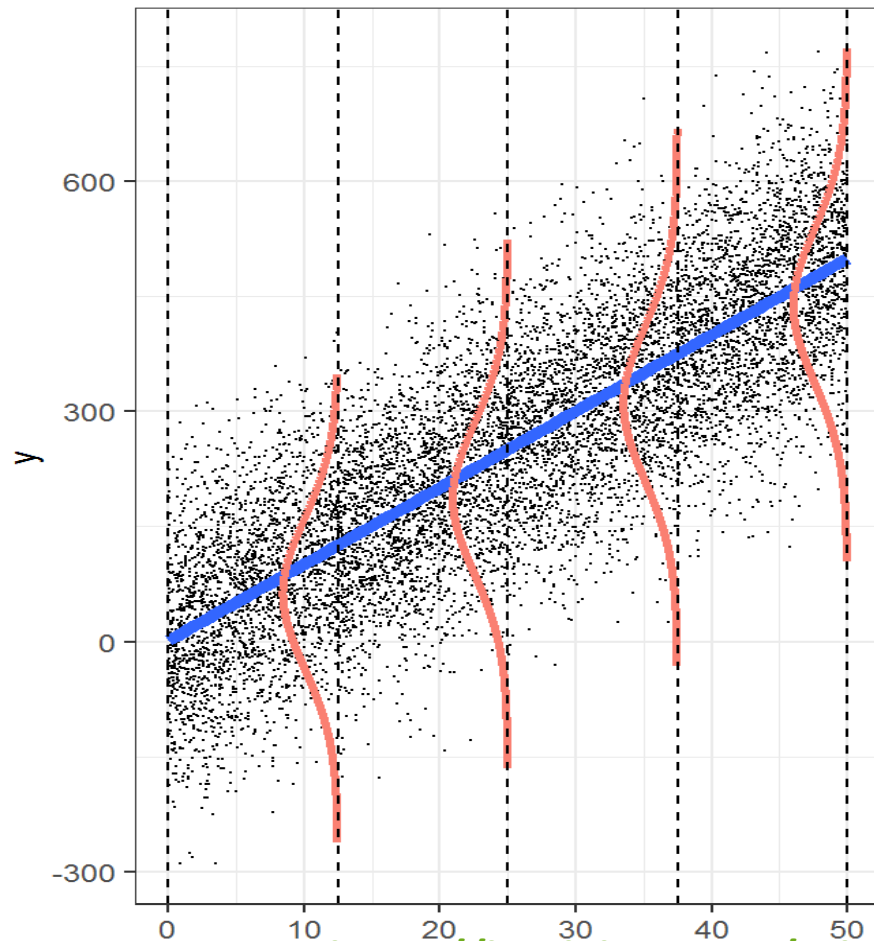
$$\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon_i$$

Logit Link

$$\log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon_i$$

Poisson regression

Julie Legler and Paul Roback



<https://bookdown.org/robback/bookdown-bysh/ch-poissonreg.html>

Poisson regression

We model $\log(\lambda_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon_i$

Assumptions

- $E(Y_i | X_i) = V(Y_i | X_i)$ (conditional mean = conditional variance)
- Independent observations
- Linearity in the mean of the response (yes, we can get around this idea by binning if we need to)

Some notes:

- In most algorithms, variance of estimators is calculated using the Hessian matrix (inverse of the second derivatives). If you see that the Hessian is *singular*, you need to respecify model.
- If the algorithm does *not converge*, you need to respecify the model (or try another minimization algorithm...in SAS, default is Newton Raphson (NRA) however, QN (Quasi Newton is an alternative).
- Careful of potential multicollinearity.

Poisson example

Bike data set:

Modeling the Number of bikes rented at a bike shop

Multicollinearity concerns:

- atemp and temp (temp is slightly higher in correlation with cnt)
- Mnth and season (going to use season)

Not using registered or casual (registered+ casual = cnt)

Only two years (not using yr)

Not using working day (will use weekday + holiday)

```
model.pois <- glm(cnt ~ factor(season) + factor(hr)+ factor(weekday) + factor(weathersit) + holiday+ temp +  
hum + windspeed, family="poisson", data=train)  
summary(model.pois)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.364499	0.007645	440.072	<2e-16	***
factor(season) 2	0.349867	0.002688	130.148	<2e-16	***
factor(season) 3	0.267660	0.003269	81.876	<2e-16	***
factor(season) 4	0.486782	0.002355	206.723	<2e-16	***
factor(hr) 1	-0.461017	0.009868	-46.717	<2e-16	***
factor(hr) 2	-0.871410	0.011498	-75.785	<2e-16	***
factor(hr) 3	-1.440937	0.014065	-102.450	<2e-16	***
factor(hr) 4	-2.070732	0.018488	-112.007	<2e-16	***
factor(hr) 5	-0.929075	0.011753	-79.048	<2e-16	***
factor(hr) 6	0.410757	0.008067	50.917	<2e-16	***
factor(hr) 7	1.440151	0.006949	207.244	<2e-16	***
factor(hr) 8	1.931444	0.006673	289.423	<2e-16	***
factor(hr) 9	1.396782	0.006912	202.080	<2e-16	***
factor(hr) 10	1.117318	0.007123	156.865	<2e-16	***


```
model.pois <- glm(cnt ~ factor(season) + factor(hr)+ factor(weekday) + factor(weathersit) + holiday+ temp +
hum + windspeed, family="poisson", data=train)
summary(model.pois)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.364499	0.007645	440.072	<2e-16	***
factor(season) 2	0.349867	0.002688	130.148	<2e-16	***
factor(season) 3	0.267660	0.003269	81.876	<2e-16	***
factor(season) 4	0.486782	0.002355	206.723	<2e-16	***
factor(hr) 1	-0.461017	0.009868	-46.717	<2e-16	***
factor(hr) 2	-0.871410	0.011498	-75.785	<2e-16	***
factor(hr) 3	-1.440937	0.014065	-102.450	<2e-16	***
factor(hr) 4	-2.070732	0.018488	-112.007	<2e-16	***
factor(hr) 5	-0.929075	0.011753	-79.048	<2e-16	***
factor(hr) 6	0.410757	0.008067	50.917	<2e-16	***
factor(hr) 7	1.440151	0.006949	207.244	<2e-16	***
factor(hr) 8	1.931444	0.006673	289.423	<2e-16	***
factor(hr) 9	1.396782	0.006912	202.080	<2e-16	***
factor(hr) 10	1.117318	0.007123	156.865	<2e-16	***

$\exp(0.349867) - 1 = 0.42....$
There appears to be a 42% increase in the average number of bikes rented in summer (season =2) compared to spring (season=1) holding all other variables constant.

```
model.pois <- glm(cnt ~ factor(season) + factor(hr)+ factor(weekday) + factor(weathersit) + temp + hum +
windspeed, family="poisson", data=train)
summary(model.pois)
```

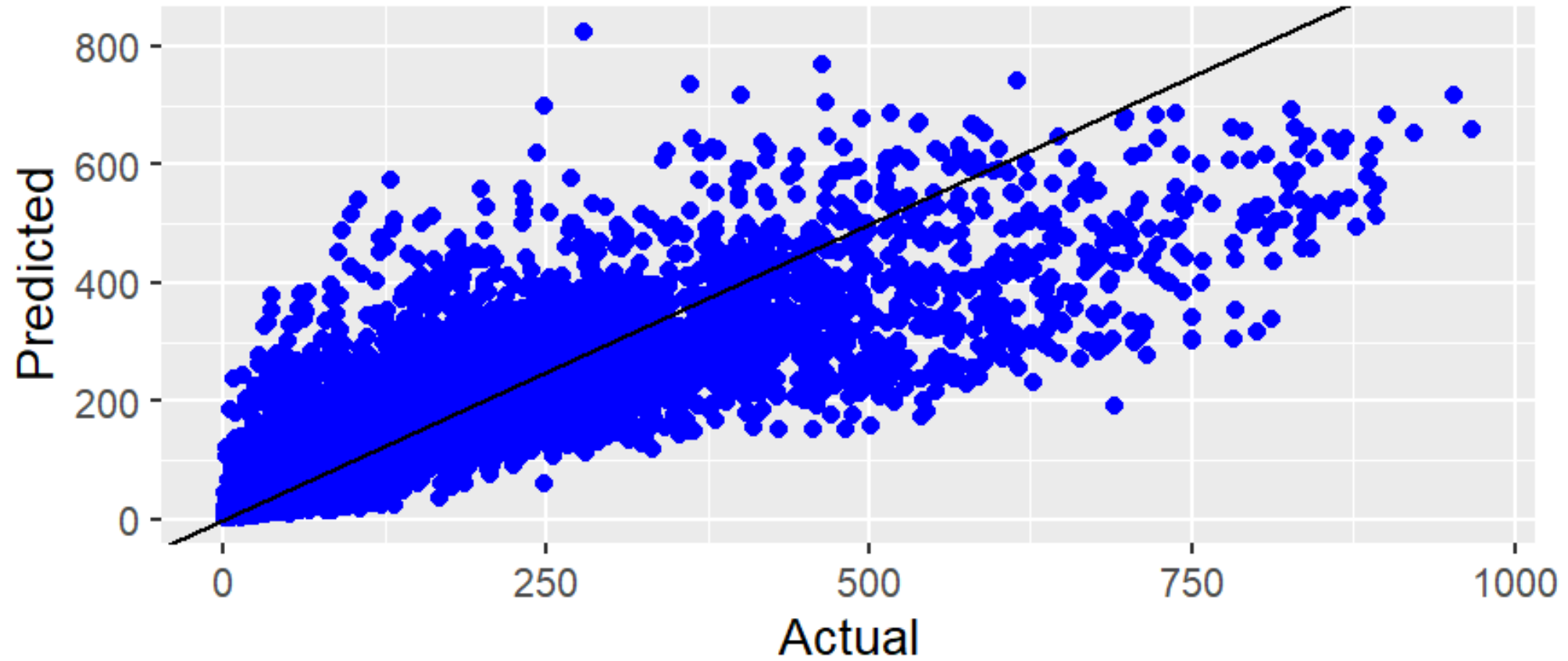
Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.362217	0.007645	439.780	<2e-16	***
factor(season) 2	0.354407	0.002689	131.792	<2e-16	***
factor(season) 3	0.273769	0.003269	83.756	<2e-16	***
factor(season) 4	0.488661	0.002355	207.513	<2e-16	***
factor(hr) 1	-0.459795	0.009868	-46.594	<2e-16	***
factor(hr) 2	-0.872058	0.011498	-75.841	<2e-16	***
factor(hr) 3	-1.440461	0.014065	-102.417	<2e-16	***
factor(hr) 4	-2.070915	0.018488	-112.017	<2e-16	***
factor(hr) 5	-0.930194	0.011753	-79.144	<2e-16	***
factor(hr) 6	0.410367	0.008067	50.869	<2e-16	***
factor(hr) 7	1.440259	0.006949	207.260	<2e-16	***
factor(hr) 8	1.932160	0.006673	289.532	<2e-16	***
factor(hr) 9	1.396795	0.006912	202.084	<2e-16	***
factor(hr) 10	1.117445	0.007123	156.884	<2e-16	***

$\exp(0.354407) - 1 = 0.425....$
There appears to be a 42.5% increase in the average number of bikes rented in summer (season =2) compared to spring (season=1) holding all other variables constant.

Pseudo $R^2 = 1 -$
Deviance(full)/
Deviance(null)=
0.736

Test data predictions



Negative Binomial Regression

Negative binomial

What happens if the conditional variance is bigger than the conditional mean? This is called “overdispersion”. We can use the Negative binomial in this case....

Model: $\log(\mu_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon_i$

Extra parameter for “overdispersion” (this is α in R)

More Negative Binomial

Some notes:

- In most algorithms, variance of estimators is calculated using the Hessian matrix (inverse of the second derivatives). If you see that the Hessian is singular, you need to respecify model.
- If the algorithm does not converge, you need to respecify the model.
- Careful of potential multicollinearity.

Negative binomial is NOT recommended for small samples.

```
model.negbin<-glm.nb(cnt ~ factor(season) + factor(hr)+ factor(weekday) + factor(weathersit) +  
holiday+ temp + hum + windspeed, link=log, data=train)
```

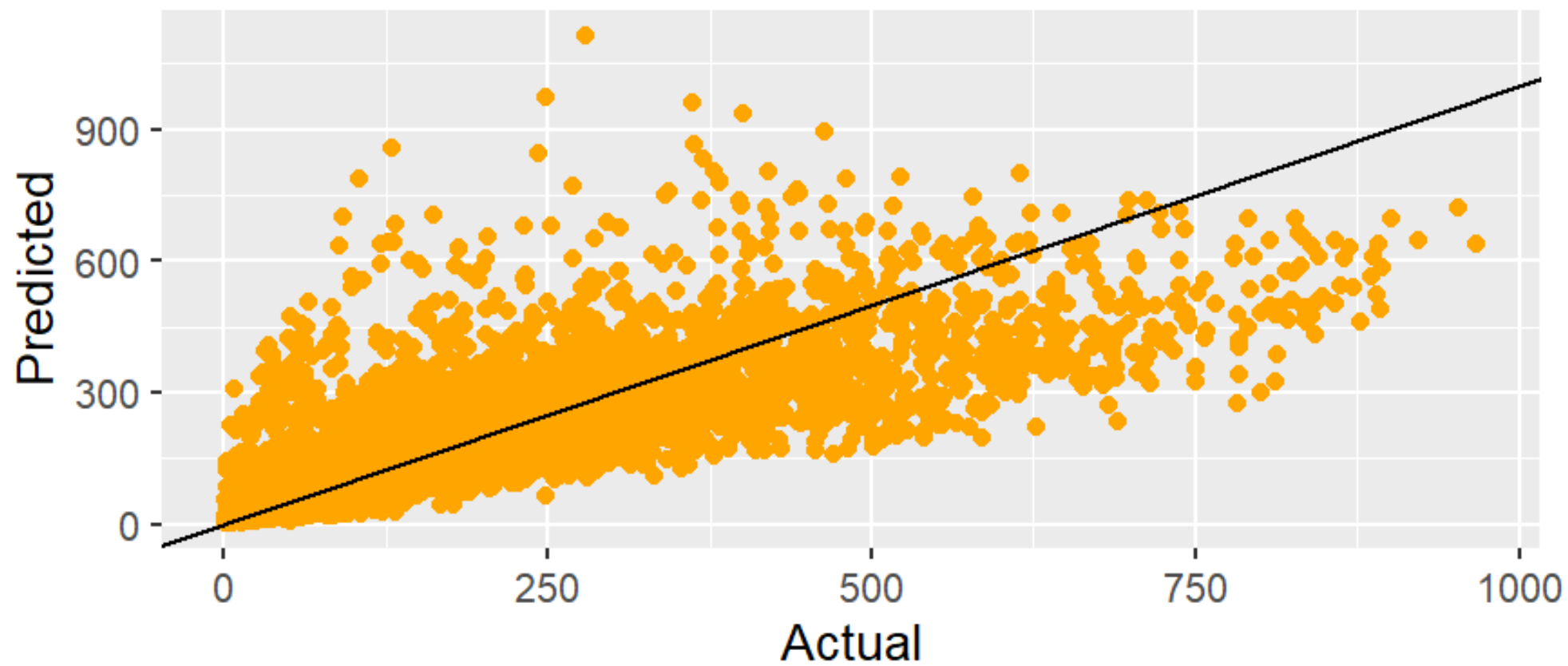
```
summary(model.negbin)
```

Coefficients:

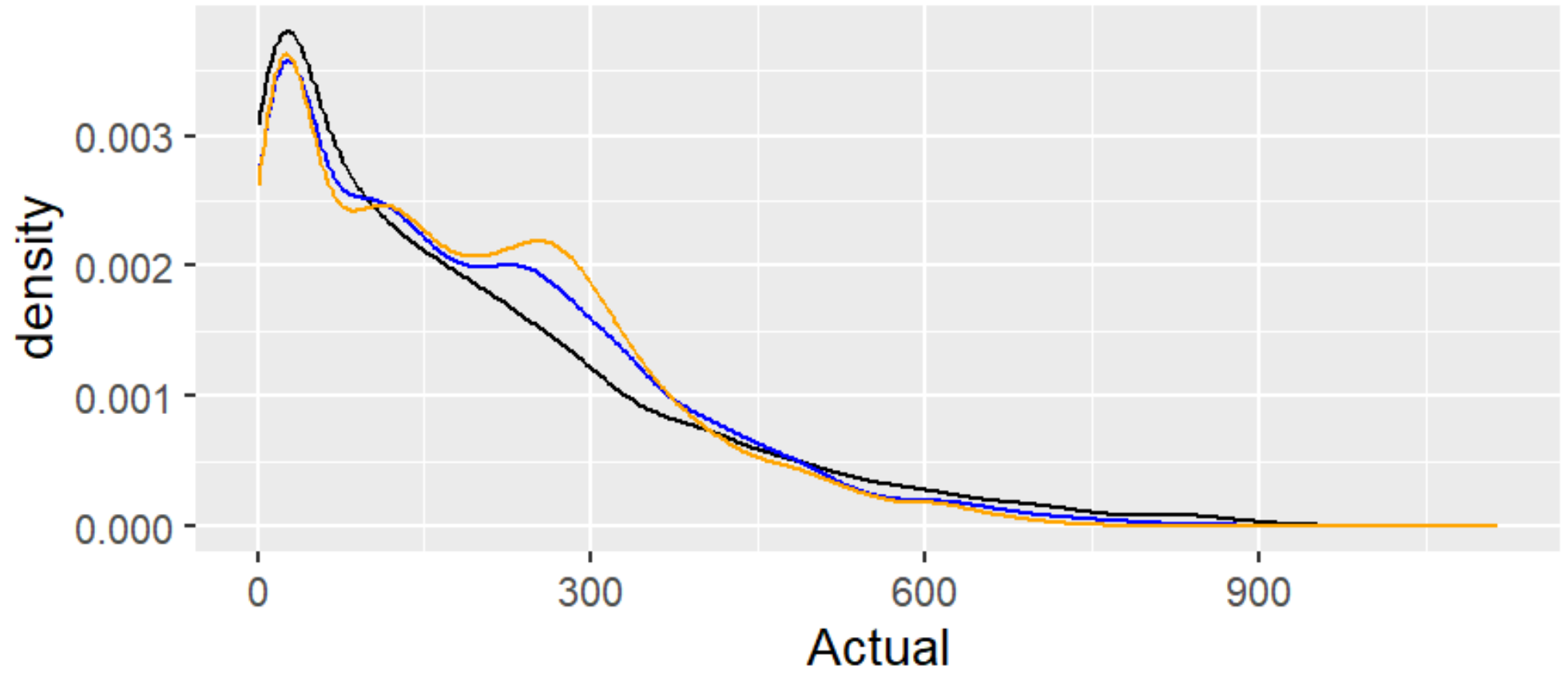
	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.362333	0.042688	78.765	< 2e-16
factor(season) 2	0.290339	0.019557	14.846	< 2e-16
factor(season) 3	0.218267	0.025125	8.687	< 2e-16
factor(season) 4	0.476707	0.016889	28.226	< 2e-16
factor(hr) 1	-0.469994	0.037928	-12.392	< 2e-16
factor(hr) 2	-0.853003	0.038747	-22.015	< 2e-16
factor(hr) 3	-1.436058	0.039515	-36.342	< 2e-16

Pseudo R2 =
0.739

Using test data



Black = Actual
Blue = Negative binomial
Orange = Poisson



Zero-inflated Poisson regression

Negative binomial example

The data (Medicare) is a cross-sectional data set from health economics. There are a total of 4406 individuals, aged 66 and over, who are covered by Medicare, a public insurance program. Originally obtained from the US National Medical Expenditure Survey (NMES) for 1987/88. The variables we will be focusing on are:

Ofp – number of physicians office visits

Hosp – number of hospital stays

Health – self-perceived health status

Numchron – number of chronic conditions

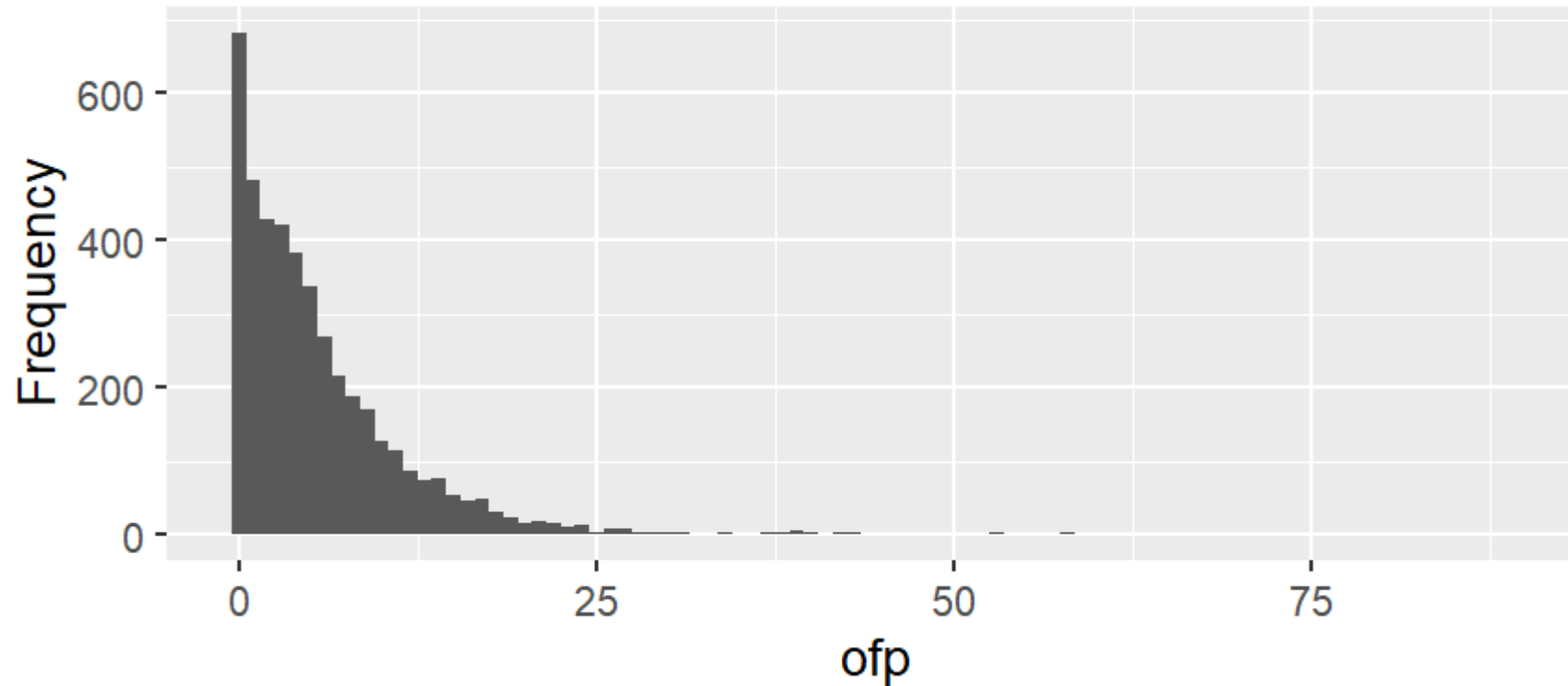
Gender –gender

School – number of years of education

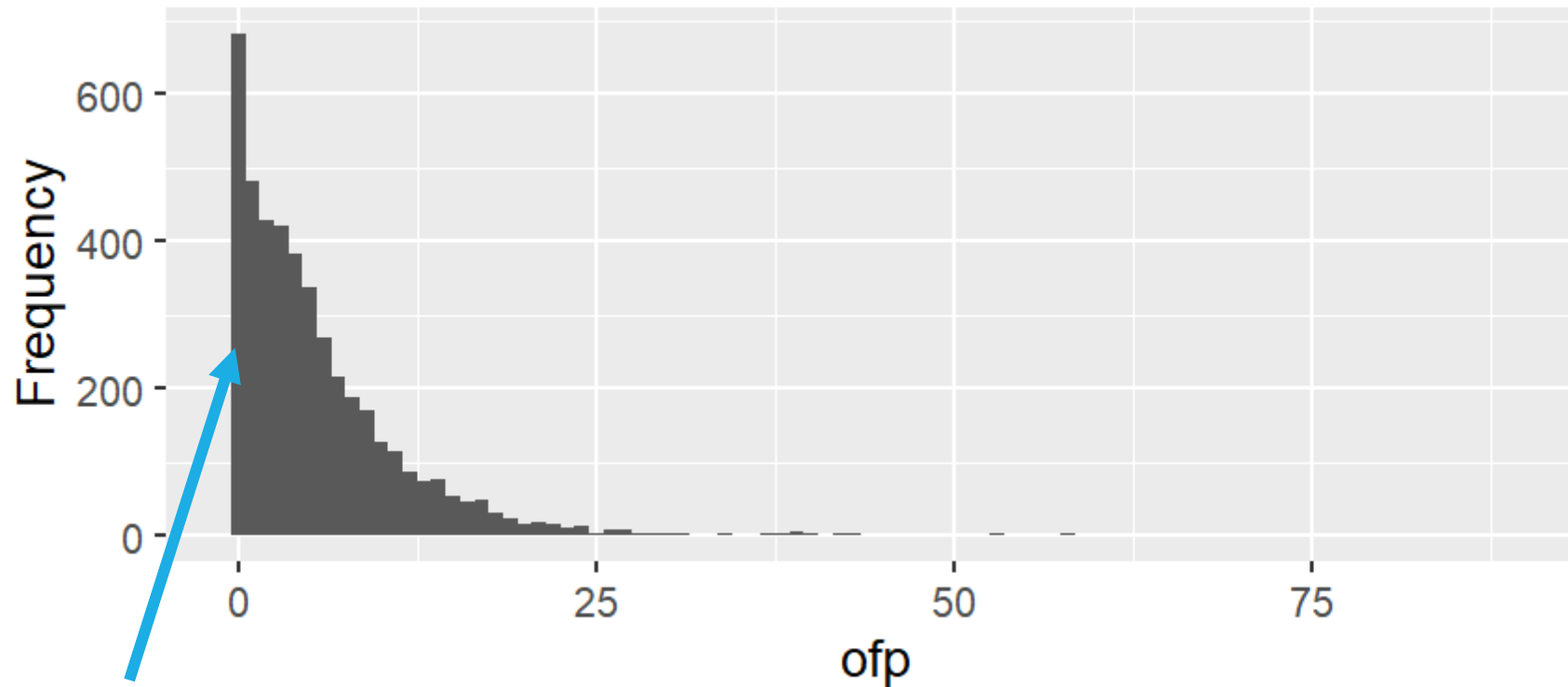
Privins – indicator variable for private insurance

Some count data have A LOT of zeros

Both the Poisson and Negative binomial can be fit with 'Zero-inflated' models



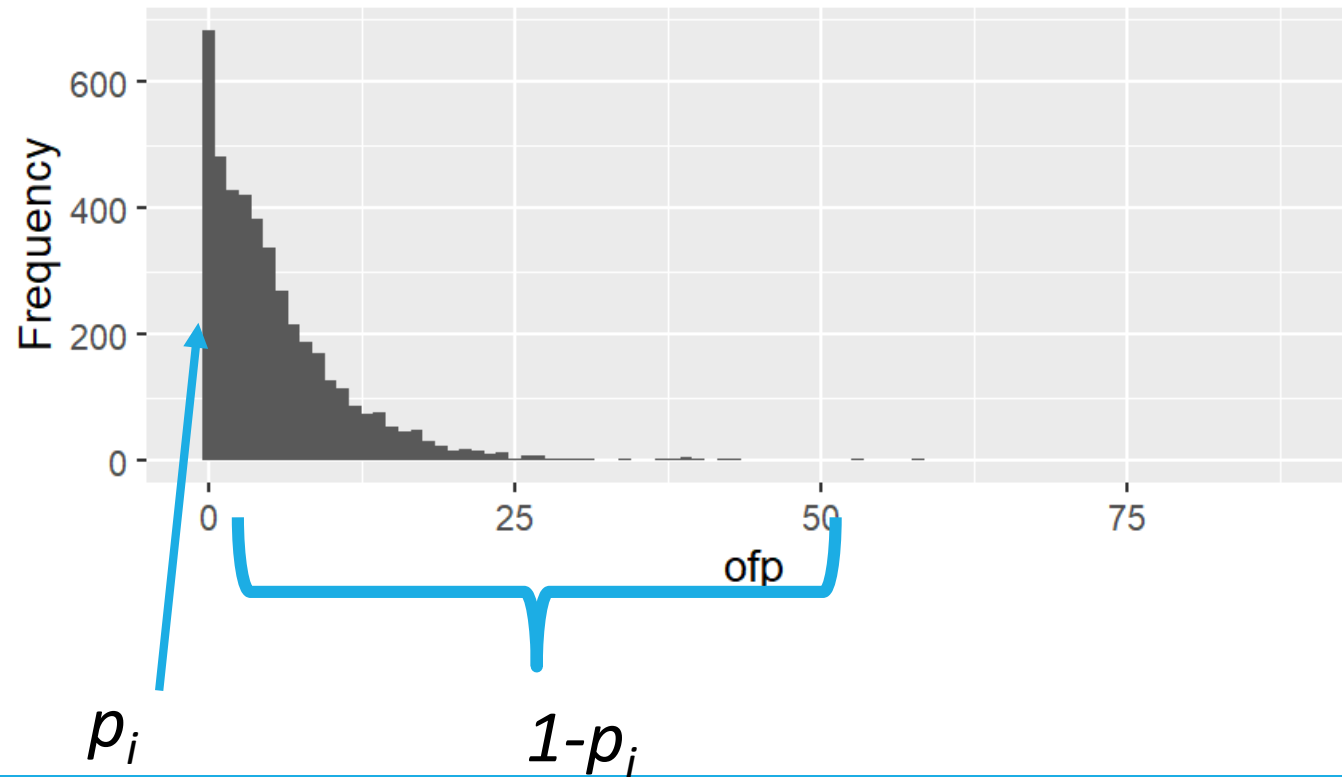
Zero-inflated Poisson



More zeros than one would expect
with a Poisson distribution

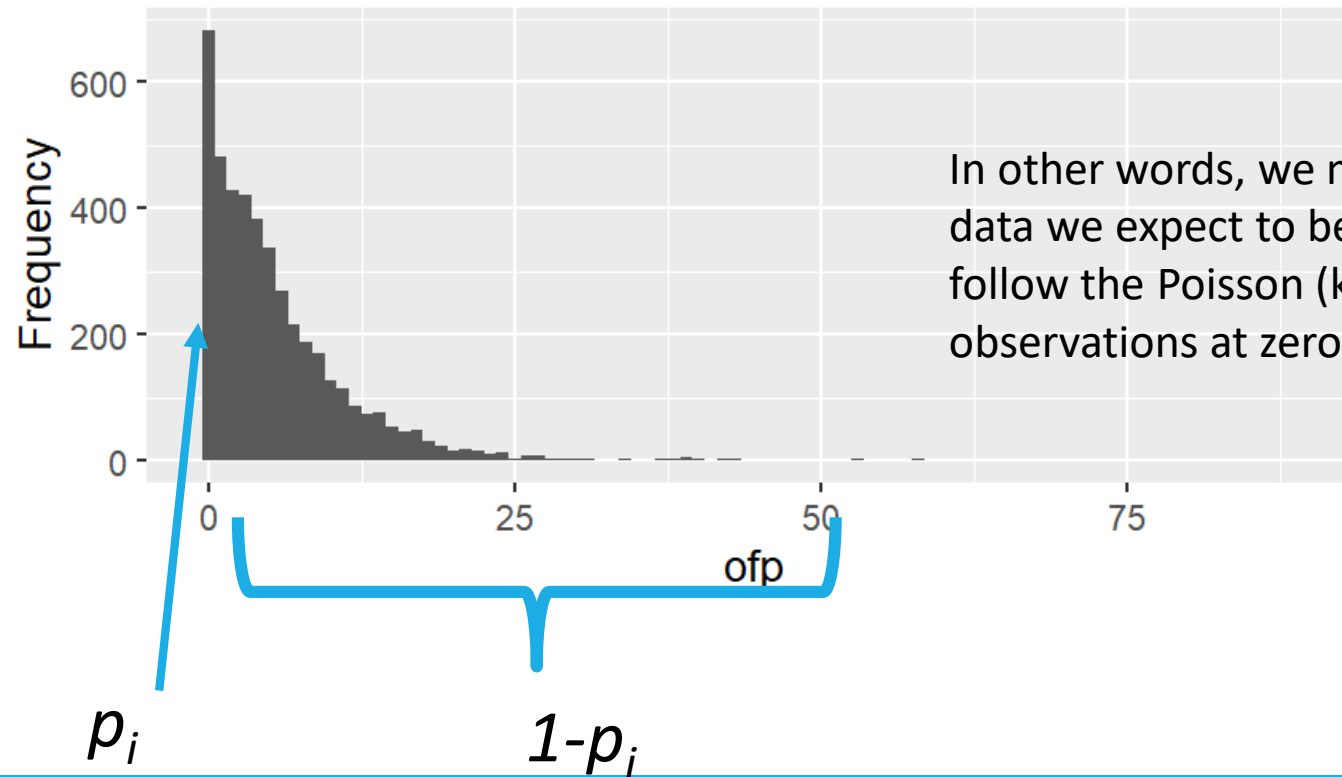
Zero-inflated Poisson

Since there are more zeros than one would expect with a Poisson distribution, we need to model this extra amount of zeros



Zero-inflated Poisson

Since there are more zeros than one would expect with a Poisson distribution, we need to model this extra amount of zeros



In other words, we model the what percent of the data we expect to be at 0 and what percent would follow the Poisson (keep in mind that some of the observations at zero are from the Poisson distribution)

Zero-inflated Poisson

There are two pieces that need to be modeled:

- Extra zero's - Predict having 0's versus not having 0's (binary outcome)
- The Poisson regression

The first part of the model fits a Logistic regression (predict 0 versus having a “count”...this is treated as binary)

The second part models the count data as Poisson with mean λ

Zero-inflated Poisson

There are two pieces that need to be modeled:

- Extra zero's - Predict having 0's versus not having 0's (binary outcome)
- The Poisson regression

The first part of the model fits a Logistic regression (predict 0 versus having a “count”...this is treated as binary)

CAN USE VARIABLES TO PREDICT THIS PART

The second part models the count data as Poisson with mean λ

Zero-inflated Poisson

There are two pieces that need to be modeled:

- Extra zero's - Predict having 0's versus not having 0's (binary outcome)
- The Poisson regression

The first part of the model fits a Logistic regression (predict 0 versus having a “count”...this is treated as binary)

The second part models the count data as Poisson with mean λ

CAN USE VARIABLES TO PREDICT THIS PART

```
model.zpois2<-zeroinfl(ofp ~
```

```
factor(health) + factor(gender)+factor(privins)+hosp+numchron+school|  
factor(gender)+factor(privins)+hosp+numchron+school,data=dat,dist='poisson')
```

```
model.zpois2<-zeroinfl(ofp ~
```

```
factor(health) + factor(gender)+factor(privins)+hosp+numchron+school|  
factor(gender)+factor(privins)+hosp+numchron+school,data=train,dist='poisson')
```



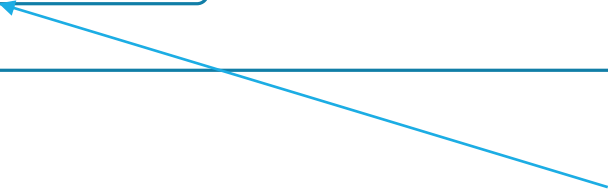
These factors are for the Poisson regression part

```
model.zpois2<-zeroinfl(ofp ~
```

```
factor(health) + factor(gender)+factor(privins)+hosp+numchron+school |
```

```
factor(gender)+factor(privins)+hosp+numchron+school) data=train,dist='poisson')
```

```
model.zpois2<-zeroinfl(ofp ~  
factor(health) + factor(gender)+factor(privins)+hosp+numchron+school |  
factor(gender)+factor(privins)+hosp+numchron+school,data=dat,dist='poisson')
```



These factors are for the Logistic regression part

```
model.zpois2<-zeroinfl(ofp ~ factor(health) + factor(gender)+factor(privins)+hosp+numchron+school |
factor(gender)+factor(privins)+hosp+numchron+school,data=train2,dist='poisson')
```

Count model coefficients (poisson with log link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	1.405600	0.024179	58.134	< 2e-16	***
factor(health)excellent	-0.307366	0.031265	-9.831	< 2e-16	***
factor(health)poor	0.253416	0.017706	14.313	< 2e-16	***
factor(gender)male	-0.062352	0.013056	-4.776	1.79e-06	***
factor(privins)yes	0.080533	0.017147	4.697	2.65e-06	***
hosp	0.159014	0.006060	26.240	< 2e-16	***
numchron	0.101846	0.004721	21.573	< 2e-16	***
school	0.019169	0.001873	10.232	< 2e-16	***

Zero-inflation model coefficients (binomial with logit link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.05937	0.14040	-0.423	0.672392	
factor(gender)male	0.41806	0.08920	4.687	2.77e-06	***
factor(privins)yes	-0.75373	0.10211	-7.381	1.57e-13	***
hosp	-0.30669	0.09121	-3.363	0.000772	***
numchron	-0.53972	0.04419	-12.212	< 2e-16	***
school	-0.05560	0.01218	-4.564	5.02e-06	***

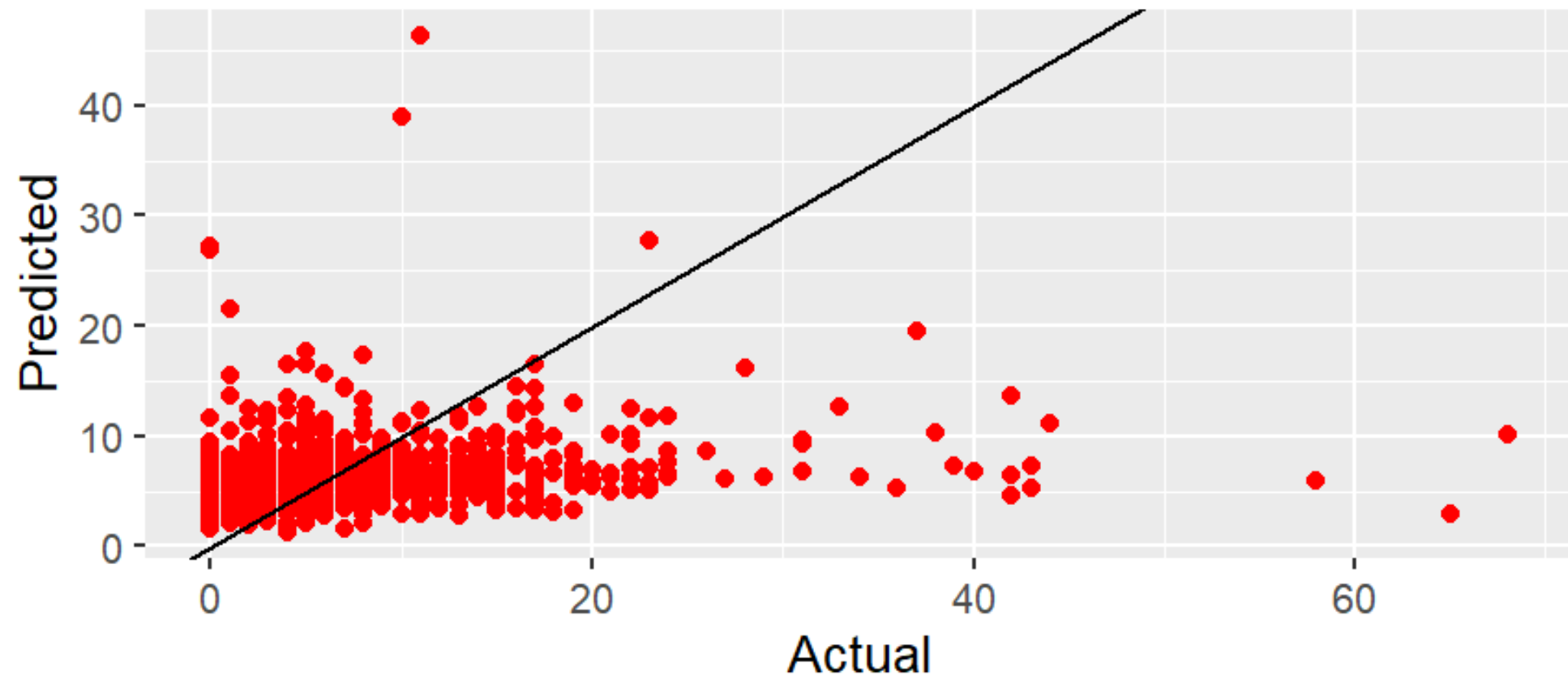
```
model.zpois2<-zeroinfl(ofp ~ factor(health) + factor(gender)+factor(privins)+hosp+numchron+school |
factor(gender)+factor(privins)+hosp+numchron+school,data=train2,dist='poisson')
```

Count model coefficients (poisson with log link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	1.405600	0.024179	58.134	< 2e-16	***
factor(health)excellent	-0.307366	0.031265	-9.831	< 2e-16	***
factor(health)poor	0.253416	0.017706	14.313	< 2e-16	***
factor(gender)male	-0.062352	0.013056	-4.776	1.79e-06	***
factor(privins)yes	0.080533	0.017147	4.697	2.65e-06	***
hosp	0.159014	0.006060	26.240	< 2e-16	***
numchron	0.101846	0.004721	21.573	< 2e-16	***
school	0.019169	0.001873	10.232	< 2e-16	***

Zero-inflation model coefficients (binomial with logit link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.05937	0.14040	-0.423	0.672392	
factor(gender)male	0.41806	0.08920	4.687	2.77e-06	***
factor(privins)yes	-0.75373	0.10211	-7.381	1.57e-13	***
hosp	-0.30669	0.09121	-3.363	0.000772	***
numchron	-0.53972	0.04419	-12.212	< 2e-16	***
school	-0.05560	0.01218	-4.564	5.02e-06	***



Zero-inflated Negative binomial

Zero-inflated Negative Binomial

Same idea as zero-inflated Poisson, except we now have overdispersion

Can again use Log-Likelihood test to see if overdispersion is needed

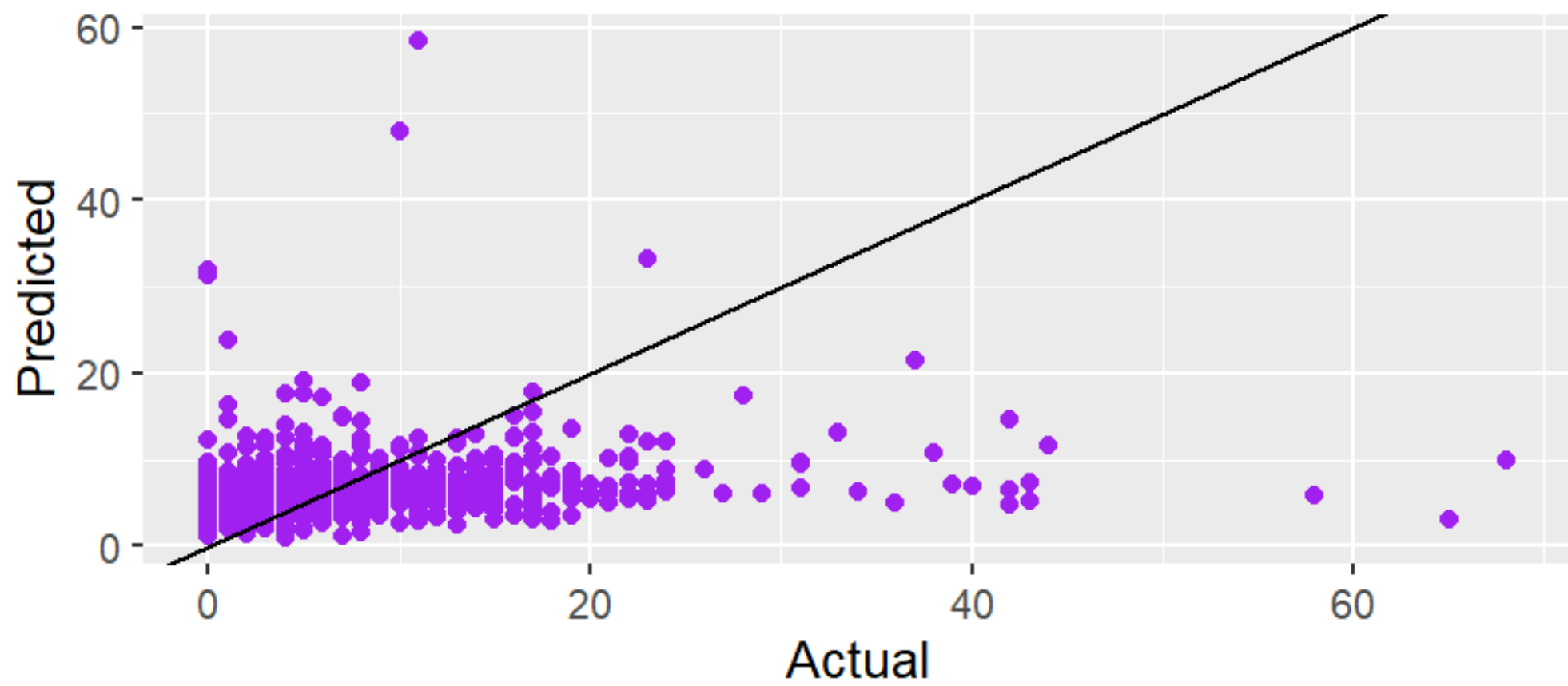
```
model.znbin2<-zeroinfl(ofp ~ factor(health) + factor(gender)+factor(privins)+hosp+numchron+school|
factor(gender)+factor(privins)+hosp+numchron+school,data=train2,dist='negbin')
```

Count model coefficients (negbin with log link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	1.193716	0.056661	21.068	< 2e-16	***
factor(health)excellent	-0.319339	0.060405	-5.287	1.25e-07	***
factor(health)poor	0.285133	0.045093	6.323	2.56e-10	***
factor(gender)male	-0.080277	0.031024	-2.588	0.00967	**
factor(privins)yes	0.125865	0.041588	3.026	0.00247	**
hosp	0.201477	0.020360	9.896	< 2e-16	***
numchron	0.128999	0.011931	10.813	< 2e-16	***
school	0.021423	0.004358	4.916	8.82e-07	***
Log(theta)	0.394144	0.035035	11.250	< 2e-16	***

Zero-inflation model coefficients (binomial with logit link):

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-0.04684	0.26855	-0.174	0.86154	
factor(gender)male	0.64766	0.20011	3.236	0.00121	**
factor(privins)yes	-1.17558	0.22012	-5.341	9.26e-08	***
hosp	-0.80046	0.42081	-1.902	0.05715	.
numchron	-1.24790	0.17831	-6.999	2.59e-12	***
school	-0.08378	0.02625	-3.191	0.00142	**



Thank you!
Happy counting....
