This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the `README.md` for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

1.a

Lets start with the definition that integral of density function is equal to 1 over the entire space:

$\int p(y;\eta)dy = 1$

Applying same to exponential family distribution and trying to find out $a(\eta)$:

$\int p(y;\eta)dy = \int b(y)\exp(\eta y - a(\eta))dy = 1$ i.e

$\int b(y)\exp(\eta y - a(\eta))dy = 1$

We can rewrite this as:

$\exp(-a(\eta))\int b(y)\exp(\eta y)dy = 1$

$\exp(a(\eta)) = \int b(y)\exp(\eta y)dy$

$a(\eta) = \log\int b(y)\exp(\eta y)dy$

Lets take derivative of $a(\eta)$ with respect to η and apply hint provided in 1.a

$\frac{\partial a(\eta)}{\partial \eta} = \frac{\partial}{\partial(\eta)}\log\int b(y)\exp(\eta y)dy$

$= \frac{\int yb(y)\exp(\eta y)dy}{\int b(y)\exp(\eta y)dy}$

As per pervious steps we can replace denominator with $\exp a(\eta)$

$= \frac{\int yb(y)\exp(\eta y)dy}{\exp a(\eta)}$

$= \int yb(y)\exp(\eta y - a(\eta))dy = E[Y;\eta]$

This proves that the first derivate of $a(\eta)$ w.r.t η is equivalent to the mean of exponentail family distribution

1.b

Lets startby computing second derivative of $a(\eta)$ w.r.t η using defintion computed in previous answer 1.a
$\frac{\partial^2 a(\eta)}{\partial \eta^2} = \frac{\partial}{\partial(\eta)} \int yb(y)\exp(\eta y - a(\eta))dy$

$= \int yb(y)\exp(\eta y - a(\eta))(y - a'(\eta))dy$

$= \int p(y;\eta)y^2 dy - a'(\eta)\int p(y;\eta)ydy$

$= E[Y^2;\eta] - E[Y;\eta]E[Y;\eta]$

$= Var[Y;\eta]$

This shows that the variance of an exponential family distribution is the second derivative of the log-partition function w.r.t. the natural parameter.

1.c

Lets start with definition of negative log likelyhood $NLL = -log(p(y;\eta))$
$= -log(b(y)\exp(\eta y - a(\eta)))$
$= -(log(b(y)) + log(\exp(\eta y - a(\eta))))$
This can be rewritten as
$= -(log(b(y)) + (\eta y - a(\eta)))$
$= -(log(b(y)) + (\theta^T xy - a(\theta^T x)))$

Now lets take hessian of the NLL wrt to θ :
$\nabla^2_\theta(NLL) = \nabla^2_\theta(-(log(b(y)) + (\theta^T xy - a(\theta^T x))))$
$= \nabla^2_\theta(-(\theta^T xy - a(\theta^T x)))$
The second order derivative of $-(\theta^T xy)$ w.r.t θ is equal to 0, so:
$= \nabla^2_\theta(a(\theta^T x))$
$= Var(Y;\eta)$
As variance of any probability distribution is non negative and therefore the Hessian of GLM's NLL loss is PSD, and hence convex.

2.a

$$J(\theta) =$$

$$\frac{1}{2}\sum_{i=1}^{n}(h_\theta(\hat{x}^{(i)} - y^{(i)})^2.$$

Differentiating this objective, we get:

$$\nabla_\theta J(\theta) =$$

$$\frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(\hat{x}) - y)^2 =$$

$$(h_\theta(\hat{x}) - y)x_j$$

The gradient descent update rule is

$$\theta := \theta - \lambda\nabla_\theta J(\theta)$$

which reduces here to:
$\theta := \theta - \lambda(h_\theta(\hat{x}) - y)x_j$
Rearranging terms and in general for i
$\theta := \theta + \lambda(y^{(i)} - h_\theta(\hat{x}^{(i)}))\hat{x}_j^{(i)}$

2.d

For k=1 (or 2) the fit is almost a straight line
For k=3 the fit starts to show the sin wave pattern
For k=5,10 the fit is more natural to data points and is closer also
For k=20 the curve passes through most of the points and also start showing signs of overfitting as we can see some curvatures beyond the given point.
So as k increases fit is passing through more and more points and tends to overfitting.

2.f

Compared to 2.c we can see the fitted model taking a sin wave pattern.
This is even true for low values of k like 1 or 2.
The reason for this is that the training data is also created using sin function.
After adding a sin(x) to the polynomial regression even for low value of x we get good fit as compared to 2.c

2.h

As the training dataset is small the fitting of the training dataset changes with K as follows:

For polynomial regression,
For lower values of K (= 1 or 2) the fit is not going over any data point
For K (=3,5) fit is closer to training data point or passes through some of the training data points and is more natural
But as K increases(10,20) we can see overfitting i.e the long curves in sin wave

For polynomial and sinusoidal features,
Fitting of the data is more natural even with lower values of K like 1,2,3,5
But as K increases we can see overfitting